



DECUS U.S. CHAPTER SIGs NEWSLETTERS

APL SIG	APL
ARTIFICIAL INTELLIGENCE SIC	AI
BUSINESS APPLICATIONS SIC	BA
COMMERCIAL LANGUAGES SIG	CL
DAARC SIG	DAR
DATA MANAGEMENT SIG	DMS
DATATRIEVE 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



Dear DECUS Member,

Welcome to the latest expansion of services available to you from DECUS -- The DECUS SIGs Newsletters -- previously available to you as many separate newsletters. Those of you familiar with newsletters will recognize that this publication is the combined efforts of all of the national Special Interest Groups (SIGs). We hope you will all take the time to leaf through this weighty document and determine for yourself the wealth of valuable information about Digital computing.

In the past, the individual (and semi-combined) newsletters were available to you separately or in any combination that you desired. Many of you ordered the entire set for \$200.00. The VAX newsletter was available for \$20.00. Many of you ordered multiple newsletters that cost in excess of \$35. It became clear to us that the needs of the typical DECUS member spans multiple special interest areas, in groups of areas that we could not predict. So, we have decided to bundle all of our newsletters into one monthly publication for a subscription rate of \$35. The lower rate has been arrived at by reducing the handling of multiple publication on various publication schedules, not by reducing the content of any of the individual newsletters. We hope the opportunity to receive information about all areas of the DECUS/Digital world at a modest fee will be irresistible to many of you.

The publication you have received today, is a free installment to introduce you to the DECUS SIGs Newsletters. Normal subscribers distribution will begin with our next issue. All of our SIG Newsletter Editors have worked extremely hard to put our best foot forward. We guarantee that we will continue to do so in the coming months as well. Although the size of the publication will vary from month to month, and probably will be largest in the months shortly after the semi-annual symposia, it will always contain valuable information about the technical world of Digital computing. Certainly, one way to insure that fact is to participate in the publication by submitting an article, question or response to questions to the appropriate SIG section. At the back of the newsletter there are instructions on how to submit an article, etc. Be our guest. Remember, DECUS is a volunteer organization and our Newsletter Editors are some of our hardest working volunteers. They need your help and support. Again, welcome to the DECUS SIGs Newsletters. We hope you'll decide to join us as a regular subscriber.

MARK GRUNDLER
COMMUNICATIONS COMMITTEE CHAIR

SANDY KRUEGER
SIG COUNCIL CHAIR

GENERAL TABLE OF CONTENTS

SECTIONS	PAGE NO.
APL SIG	
. Steering Committee Listing	APL- i
. APL SIG Forms "Free Whitlock Committee"	APL- 1
. Anaheim Symposium Highlights	APL- 1
. VAX APL V2 Discussed Over Pizza	APL- 1
. APL Button Slogan Contest	APL- 2
. The Special Character Set	APL- 2
. First Time Fear	APL- 6
. New APL Symposium Rep	APL- 7
. Designing Character Sets for the VT200	APL- 8
. Spring DECUS VAX APL Slides	APL-15
. I/D Space APL Language Interpreter on RSX-11 M-PLUS	APL-24
ARTIFICIAL INTELLIGENCE SIG	
. Steering Committee Listing	AI- i
. From the Editor	AI- 1
. From the Chair	AI- 2
. AI at Digital	AI- 7
. Suggested Reading List	AI- 9
. Book Review	AI-10
BUSINESS APPLICATIONS SIG	
. Steering Committee Listing	BA- i
. Letter from the Editor	BA- 1
. Legal Problems with Software Protection and Software and Hardware Warranties	BA- 3
COMMERCIAL LANGUAGES SIG	
. Steering Committee Listing	CL- i
. User's Guide	CL- 1
. Dear Language Sensitive Editor	CL- 1
. Call for Submissions from Past Wizards	CL- 1
. Birth of a SIG	CL- 2
. It's Magic	CL- 3
. VAX Users: Get Rid of Your System/3 or System/34	CL- 5
. RPG Working Group News	CL- 6
. Stupid Questions	CL- 9
. More Languages	CL-14
DATA ACQUISITION, ANALYSIS, RESEARCH, AND CONTROL SIG	
. Steering Committee Listing	DAR- i
. Letter from the Chair	DAR- 1
DATA MANAGEMENT SIG	
. Steering Committee Listing	DMS- i
. SIG Chair's Message	DMS- 1
DATATRIEVE SIG	
. Steering Committee Listing	DTR- i
. Table of Contents	DTR- 1
. About the Cover	DTR- 1
. Chairman's Corner	DTR- 2
. From the Editor's Pen	DTR- 3
. Horizontal Bargraphs Without Graphics Devices	DTR- 4
. Previews of Coming Attractions	DTR- 9
. VAX-DATATRIEVE Graphics Internals	DTR-10
. Modifying VAX-DTR Plots	DTR-30
. Additional DATATRIEVE Plots	DTR-35

EDUSIG

. Steering Committee Listing	EDU- i
. EDUSIG Home Runs	EDU- 1
. What (or who) is a Lighthouse?	EDU- 3

GRAPHICS SIG

. Steering Committee Listing	GRA- i
. From the Chairman	GRA- 1
. Letter from the Editor	GRA- 2
. Speaking of Standards	GRA- 3

HARDWARE MICRO SIG

. Steering Committee Listing	HMS- i
. Table of Contents	HMS- 1
. From the SIG Chair	HMS- 2
. From the Editor	HMS- 3
. How to Report Hardware Hints and Kinks	HMS- 3
. Hardware Hints & Kinks	HMS- 4
. Okidata Pacemark 2410 Printer	HMS- 6
. Data Translation DT2782 A/D Converter "PGH" Option	HMS- 7
. KDJ11-?? or "All 11/73's are not Created Equal"	HMS- 8

IAS SIG

. Steering Committee Listing	IAS- i
. Curley's Corner	IAS- 1
. Letter from the Editor	IAS- 2
. Letter from Klaus Centemeyer	IAS- 4
. Borger's Browsings - Notes on IAS Version	IAS-11
. Borger's Browsings - Disassemblers	IAS-17
. Using MACROs and Assembly Language Hacks	IAS-22

LANGUAGES & TOOLS SIG

. Steering Committee Listing	L&T- i
. Table of Contents	L&T- 1
. Editor's Note	L&T- 2
. Letter from the Chair	L&T- 3
. Preliminary Plans for Fall Symposium	L&T- 4
. Letters	L&T- 5
. Fun with DCL	L&T- 7
. Documentation Survey Form	L&T- 9
. FORTRAN 8X Standards Improvement Request Ballot	L&T-11
. Status of Work Towards Revision of Programming	L&T-24
Language FORTRAN (FIB-1)	

LARGE SYSTEMS SIG

. Steering Committee Listing	LS- i
. Chairperson's Message	LS- 2
. From The Editor	LS- 3
. Spring 1985 SIG Menu	LS- 4

MUMPS SIG

. Steering Committee Listing	MMP- i
. Mumps is Catching	MMP- 1

NETWORKS SIG

. NETWords Cover	NTW- 1
. Networks SIG Steering Committee	NTW- 3
. In This Issue	NTW- 4
. Editor's Cobwebb	NTW- 5
. Who is Doctor DECnet?	NTW- 6
. Network Troubleshooting and Other Mystical Things.....	NTW- 7
. Doctor DECnet, The Ultimate Networker	
. XEDRIVER Commonly Asked Technical Questions	NTW-11
. DECnet and the High Energy Physics Network	NTW-17
. Datagrams	NTW-20

OFFICE AUTOMATION SIG STEERING COMMITTEE.

. Steering Committee Listing	OA- i
. Table of Contents	OA- 1
. From the Editor	OA- 2
. Chairman's Chat	OA- 2
. Symposium Coordinates	OA- 3
. The Forum	
- ALL-IN-1 Questionnaire	OA- 4
- OASIG Swap Tape	OA- 7
- The OASIG SIR	OA- 9
. ALL-IN-1 in Action	OA-21
. Feature Articles	OA-24
. 12-Bit World	OA-28

PERSONAL COMPUTER SIG

. Steering Committee Listing	PC- i
. Table of Contents	PC- 1
. From the Editor	PC- 2
. How to Submit Articles to the PC Newsletter	PC- 4
. Greetings from the Chairperson	PC- 5
. Rainbow Corner	PC- 6
. Sysop's Corner	PC- 7
. Guest Editorial	PC- 9
. Rainbow Section	PC-10
. DECmate Section	PC-53
. PRO Section	PC-60
. ROBIN Section	PC-63
. General Information	PC-64
. User Group Column	PC-78
. Rainbow Oriented Bulletin Boards	PC-83

RSTS SIG

. Steering Committee Listing	RST- i
------------------------------------	--------

RT SIG

. Steering Committee Listing	RT- i
. Table of Contents	RT- 1
. RT-11 SIG Announcement	RT- 2
. RT-11 VAX/RT Position Paper	RT- 3
. RUNOFF Fix	RT-10
. New Use for Virtual Memory (VM) Handler	RT-11
. Annotated Directory RT-11 SIG Tape Spring 1985	RT-25

RSX SIG

. Steering committee Listing	RSX- i
. From the Editor	RSX- 1
. Letters to the Multitasker	RSX- 2
. The Bag of Tricks: MACRO-11	RSX- 4
. Sort V3.0 - Part I	RSX- 5
. Enhancements to RSX Phase IV DECnet	RSX- 9
. DECnet Changes for RSX-11 M Systems Modified... ..	RSX-11
. Professional 300 Series Communications - Part I	RSX-18
. The QIO	RSX-35

SITE MANAGEMENT & TRAINING SIG

. Steering Committee Listing	SIT- i
. Table of Contents	SIT- 1
. From the Editor	SIT- 2
. VAX SPM: An Advanced Performance Tool for VMS Systems	SIT- 3
. The Site SIG System Manager's Manual	SIT-14
. Wanted	SIT-14
. The Importance of Power	SIT-15
. Wiring the Campus	SIT-23

UNISIG

. Steering Committee Listing	UNI- i
. From the UNISIG Chair	UNI- 1
. Digital's UNIX Program Office	UNI- 3
. From the UNISIG Standards Coordinator	UNI- 3
. From the UNISIG Session Notes Editor	UNI- 4
. Netnews Items	UNI- 4
. Whither UNIX	UNI- 4

VAX SYSTEMS SIG

. Steering Committee Listing	VAX- i
. How to Backup to Magtape over DECnet	VAX- 3
. Plugging a Security Leak in CERBERUS	VAX- 7
. The Bedford Store Is No More	VAX- 9
. More on VMS Timekeeping	VAX-10
. VAX-11 RSX Device Name Performance	VAX-11
. A Proposal on RMS	VAX-19
. More on a Comprehensive Backup SIR	VAX-22
. LeLisp: A LISP Environment for VAX/VMS	VAX-23
. "It's in the Microfiche ..."	VAX-27
. VAX System SIG Tape Time Bomb	VAX-31
. VMS 4.2 XQP breaks previous privileged code	VAX-32
. An Electronic "Campground" For VAX Users	VAX-32
. Building Efficient Images on VAX/VMS	VAX-34
. Global Section Problems	VAX-41
. System Improvement Request Ballot	VAX-44
. VAX System SIG Committee List(redundant)	VAX-73
. INPUT/OUTPUT	VAX-77
. Steering Committee Nominations	VAX-89

LIBRARY SECTION	LIB- 1
------------------------------	---------------

HOW TO SUBMIT AN ARTICLE ... ORDER FORMS SECTION

. How to Submit an Article	HOW- i
. Article Submission for HMS	HOW- 1
. Subscription Service Order Form	HOW- 3
. Membership Application Form	HOW- 5

QUESTIONNAIRE SECTION

. AI SIC Survey and Questionnaire	QU- 3
. Hardware Submission Form	QU- 5
. DATAGRAM	QU- 7
. Personal Computer SIG Questionnaire	QU- 9
. PC PRO 300 Series Wish List Ballot	QU-11
. RSX Questionnaire	QU-13
. VAX Systems SIG Steering Committee Nomination Form	QU-17
. VAX Systems SIR Ballot	QU-19
. VAX Systems Input/Output Submission Form	QU-21
. VAX Systems System Improvement Request Submission Form	QU-23

HELP WANTED -- JUNTA

The APL SIG is very open to new volunteers. The steering committee is organized as a junta with emergency powers for life. The reason for this is a lack of volunteers. We figure that if you don't have a fixed term of office you won't quit until you have to. We've never had an election because we never have had more people available than we needed.

At the moment, the SIG is in desperate need of a VAX APL user to be our VAX contact. The "New DECUS" organizational structure has also created lots of positions for SIG reps to committees. We would also like to have more foreign contacts.

As a junta, we can be very flexible on your duties. Wouldn't you like to have emergency powers for life? Please contact Larry LeBlanc or any other junta member listed below.

APL SIG STEERING COMMITTEE

Chairman: Larry LeBlanc
(RSTS) Teletype Corp.
2330 Eastern Ave.
Elk Grove, IL 60007
312-860-8181

Library
Coordinator: Susan Abercrombie
Ventrex Laboratories
217 Read Street
Portland, ME 04103
207-773-7231

Newsletter
Editor: Douglas Bohrer
(RT,TSX) Bohrer and Company
903 Ridge Road Suite 3
Wilmette, IL 60091
312-251-9449

Symposia
Coordinator: Bob Awde Jr.
(RSX) General Mills Inc.
9000 Plymouth Ave. North
Minneapolis, MN 55427
612-540-4432

Symposia
Assistant: Bob van Keuren
UserWare International, Inc.
2235 Meyers Ave.
Escondido, CA 92025
619-745-6006

Standards
Representative: Dan Esbensen
Touch Technologies
609 S. Escondido Blvd. Suite 101
Escondido, CA 92025
619-743-0494

VAX APL
Contact: OPEN

European
Contact: Jean-Pierre Barasz
(TOPS) BARTS
54 Rue d'Amsterdam
75008 Paris, France
33-1-633-3297

Digital
Counterpart: Dave Quigley
Digital Equipment Corp.
110 Spit Brook Rd ZK2-3/Q08
Nashua, NH 03062
603-881-2343

WRITE TO LONELY APLers

by Douglas Bohrer
Bohrer and Co.
903 Ridge Rd Suite 3
Wilmette, IL 60091

There are lots of lonely APLers in the world. They are lonely because there are so few of them in any one place. You can encourage these lonely special characters to continue to be as productive as the APL language is itself. You can especially encourage the editor of this publication. All you have to do is send me something to print.

Write an article on how your site uses APL or on how your site chose APL in the first place. Write an article on that handy function you just wrote. Write an article on how APL changed your whole life. Send me an article on an APL topic you already published somewhere else. Please, PLEASE, just give me something interesting on APL to print.

Suggested Format

Basically, it's helpful if your contribution looks like this article. Use black ink. Type your text single spaced. Use only one side of each sheet of paper. Skip one line between paragraphs. Set your margin at 4 3/8 inches wide. For type with 10 characters per inch, you would have 43 characters per line. For type with 12 characters per inch, you would have 52 characters.

Titles

Please start the title of your article on the left margin using all capital letters. Skip one line, then put your name, title, firm and mailing address. (You may omit your title, firm and mailing address for reasons of modesty, privacy, shame or whatever.) Sub-heads in your article should begin at the left margin with a blank line above and below them.

Letterhead will not be reproduced, unless you are volunteering for the steering committee. Drawings and listings should be either 4 3/8 or 9 inches wide.

Other Formats

I am able to read RT-11 floppies or any kind of magnetic tape. I am willing to retype things or use them as they are if you can't put your article into the suggested format. JUST GIVE ME SOMETHING TO PRINT!

Commercial Guidelines

The newsletter will follow strictly the guidelines related to the non-commercial nature of DECUS. If you have any questions on these, please contact the newsletter editor or the DECUS office.

Deadlines

An article that reaches the editor by the 15th of any month will go to press the following month. At least twice a year a "full issue" is prepared which has a "Local Variables" section of SIG news. The next "full issue" deadline is currently planned for January 15, 1986.

APL SIG FORMS "FREE WHITLOCK COMMITTEE"

Doug Bohrer
Editor, The Special Character Set

The APL SIG junta (..uh, I mean steering committee) thinks Stan Whitlock, DEC's APL Development Leader, may be under some kind of DEC house arrest. None of us has actually seen Stan since the Miami Symposium in 1981. While some APL greybeards think Stan died of food poisoning after eating the food in Miami, most of us think that the big guy is too tough to have let a little bad chicken spoil his afternoon. This situation has led your junta to form the "Free Stan Whitlock Committee of Correspondence".

Here's what you can do to help. Write a letter to the APL SIG Digital Counterpart, Dave Quigley, DEC, 110 Spit Brook Rd ZK2-3/Q08, Nashua, NH 03062. Or you can call Dave at 603-881-2343. Tell Dave you are interested in VAX APL V2, plan to be in Anaheim, and hope to see Stan there. Or, better yet, that you will be there only if you know Stan will be there. To preserve our credibility, make sure you really are interested and really will be there. In case Dave wants to check on your sincerity, please give him your name, address and phone.

Remember, you can help free Stan only if you express your interest. If you do nothing, we may never find out what's happened to Stan except from articles, like the ones in this issue, which he smuggles out to us. Please contact Dave Quigley. Free Stan Whitlock! Do it today.

ANAHEIM SYMPOSIUM HIGHLIGHTS

Susan M. Abercrombie
APL SIG (outgoing) Symposium Rep

The Anaheim Symposium in December should be an exciting one for APLers.

The VAX APL development group at DIGITAL has submitted two sessions. The first will cover the latest features in VAX-11 APL and DEC's plans for the product. The second session might best be described as "VAX talks to APL," covering the integration of APL into the environment available to more traditional languages, thus making system features available within APL.

A tutorial session will introduce APL to new users. Here's a chance for the curious to learn what's so special about those characters.

User papers will be presented which discuss applications in APL on both VMS and TOPS-20 systems.

Mark Brittinham of the University of Delaware will describe the use of APL in modeling artificial intelligence. Specific examples will be a word perception model and a natural language parser.

SESSION TO ANNOUNCE VAX APL V2.0

Douglas Bohrer
Editor, The Special Character Set

Digital Equipment Corporation has asked to schedule a session for the DECUS Symposium in Anaheim which is titled "ANNOUNCING VAX APL V2.0". Features likely to be included in the release are described in Stan Whitlock's article later on in this issue. I have printed the entire session description below.

"VAX APL V2 represents an important advance in performance and feature levels for the product. A representative from Digital will present an overview of the highlights of V2. This session will also provide a forum for discussion with a member of the development team and plans for the product."

[Editor's note: Long time readers of this newsletter will realize that DEC has not always been so clear about its product intentions as they are now. The new candor is so different from previous discussions that it is interesting in itself. DEC must really be interested in APL to change so much.

For those of you for whom change is coming too fast, those who yearn for the more romantic times of yesteryear, and those who learned to love mystery and intrigue during former days, I have reformulated the article above as it would have appeared three years ago.]

VAX APL V2 DISCUSSED OVER PIZZA

Doug Bohrer
Editor, The Special Character Set

A usually reliable source in New Hampshire has told the staff of The Special Character Set that he has been hearing lots of discussion of a VAX APL V2 in Nashua pizza parlors lately. I showed him an advance copy of Stan Whitlock's article, "VAX APL OVERVIEW", included in this issue. "Ya! They've been talking about all dis [sic] stuff over pizza," he exclaimed excitedly. My source thinks that a product announcement for VAX APL V2 is likely in Anaheim based on his past experience with DEC and pizza.

Jean-Pierre Barasz of Barts (Paris) will present two sessions. One which might be referred to as "APL talks to TOPS-20" will describe an extended APL which allows APL programs to create and communicate with subtasks. The second will describe a point of sale transaction and authorization system implemented in APL.

Come to our sessions to discover the present status and future plans for VAX-11 APL, to learn about the general features of APL and some of the applicaitons for which it is particularly suited, and to hear about some substantial projects implemented in APL.

See you in Anaheim!

USING APL AS A DESIGN AND DECISION TOOL

Peter Haynes
Digital Equipment Corp.

This function began life last spring when I was looking for a new bicycle, but I think it illustrates well the use of APL as a Laboratory.

The basic problem was to select a set of gears, chainwheels for the front and freewheels for the back which would be good for general riding around town as well as for week long tours. I began by writing a simple function which calculates the gear ratio for each combination of front and rear gears on the bikes I was considering.

This function, GEARS, however only presented a table of the ratios from which it is difficult to draw any conclusions. So my next step was to evolve the main function GEARINFO which presents the ratio information in several graphical as well as numeric forms. This first graph shows how the front and rear gear combinations relate to each other.

This still was not quite what I wanted. What I wanted was to see the total picture of how each of the ratios related to each other in the sequence from the lowest to the highest. To do this I simply (for APL) sorted the ratios and their corresponding combinations and drew the graph again.

Now I was getting somewhere. I could now vary the chainwheels and freewheels and see the affect on the sequence of the ratios. I then added bells and whistles like titles and other useful calculations.

With this little function I was able to try many combinations until I found one which satisfied my needs. There are of course many other features which could easily be added to the function to adapt it to suit some other design question. But as is it enabled me to choose a set of gears which have kept me happy.

Chart 1 shows the graphs for the gears as supplied by the manufacturer. Chart 2 shows the gearing I finally settled on. Notice how the curve in the second graph is smoother than the one in chart 1.

APL BUTTON SLOGAN CONTEST

Doug Bohrer
Editor, The Special Character Set

Every six months the APL SIG tries to come up with a new slogan for buttons we pass out at symposia. Past slogans have included "APL SPECIAL CHARACTER" and "APLers DO IT WITH SPECIAL CHARACTERS". Some observers have noticed that the buttons, taken together, indicate a happy state of affairs among APL users. However, we must come up with a new slogan for Anaheim. So we decided on a contest. The prize is that we'll send you 10 buttons with your slogan on them, airmail for overseas winners, no matter how remote you are from the rest of the world. You can be working for anybody, using whatever equipment you like, and you need not ever go to a symposium, or anywhere else, to win.

The rules are simple. Your junta will decide who wins based on whatever hits their fancy the best. The entries must be in a language we can read and translate. English is a good choice; French and Spanish are acceptable; COBOL is unreadable even if structured. Duplicate entries may or may not get duplicate prizes, depending on how much budget we have left after printing the buttons. For sure, the first person to send in the winning slogan will get 10 buttons. We can't use any slogan not in the public domain, more or less. It helps if the slogan is understandable by non-APLers and stresses some APL feature or advantage.

To be useful, entries must reach me by 1 November 1985 so I have time to get the buttons printed for the Anaheim Symposium. Send your entry to:

Doug Bohrer
Bohrer and Company
903 Ridge Rd, Suite 3
Wilmette, IL 60091 USA

The only entry so far is "APL CHARACTERS: EFFECTIVE WITH FEWER STROKES". If we get no other entries we may have to go with this.

THE SPECIAL CHARACTER SET September 1, 1985 No.10

It is assumed that all articles submitted to the editor of The Special Character Set or his representatives are with the author's permission to publish in any DECUS publication and that the author has the right to grant such permission. These articles are the responsibility of the authors and, therefore, the editor assumes no responsibility or liability for articles or information appearing in The Special Character Set. The views expressed are those of the authors and do not necessarily express the views of DECUS, the APL Special Interest Group, the Digital Equipment Corporation or the editor.

The Special Character Set is the official publication of the APL Special Interest Group of the Digital Equipment Computer Users Society. Unsolicited manuscripts are desperately sought by the editor.

VCW GEARINFO FW ;R ;K ;T ;IO ;PW

- [1] $\square IO+0 \diamond \square PW+132$
- [2] **T+CW GEARS FW**
- [3] $R+T[0; ;]$
- [4] $T+T[1; ;]$
- [5] $\square CTRL[13 10]$
- [6] $(\uparrow(\rho CW) \times \rho FW), ' SPEEDS', \square CTRL[13 10]$
- [7] $K+\uparrow \rho FW$
- [8] $(' X3, ', K, ' I6') \square FMT(1, \rho FW) \rho FW$
- [9] $(' I2, " ", ', K, ' G' ZZ9.9"') \square FMT(CW; 10 \times R)$
- [10] $\square CTRL[13 10]$
- [11] $' X5, 120G' Z' ' \square FMT 10 10 10 \tau \epsilon 120$
- [12] $' G' 99-99", 120A1' \square FMT(, T; '* '[(\{0.5+R\} \circ . \neq \epsilon 120))$
- [13] $\square CTRL[13 10]$
- [14] $K+\Delta, R$
- [15] $R+(, R)[K]$
- [16] $T+(, T)[K]$
- [17] $' X5, 120G' Z' ' \square FMT 10 10 10 \tau \epsilon 120$
- [18] $' G' 99-99", 120A1' \square FMT(T; '* '[(\{0.5+R\} \circ . \neq \epsilon 120))$
- [19] $\square CTRL[13 10]$
- [20] **CW CAPACITY FW**

▽

VZ+CW GEARS FW

- [1] $Z+27 \times CW \circ . \div FW$
- [2] $Z+Z, [\square IO-0.5](100 \times CW) \circ . +FW$

▽

VCW CAPACITY FW

- [1] **'FRONT DERAILLEUR RANGE: ', $\uparrow, \uparrow(\uparrow / CW) - \lfloor / CW$**
- [2] **'REAR DERAILLEUR RANGE : ', $\uparrow, \uparrow((\uparrow / CW) - \lfloor / CW) + (\uparrow / FW) - \lfloor / FW$**

▽

THE SPECIAL CHARACTER SET

FIRST TIME FEAR

Douglas Bohrer
Bohrer and Company
903 Ridge Road, Suite 3
Wilmette, IL 60091

Most of us fear the first time we do anything. We fear failure and embarrassment. If failure looks likely, we may be unwilling to take the risk of trying unless the rewards look great. For many, trying APL involves a lot of first time fear. The strange characters make failure look likely. Since APL is only one computer language among many, the rewards don't look outstanding.

My first time with APL wasn't very frightening. It was 1968. I was a high school student in suburban Chicago. My teacher in Advanced Physics recruited me to try an experiment in education. A subsidiary of IBM was trying the then novel idea of teaching computer programming to high school students. The language for this experiment was APL.

While computers were interesting, I had never considered a career in them. I had very little at stake so I wasn't too worried. I found APL to be easy to learn. How can you miss when 2+2 carriage return is all you need to get started? I didn't know how to type then, so I found using funny characters just as easy as using "normal" ones. Since we were using 10 character per second dial up lines, I found that using a small amount of characters to get a large amount of work was convenient. The syntax rules seemed simple compared to physics equations. Compared to my first dancing lesson at age 12 in the Presbyterian Church gymnasium, this was really easy.

Whatever you learn early, you assume is normal. When I was 9 years old, I assumed that since my grandmother slept with a shotgun on the nightstand to shoot porcupines eating the roses, that this was normal for grandmothers. At 17, I also assumed that the elegant efficiency of APL was normal for computer languages. It didn't take me too long in either case to find out how wrong I was.

In college, I learned FORTRAN and PL/I. What a pain! In APL, I wrote A+B and no matter what shape A and B were, every element of A was added to the corresponding element of B. Now I needed double indexing and nested loops to do the same thing. Worse, I had to decide in advance not only what size and shape A and B were, but also what storage type. And if I wanted to do the same thing to several different storage types, I had to write one routine for each type. It seemed that APL had done lots of work for me that I now had to do for myself. The wait for results from a compile and run was also a new experience. Instant responses no longer followed the carriage returns.

I thought I had seen the worst. Then I got to the Air Force and vacuum tube machines. I learned JOVIAL, an ALGOL with a dollar sign as the statement terminator. Compile waits lengthened to hours. Waiting to get at the machine sometimes took a day or two. I learned assembly language for the machine. It may have been lots quicker for the machine, but it was a lot slower for me.

After the Air Force and graduate school, I went to work for a large Chicago bank. I finally got to choose for myself the language I would work with. I chose APL. I was tired of doing all this extra work that I knew the machine could be doing instead.

That's when I really learned about first time fear: other people's first time fear. "What in the Hell is that stuff? It sure looks funny. Why can't you use a 'normal' character set like everybody else?"

First time fear is hard to reduce. I did the wage/price guideline calculations, including all the programming to track the history we didn't keep on the payroll files, in less than two weeks. The fear was still there, but more in the Biblical sense of respect than before. Now I was the magician who used incomprehensible symbols to get the work done.

I've been working in Whitesmiths C for the last two years, writing an APL interpreter. I'm almost done. I can't wait. I want to return to the quick rewards of APL. There is just no comparison. What would take a day in APL takes a week in C.

My experiences bring me to the central question facing APL today. How can the APL community convince people that despite its strange appearance, APL is both easy to learn and well worth the effort?

The first time fear must be lowered. We must stress that almost anybody can learn enough APL to use it almost immediately. We must stress that you don't need to know everything there is to know about APL to get some good out of it. You ease into it, learning what you need by interacting with the interpreter as you get the work done.

We APLers must also reduce our threatening attitude on what is good APL style. Bashing people for using loops creates fear in people who have always used them. A good APL program is one that does what you want it to do. Quick coding is the APL advantage. Avoiding loops is only a usually useful way of coding APL quickly, not an end in itself.

The APL community also has to stress the rewards. Using APL is fun. Using the more traditional languages is more like work.

My grandmother owned a pink '59 Thunderbird which everybody wanted to drive or ride in. We used to zoom down Montana dirt roads at over 50 miles an hour. It was a blast! We also had a '48 Dodge Powerwagon on the ranch. It was painted Army green and had 4 wheel drive which took it anywhere. If you had to go down logging roads, you took the truck. Otherwise, you tried to talk grandma into using the car.

It's the same question in choosing a programming language. If you're doing real time, communications protocols, interpreters or other rough stuff use assembler, C, JOVIAL, or whatever will get you where you want to go. But if you're doing normal applications, zoom down the road using APL. Have a good time and get where you have to go faster.

(c) Copyright 1985 by Douglas Bohrer. Used with permission.



General Mills, Inc.
James Ford Bell Technical Center

9000 Plymouth Avenue North
Minneapolis, Minnesota 55427

AWD/17/85X

June 18, 1985

Mr. Doug Bohrer
Bohrer and Company
903 Ridge Road, Suite 3
Wilmette, Illinois 60091

Dear Doug:

This letter confirms my desire to join the APL SIG Steering Committee as the Symposia Coordinator. Please advise me in writing of the responsibilities of the Symposia Coordinator and any other information regarding the APL SIG Steering Committee.

Also enclosed on Floppy Disk in RT-11 format is a copy of the presentation on the I/D Space APL Interpreter I delivered at the New Orleans National DECUS Symposium. Feel free to publish it in "The Special Character Set".

I certainly enjoyed meeting you at New Orleans and look forward to working with you on the APL SIG Steering Committee in the future.

Sincerely,

Bob Awde, Jr.
Engineering Section Leader

BA/mw

NEW APL SYMPOSIUM REP

Doug Bohrer
Editor, The Special Character Set

The APL SIG has signed a new steering committee member, Bob Awde, as our new Symposium Coordinator. Bob's official volunteer letter appears above. Bob Awde will be assisted by Bob van Keuren. The two Bobs will take some of the load off of Sue Abercrombie, who will remain our Library Representative. The fact that it takes two guys to do part of Sue's former chores tells you just how hard Sue has been working for the SIG.

Bob Awde's slides on how to covert APL-11 to use IND space under RSX appear elsewhere in this issue. His article on how to make APL characters appear on VT100 and VT200 terminals appeared in issue 9 of The Special Character Set.

THE SPECIAL CHARACTER SET

DESIGNING CHARACTER SETS FOR THE VT200

Stan Whitlock
 APL Development Project Leader
 DEC ZK02-3/N30
 110 Spit Brook Road
 Nashua, NH 03062

In the April, 1985, volume of "The Special Character Set", Bob Awde described COMPOSE, a VT200 custom character set generator program. There has been some interest in creating an APL character set for the VT200, both inside and outside of DEC. Unfortunately the VT200 has only one 96-character loadable character set (DRCS) and the VAX APL character set contains 176 unique printing characters. The VT200 series terminals are not capable of displaying overstrikes and, to add insult to injury, VMS v4 has used the BACKSPACE control character on input for its terminal line-editing feature to move the cursor to the beginning of the line.

Even with all of these obstacles, I have been sent several attempts at creating an APL character set for the VT200. It usually arrives via computer mail as a set of SIXEL sequences used to load the DRCS. I needed to analyze these different character sets, compare them, and display them in some fashion. I also wanted to create overstrikes from these characters to see how well the individual characters overlaid each other.

A brief digression about SIXEL patterns for the VT200 is needed. The following picture is the format of a character cell:

```

1 2 3 4 5 6 7
x x x x x x x
x x x x x x x
x x x x x x x
x x x x x x x
x x x x x x x
x x x x x x x
Y Y Y Y Y Y Y
Y Y Y Y Y Y Y
Y Y Y Y Y Y Y
Y Y Y Y Y Y Y
  
```

This makes up the 7x10 character cell dots. That's 10 rows and 7 columns - I have no idea why they pronounce the shape backwards. The columns of x's make up the first part of the sixel S...S bytes and the columns of y's make up the second set of sixel bytes /S...S. These are computed by bit values of the columns and adding 77 (octal) to make it printable. This makes an ASCII byte to put into the sixel string. This top bit is least significant. A letter "A" could look like:

```

1 2 3 4 5 6 7
. . . . . . .
. . . x . . .
. . x . x . .
. x . . . x .
x . . . . . x
x x x x x x x
y . . . . . y
Y . . . . . Y
. . . . . . .
. . . . . . .
  
```

The first column of x's would be computed as 110000 (binary) = 60 (octal) + 77 (octal) = 157 (octal) = "o" (ASCII). The second column of x's would be 101000 (binary) = 50 (octal) + 77 (octal) = 147 (octal) = "g", etc. The first column of y's would be computed as 0011 (binary) = 3 + 77 (octal) = 102 (octal) = "B". No bits on is a 77 (octal) which would be a "?". This "A" would be specified in the sixel string:

ogcacgo/B?????B

I didn't have alot of time to play around with this stuff so I started creating some APL functions to help me understand sixels and character cells. (All of the functions mentioned are listed in alphabetical order at the end of the article along with some example output.) All of this works in VAX APL, even though I used some features that will become available in VAX APL v2.0.

Since all of the character sets I wanted to analyze were in sixel format, I wanted to read in these patterns as ASCII text into APL. This is easy to do using pure data input on a /IS file. `AV` in VAX APL contains all of the ASCII characters, lower case letters included, but pure data mode 6 does some strange translations to look like TTY mnemonics (eg, ASCII underscore is translated to APL's left-arrow). Fortunately, a new pure data mode, 11, was available to me which keeps ASCII characters as ASCII (eg, ASCII underscore becomes APL's underscore).

An APL character set file looks like

```

${62;1"p
$P;1;1;4{X
CAC?CAC/????????;
...
GSS}SS/?@B@?;
$
$)x
  
```

which contains escape sequences to load and enable the DRCS. All of those non-sixel lines have to be ignored. The sixel patterns are in order from the character that matches octal 041 through octal 176. For APL that's " through \$.


```

V OVFONT ← BLD OV FONT ; IO; OAV; AV; N; TAB; C1; C2; C3; P1; P2; P3; S; I
[1] IO ← 1
[2] OAV ← ((AV + 128 ↓ AV) COQ 0 13) CIQ 0 5
[3] N ← ρ OAV + (2 + 1 ↑ OM ('', CTRL[9], '')) SS OAV ↑ OAV
[4] OAV ← ((N ÷ 3), 2) ρ (N ρ 1 0 1) / OAV
[5] TAB ← CTRL[10]
[6] OVFONT ← 0 15 ρ ''
[7] I ← 1
[8] L: C1 ← FONT DISP OAV[I; 1] ◊ C2 ← FONT DISP OAV[I; 2]
[9] P1 ← BITS C1 ◊ P2 ← BITS C2
[10] P3 ← P1 ∨ P2 ◊ C3 ← BOX ' *'[1 + P3]
[11] S ← 1 15 ρ SIXEL C3 ◊ OVFONT ← OVFONT ⋮ S
[12] '3(9A1, X2), 15A1, X2, 5A1' FMT (C1; C2; C3; S; OAV[I; 1]; TAB; OAV[I; 2]; TAB; AV[I])
[13] +(0 ≠ 4 | I) / LC + 1 ◊ + CTRL[13]
[14] + ''
[15] + ((ρ OAV) ≥ I + I + 1) / L
V

```

```

V Z ← {D} BOX C; SIDE; TOP
[1] ⍎ SURROUND CHAR MATRIX C WITH BOX OF CHARS D
[2] +(0 ≠ NC 'D') / LC + 1 ◊ D ← ''
[3] SIDE ← (1 ↑ ρ C) ρ D
[4] Z ← SIDE, C, SIDE
[5] TOP ← '+', ((¬ 1 ↑ ρ C) ρ D), '+'
[6] Z ← TOP ⋮ Z ⋮ TOP
V

```

```

V FONT1 CMPALL FONT2 ; A; T1; T2; I; SS; M; IO
[1] ⍎ FONTN IS IN SIXELS STARTING WITH ''
[2] IO ← 1
[3] +((ρ FONT1) = ρ FONT2) / LC + 1 ◊ BREAK 'FONT LENGTH ERROR'
[4] A ← (1 ↑ ρ FONT1) ↑ 33 ↓ AV
[5] I ← 1
[6] L: T1 ← FONT1 DISP A[I]
[7] T1 ← T1[1 + ⌊ 10; 1 + ⌊ 7]
[8] T2 ← FONT2 DISP A[I]
[9] T2 ← T2[1 + ⌊ 10; 1 + ⌊ 7]
[10] SS ← (FONT1[I; ⌊ 7], '/', FONT1[I; 7 + ⌊ 7]), [0.5] FONT2[I; ⌊ 7], '/', FONT2[I; 7 + ⌊ 7]
[11] M ← 'NY'[1 + SS[1;] = SS[2;]]
[12] 'A1, X2, 2(9A1, X2), 15A1, X2, A1' FMT (M; BOX T1; BOX T2; ; SS; A[I])
[13] +(0 ≠ 4 | I) / LC + 1 ◊ + CTRL[13]
[14] + ''
[15] + ((ρ A) ≥ I + I + 1) / L
V

```

THE SPECIAL CHARACTER SET

```

V Z+{FONT} DISP CAV ; C; CN; CB; TOP; BOT; I; CHAR
[1]  ⑆          DISPLAY CHAR CAV FROM FONT
[2]  ⑆          CAV IS EITHER A SCALAR CHAR (LOOKED UP IN FONT) OR A SIXEL STRING
[3]  +(0≠⊠NC 'FONT')/⊠LC+1  Ⓞ  FONT+CFONT          ⑆  DEFAULT FONT
[4]  C+CAV
[5]  +(0≠ρρC)/L          ⑆  CHAR OR SIXEL ?
[6]  I+⊠AV  Ⓛ  CAV          ⑆  LOOK UP CHAR
[7]  C+FONT[I-33;]
[8]  L:CN+(ALPHABET  Ⓛ  C) - 31
[9]  CB+(8ρ2)+CN
[10] TOP+~6 7↑CB
[11] BOT+~4 ~7↑CB
[12] CHAR+(ⓄTOP)~ ⓄBOT
[13] Z+BOX ' *'[CHAR+1]
V

```

```

V DISPALL FONT ; A
[1]  ⑆          DISPLAY ALL ELEMENTS OF SIXEL FONT
[2]  A+(1↑ρFONT)↑33↓⊠AV
[3]  L:9ρ'-' ; ' CHAR: ' ; A[1]
[4]  FONT DISP A[1]
[5]  +(0<ρA+1↓A)/L
V

```

```

V EDITCHAR ; T; C
[1]  ⑆          CALL TPU TO CREATE/MODIFY A CHARACTER PICTURE
[2]  ⑆          FILL IN 10 BY 7 CHAR CELL WITH *
[3]  ⑆          RESULT IS IN NEW_<NAME GIVEN AS INPUT>
[4]  T+ρ⊠+'ENTER CHARACTER NAME: '  Ⓞ  C+T↓⊠
[5]  +(0≠⊠NC C)/⊠LC+1  Ⓞ  T+10 7ρ' '  Ⓞ  Ⓛ  C,'+T'
[6]  T+BOX Ⓛ  C
[7]  T[1+Ⓛ10;9]+' '
[8]  G_NEW + T
[9]  T+ε ' )EDIT G_NEW'
[10] T+G_NEW  Ⓞ  ⊠SINK+⊠EX 'G_NEW'
[11] T+T[1+Ⓛ10;1+Ⓛ7]
[12] Ⓛ  'NEW_',C,'+T'
V

```

THE SPECIAL CHARACTER SET

```

V FONT ← READFONT FILE ; R
[1]  ⚪SINK+⚪ASS '1 ' ,FILE, '/ IS'
[2]  FONT←O 14 ρ' '
[3]  B:R+⚪1 11
[4]  +(2=ρρR)/ERR
[5]  +(⚪CTRL[28]≠1↑R)/B
[6]  +('P1;'Λ.=3↑1↓R)/L
[7]  +('P0;'Λ.=3↑1↓R)/L
[8]  +('P;'Λ.=2↑1↓R)/L
[9]  + B
[10] ERR:⚪BREAK 'NO SIXELS FOUND IN ' ,FILE
[11] L:R+⚪1 11
[12] +(2=ρρR)/DONE
[13] +(⚪CTRL[28]=1↑R)/DONE
[14] R+(7↑R),7↑8↓R
[15] FONT←FONT ; R
[16] → L
[17] DONE:⚪DAS 1
V

```

```

VZ←SIXEL C ; T; TOP; BOT
[1]  ⚪          TURN CHAR PICTURE OF *'S INTO SIXEL ENCODING
[2]  ⚪(2≠ρρC)/'⚪BREAK ''NOT MATRIX''
[3]  +(10 7Λ.=ρC)/⚪LC+1  ⚪ C+C[1+⚪10;1+⚪7]
[4]  T←C='*'
[5]  TOP←2⚪6 7↑T
[6]  BOT←2⚪6 0↓T
[7]  Z←ALPHABET[ TOP+31], '/', ALPHABET[ BOT+31]
V

```

THE SPECIAL CHARACTER SET

Here is an example of the output from BLDOV, building the overstrikes for `Ⓛ` and `Ⓜ`:

```

+.....+ +.....+ +.....+ ?OgwgO?/AAABAAA Ⓛ Ⓜ
.      .      .      .
.      .      .      .
.  *    .    *    .    *    .
.  *    .    *    .    *    .
.  *    .    *    .    *    .
.  *    .    *    .    *    .
.*****. .      . .*****.
.      .      .      .
+.....+ +.....+ +.....+
+.....+ +.....+ +.....+ GWgwgWG/???B??? Ⓛ Ⓜ
.      .      .      .
.      .      .      .
.*****. .      . .*****.
.  *    .    *    .    *    .
.  *    .    *    .    *    .
.  *    .    *    .    *    .
.  *    .    *    .    *    .
.      .      .      .
.      .      .      .
+.....+ +.....+ +.....+

```

Here is an example of the output from CMPALL, comparing two VT200 APL character sets on the two characters `Ⓜ` and `)`:

```

N +.....+ +.....+ CAC?CAC/??????? Ⓜ
.  *    .    *    .    *    .
.  *    .    *    .    *    .
.      .      .      .
.      .      .      .
.      .      .      .
.      .      .      .
.      .      .      .
.      .      .      .
+.....+ +.....+
N +.....+ +.....+ ??ACw??/?A@???? )
.  *    .    *    .    *    .
.  *    .    *    .    *    .
.  *    .    *    .    *    .
.  *    .    *    .    *    .
.  *    .    *    .    *    .
.  *    .    *    .    *    .
.  *    .    *    .    *    .
.      .      .      .
+.....+ +.....+

```

THE SPECIAL CHARACTER SET

VAX APL OVERVIEW FROM SPRING '85 DECUS

Stan Whitlock
APL Development Project Leader
DEC ZKO2-3/N30
110 Spit Brook Road
Nashua, NH 03062

The following article summarizes the "VAX APL Overview" session that was presented by DEC at Spring '85 DECUS in New Orleans. That session covered future possibilities for the VAX APL product as well as showed examples of some of the unique features that are already available in VAX APL v1.3.

Spring '85 DECUS Slides

VAX APL

Current Status

- o VAX APL v1.3-332 currently in the field
 - v1.3 is a strict maintenance release
 - bug fixes only
 - support for RX50 media for microVAX I
- o VAX APL v1.2 contained performance improvements
- o VAX APL v1.1 contained new functionality

VAX APL

Future Possibilities

- o VAX APL v2.0 currently under development
 - support for full screen editing
 - more global performance improvements
 - run-time (execute-only) APL support
 - on-line HELP for APL language features

VAX APL

Future Possibilities (cont)

- calling out to external routines
 - * access VMS product set, eg, Rdb, SORT
 - * call RTL routines, eg, MTH\$, SMG\$
 - * prototype and debug algorithm in APL;
compile performance-critical sections

THE SPECIAL CHARACTER

VAX APL

Future Possibilities (cont)

- an external Fortran routine

```
COMPLEX*16 FUNCTION CSUM (C, N)
INTEGER N
COMPLEX*16 C(N)
CSUM = 0
DO I = 1, N
  CSUM = CSUM + SQRT (C (I))
END DO
END
```

- create a shared image

```
$LINK/SHAR=f.exe f.obj,SYS$INPUT:/OPTIONS
UNIVERSAL=CSUM
^Z
```

VAX APL

Future Possibilities (cont)

- defining an external routine

```
A + 'z/TYP:DC+ f a/TYP:DC/MECH:REF b/TYP:L/MECH:REF'
B + 'wrk:f/ENTRY:csum'
A MAP B
```

- invoking an external routine

```
F ( (,10),[1.5] 0 ; 10)
22.46827819 0
```

THE SPECIAL CHARACTER SET

VAX APL

Future Possibilities (cont)

- dyadic grade
 - `MONITOR` for performance coverage analysis
 - `WATCH` to find when a variable is changed
 - `)STEP` to single-step and trace execution
 - wild cards on `)VARS` etc and `)COPY` etc
-

VAX APL

Future Possibilities (cont)

- `OM A ↔ A / ρ A` `APLSF ω`
 - `)XLOAD` to load without `LX`
 - `SS` string search
 - `,` insertion in `v` editing
 - `$` search/replace in `v` editing
-

THE SPECIAL CHARACTER SET

VAX APL

Future Possibilities (cont)

- more RMS features
 - * multi-key ISAM with character keys
 - * fixed length records and stream files
 - * /DISPOSE = PRINT and SUBMIT
- set functions
 - * union, intersection, difference
 - * subset, contains, remove duplicates
 - * match = (= overstruck with underscore)

VAX APL

Unique Features

- o replicate - an extension of compress

1 1 2 1 2 1 2 1 / 'MISISIPI'
MISSISSIPPI

TEXT
HERE IS THE FIRST LINE OF TEXT
HERE IS A LINE WE DO NOT WANT
HERE IS THE SECOND LINE OF TEXT

1 -1 0 1 / TEXT
HERE IS THE FIRST LINE OF TEXT
HERE IS THE SECOND LINE OF TEXT

THE SPECIAL CHARACTER SET

VAX APL

Unique Features (cont)

- o local functions stored on files

```

V CHAN PUT_DEF FUNCTION ; IO
[1] IO + 1  (CR FUNCTION) CHAN
[2] FUNCTION, '_DEF + ', CHAN, (FLS CHAN)[2]
V

```

```

V Z + A F B ; F
[1] SINK + FX [+ [1 F_DEF] 1 F_DEF
[2] Z + A F B
V

```

VAX APL

Unique Features (cont)

- o groups of data established when needed

```

V Z + A F B
[1] SINK + QCO 'LIB_WS GRP_F'
[2] Z + A REAL_F B  ALL NECESSARY DATA PRESENT
[3] SINK + ε ') ERASE GRP_F'
V

```

- o NG affects + and FI and VI

```

NG + 1  ( -\t5
1 -1 2 -2 3
NG + 0  ( -\t5
1 -1 2 -2 3

NG + 1  ( (VI X)/FI x + '1 A 2 -4 3'
1 2 3
NG + 0  ( (VI X)/FI x + '1 A 2 -4 3'
1 2 -4 3

```

THE SPECIAL CHARACTER SET

VAX APL

Unique Features (cont)

- o `⊞SF` - set `⊞` input prompt

```

⊞SF ← 'ENTER NUMBER: ' ⊞ N ← ⊞
ENTER NUMBER: +/2 4 5 1
      N
12
    
```

- o `⊞FMT` reporting

```

      HEAD ; ROWS , 'F9.2' ⊞FMT (DATA1;DATA2;DATA1+DATA2)

TEST          SLOW      FAST      SLOW
CASES         DATA     DATA     -FAST

+ / BOOLVEC   25.20     5.20     4.85
CHARMAT^.=CHARVEC 94.50     6.90     13.70
CHARMAT^.#CHARVEC 89.90     7.60     11.83
v\ BOOLVEC    94.50     2.50     37.80
    
```

VAX APL

Unique Features (cont)

- o Spawning Subprocesses

```

V EDIT FUNCTION ; ⊞PW ; I ; PW
[1] PW ← ⊞PW ⊞ ⊞PW + 200
[2] I ← (⊞NUM, 'ABCDEF') [⊞IO+(8ρ16)τ⊞UL]
[3] ⊞SINK ← ε ') OUTPUT ', I + 'T', I, '.AAS'
[4] ⊞VR FUNCTION
[5] 1 ⊞ ⊞SINK ← ε ') OUTPUT'
[6] ⊞PW ← PW
[7] ⊞SINK ← ε ') PUSH EDIT/EDT ', I
[8] ⊞SINK ← ⊞EX FUNCTION
[9] ⊞SINK ← ε ') INPUT ', I
[10] I ← ⊞
[11] ⊞SINK ← ε ') INPUT/REVERT'
V
    
```

THE SPECIAL CHARACTER SET

VAX APL

Unique Features (cont)

o Escape sequences on the VT100 terminal

▽ *ATTRIB B ; T*
[1] *␣CHARACTER ATTRIBUTES == \$ [PS ; PS ; ... PS M*
[2] *␣ 0 == OFF ; 1 == BOLD ; 4 == UNDERSCORE*
[3] *␣ 5 == BLINK ; 7 == REVERSE VIDEO*
[4] *T + $\bar{1}\downarrow, ((\rho B), 1)\rho 48+, 10 \tau B), 59$*
[5] *□ARBOU 27 91, T, 109*

▽

▽ *CLEAR*
[1] *␣ERASE FROM CURSOR TO END OF SCREEN == \$ [O J*
[2] *□ARBOU 27 91 48 74*

▽

▽ *DHBOTTOM*
[1] *␣DOUBLE HEIGHT BOTTOM HALF == \$ # 4*
[2] *□ARBOU 27 35 52*

▽

▽ *DHTOP*
[1] *␣DOUBLE HEIGHT TOP HALF == \$ # 3*
[2] *□ARBOU 27 35 51*

▽

▽ *DWSH*
[1] *␣DOUBLE WIDTH SINGLE HEIGHT == \$ # 6*
[2] *□ARBOU 27 35 54*

▽

▽ *ERASEL*
[1] *␣ERASE ENTIRE LINE CONTAINING CURSOR == \$ [2 K*
[2] *□ARBOU 27 91 50 75*

▽

▽ *HOME*
[1] *␣HOME CURSOR == \$ [H*
[2] *□ARBOU 27 91 72*

▽

▽ *POSITION P;L;C*
[1] *␣POSITION CURSOR == \$ [PL ; PC H*
[2] *L+1↑P ◊ C+1↑1↓P*
[3] *L+48 + ,10 10 τ L ◊ C+48 + ,10 10 τ C*
[4] *□ARBOU 27 91, L, 59, C, 72*

▽

THE SPECIAL CHARACTER SET

```
▽ REGION P;T;B
[1]  ⚠°DEFINE REGION == $ [ PT ; PB R
[2]  T+1↑P ◊ B+1↑1↓P
[3]  T+48 + ,10 10 τ T ◊ B+48 + ,10 10 τ B
[4]  □ARBOU 27 91, T, 59, B, 114
▽

▽ RESTORE
[1]  ⚠°RESTORE CURSOR AND ATTRIBUTES == $ 8
[2]  □ARBOU 27 56
▽

▽ SAVE
[1]  ⚠°SAVE CURSOR AND ATTRIBUTES == $ 7
[2]  □ARBOU 27 55
▽

▽ T BANNER P
[1]  SAVE
[2]  POSITION P
[3]  ERASEL ◊ DWSH ◊ DHTOP ◊ ATTRIB 1 5 ◊ □+T
[4]  ERASEL ◊ DWSH ◊ DHBOTTOM ◊ □+T
[5]  RESTORE
▽

▽ CLEANUP
[1]  REGION 1 24 ◊ ATTRIB 0 ◊ HOME ◊ CLEAR
▽

▽ T FLASH P
[1]  SAVE
[2]  POSITION P ◊ ERASEL
[3]  ATTRIB 1 5
[4]  □+T
[5]  RESTORE
▽

▽ SETUP
[1]  HOME ◊ CLEAR
[2]  REGION 22 23
[3]  POSITION 22 0
▽
```

THE SPECIAL CHARACTER SET

VAX APL

Unique Features (cont)

o Mailboxes and Event Flags

- example where EF = 0 if mailbox is empty
EF = 1 if message is waiting

CHAN + □ASS 'MAILBOX/AS/MBX/SHARE/EFN=64'

▽ BUSY + DATA MBXOUT CHAN
[1] + (BUSY + □EFS CHAN)/0
[2] DATA □[4]CHAN □ WAITS UNTIL MESSAGE IS READ
▽

▽ INPUT + MBXIN CHAN
[1] INPUT + 0 75ρ0 ◊ + (0=□EFC CHAN)/0
[2] INPUT + □[4]CHAN
▽

VAX APL

Unique Features (cont)

o Pure Data I/O and Packed Data

- Fortran file containing unformatted F-floating data:

□ ISFILE, 3 □ read a record

NUMBERS □ ISFILE, 3 □ write a record

- Fortran record with 2 F-floating numbers and 3 bytes

RECORD + 2 ↓ □ ISFILE, 7 □ read bytes

FLOATING + ((8↑RECORD) □coq 0 7) □ciq 0 3

INTEGERS + 8↓RECORD □ RECORD is integer

THE SPECIAL CHARACTER SET

VAX APL

Unique Features (cont)

o Error Handling

- catching local errors

`⊞FX sets ⊞ERROR`

`+ (2=⊞⊞INPUT + ⊞[2]ASFILE)/EOF`

`+ (2=⊞⊞RESULT + ⊞ε')LIB')/NOQUOTA`

- trapping and signalling errors

`⊞TRAP + '→ TRAP_FUNC ⊞ ⊞SIGNAL 500'`

`∇ Z + TRAP_FUNC ; ⊞TRAP ; T ; E`

[1] `T + ⊞TRAP ⊞ ⊞TRAP + ''`

[2] `E + ⊞FI 4 ↑ ⊞ERROR`

[3] `+ (500 ≠ E)/⊞LC+1 ⊞ ⊞BREAK 'USER SIGNAL'`

[4] `Z + DOMERR_LABEL ⊞ + (15=E)/0`

[5] `Z + ε0`

`∇`

THE SPECIAL CHARACTER SET

DEC USER SOCIETY (DECUS) SYMPOSIUM PRESENTATION

I/D SPACE APL LANGUAGE INTERPRETER ON RSX-11M-PLUS

1985 SPRING DECUS U.S. SYMPOSIUM
NEW ORLEANS, LOUISIANA
MAY 27-31, 1985
BY BOB AWDE JR.
GENERAL MILLS, INC.
MINNEAPOLIS, MINNESOTA

PRESENTATION OUTLINE

1. PROBLEM STATEMENT AND SOLUTION
2. HARDWARE AND SOFTWARE REQUIREMENTS
3. DEFINITION OF A I/D SPACE TASK
4. DEFINITION OF SUPERVISOR MODE LIBRARIES
5. RE-BUILDING APL USING I/D SPACE AND SUPERVISOR-MODE LIBRARIES
6. COMPARISONS BETWEEN THE ORIGINAL VERSION OF APL AND THE I/D SPACE VERSION
7. PROBLEMS ENCOUNTERED IN THE CONVERSION PROCESS
8. APL CHARACTER SET FOR VT200 SERIES TERMINALS
9. PROGRAM AVAILABILITY
10. QUESTIONS AND ANSWERS

THE PROBLEM:

WE HAVE DEVELOPED A SET OF TECHNICAL PROGRAMS THAT COULD BE EFFICIENTLY PROGRAMMED IN APL. HOWEVER, THE SEVERE WORKSPACE LIMITATIONS OF APL-11 RESTRICTED US FROM IMPLEMENTING THESE PROGRAMS IN APL.

THE SOLUTION:

AFTER SUCCESSFULLY UPGRADING OUR OPERATING SYSTEM FROM RSX-11M TO RSX-11M-PLUS AND OUR CPU FROM A PDP-11/23 TO A PDP-11/73, THE POSSIBILITY EXISTED TO RE-BUILD APL-11 TO USE BOTH INSTRUCTION/ DATA (I/D) SPACE AND SUPERVISOR-MODE LIBRARIES.

GENERAL HARDWARE AND SOFTWARE REQUIREMENTS:

THE FOLLOWING HARDWARE SUPPORTS BOTH I/D SPACE AND SUPERVISOR-MODE LIBRARIES:

1. PDP-11/70
2. PDP-11/44
3. PDP-11/73
4. PDP-11/84

THE FOLLOWING SOFTWARE SUPPORTS BOTH I/D SPACE AND SUPERVISOR-MODE LIBRARIES:

1. RSX-11M-PLUS V2.1
2. RSTS/E V9.0 ?

THE SPECIAL CHARACTER SET

DEFINITION OF AN I/D SPACE TASK:

ON ALL PDP-11 CPUS WITH MEMORY MANAGEMENT, THERE ARE EIGHT ADDRESS PAGE REGISTERS (APR'S) CAPABLE OF ADDRESSING UP TO 64KB OF VIRTUAL ADDRESS SPACE.

ON THE CPUS MENTIONED ABOVE (PDP-11/44, PDP-11/70S, PDP-11/84), THERE EXISTS ANOTHER SET OF EIGHT DATA PAGE REGISTERS (DPRS) THAT CAN MAP UP TO 64KB OF VIRTUAL DATA SPACE. THE BENEFIT OF HAVING DPRS IS THAT PROGRAMS COULD BE POTENTIALLY TWICE AS LARGE AS A NORMAL PROGRAM RUNNING ON OTHER PDP-11S. HOWEVER, THE I-SPACE APRS CAN ONLY MAP INSTRUCTIONS WHILE THE D-SPACE DPRS CAN ONLY MAP DATA.

THESE TWO CONFIGURATIONS ARE DIAGRAMMED BELOW:

ALL PDP-11S		PDP-11/44,70S,84	
ALL SPACE		I-SPACE	D-SPACE
APR0	-----	APR0	-----
APR1	-----	APR1	-----
APR2	-----	APR2	-----
APR3	-----	APR3	-----
APR4	-----	APR4	-----
APR5	-----	APR5	-----
APR6	-----	APR6	-----
APR7	-----	APR7	-----

DEFINITION OF SUPERVISOR-MODE LIBRARIES

ON ALL PDP-11S WITH MEMORY MANAGEMENT, BOTH KERNAL AND USER MODE ARE SUPPORTED. USER MODE CONSISTS OF A FULL SET OF CPU REGISTERS AND ADDRESS PAGE REGISTERS (APRS) DEDICATED TO RUNNING USER PROGRAMS. KERNAL MODE CONSISTS OF ANOTHER FULL SET OF CPU REGISTERS AND APRS DEDICATED TO RUNNING THE OPERATING SYSTEM AND PRIVILEGED PROGRAMS. THE USE OF SEPARATE SETS OF CPU REGISTERS AND APRS REDUCES THE TIME IT TAKES TO PERFORM A CONTEXT SWITCH (SWITCHING BETWEEN USER PROGRAMS AND THE OPERATING SYSTEM). THIS CONFIGURATION IS DIAGRAMMED BELOW:

ALL PDP-11S

USER MODE	KERNAL MODE		
R0	-----	R0	-----
R1	-----	R1	-----
R2	-----	R2	-----
R3	-----	R3	-----
R4	-----	R4	-----
R5	-----	R5	-----
R6	-----	R6	-----
R7	-----	R7	-----
APR0	-----	APR0	-----
APR1	-----	APR1	-----
APR2	-----	APR2	-----
APR3	-----	APR3	-----
APR4	-----	APR4	-----
APR5	-----	APR5	-----
APR6	-----	APR6	-----
APR7	-----	APR7	-----

THE SPECIAL CHARACTER SET

APL-25

ON THE PDP-11/44, 70S, AND 84, THERE EXISTS AN ADDITIONAL MODE CALLED SUPERVISOR-MODE. RSX-11M-PLUS SUPPORTS MEMORY RESIDENT LIBRARIES THAT BE CALLED IN SUPERVISOR MODE. THE BENEFIT IS THAT PROGRAMS CAN POTENTIALLY BE 64KB LARGER THAN CONVENTIONAL PDP-11 PROGRAMS. SUPERVISOR-MODE LIBRARIES ARE CALLED USING A SPECIAL INSTRUCTION SUPPORTED ON THESE PDP-11S (CALL SUPERVISOR MODE - CSM). THE SUPERVISOR MODE SUPPORT CONFIGURATION IS DIAGRAMMED BELOW:

USER MODE	SUPERVISOR MODE	KERNAL MODE	
R0	-----	R0	-----
R1	-----	R1	-----
R2	-----	R2	-----
R3	-----	R3	-----
R4	-----	R4	-----
R5	-----	R5	-----
R6	-----	R6	-----
R7	-----	R7	-----
APR0	-----	APR0	-----
APR1	-----	APR1	-----
APR2	-----	APR2	-----
APR3	-----	APR3	-----
APR4	-----	APR4	-----
APR5	-----	APR5	-----
APR6	-----	APR6	-----
APR7	-----	APR7	-----
DPR0	-----	DPR0	-----
DPR1	-----	DPR1	-----
DPR2	-----	DPR2	-----
DPR3	-----	DPR3	-----
DPR4	-----	DPR4	-----
DPR5	-----	DPR5	-----
DPR6	-----	DPR6	-----
DPR7	-----	DPR7	-----

RE-BUILDING APL-11 TO USE I/D SPACE AND SUPERVISOR-MODE LIBRARIES:

FORTUNATELY, THE APL WORKSPACE IS A DATA STRUCTURE AND THUS OCCUPIES THE D-SPACE PORTION OF AN I/D SPACE TASK. THUS THE WOKSPACE CAN EXPAND TO OCCUPY JUST ABOUT THE ENTIRE DATA SPACE AVAILABLE.

THE USE OF THE FILE CONTROL SERVICES (FCS) SUPERVISOR-MODE LIBRARY MADE IT POSSIBLE TO RE-BUILD APL-11 WITHOUT OVERLAYING THE INSTRUCTION SPACE (I-SPACE) PORTION OF THE TASK. THE ONLY OVERLAYING THAT OCCURS IS IN D-SPACE WHERE ALL THE ERROR MESSAGE TEXT WERE LEFT OVERLAYED IN ORDER TO MAXIMIZE THE SIZE OF THE WORKSPACE.

THE SPECIAL CHARACTER SET

APL-26

APL-11: THE FOLLOWING TASK BUILD COMMAND FILE WAS USED TO RE-BUILD

```

APL/FP/CP/ID,APL/-WI/SH/-SP=APL/MP ; /ID IS I/D SPACE TKB
; ; SWITCH
;
SUPLIB=FCSFSL ; LINK TO FCS SUPERVISOR-MODE LIBRARY
TASK=.,.APL ; OTHER OPTIONS ARE AS
STACK=256 ; ORIGINALLY DEFINED
UNITS=15
ASG=SY0:5:6:13:14,TI0:15
ACTFIL=0
/
APL.ODL WAS MODIFIED TO ELIMINATE ALL I-SPACE OVERLAYS
AND REFERENCES TO SYSLIB WERE COMMENTED OUT SINCE THE
FCS SUPERVISOR-MODE LIBRARY WAS BEING USED FOR THESE
ROUTINES.
INSTALLATION INSTRUCTIONS TO MAXIMIZE THE WORKSPACE
SIZE ARE AS FOLLOWS:

```

MCR>INS APL/INC=64240

COMPARISONS BETWEEN THE ORIGINAL VERSION AND THE I/D SPACE VERSION:

CRITERIA	ORIGINAL APL-11	I/D SPACE APL-11	PERCENT CHANGE
WORKSPACE SIZE (BYTES)	32,688	53,548	63.8 % INCREASE
DISK SPACE USED (BLOCKS)	158	119	24.7 % DECREASE
SPEED		SLIGHTLY FASTER	DUE TO NO OVERLAYS
MEMORY REQUIREMENTS		ABOUT 2 TIMES MORE MEMORY CONSUMED	DUE TO I/D SPACE USE

MAIN PROBLEM ENCOUNTERED DURING THE CONVERSION PROCESS:

APL ESTABLISHES A LINKED LIST OF CORE ALLOCATION BLOCKS TO KEEP TRACK OF MEMORY UTILIZATION WITHIN A WORKSPACE. ONE WORD WITHIN THE BLOCK DEFINES THE SIZE OF THE BLOCK IN WORDS AND WHETHER THE BLOCK IS CURRENTLY IN USE. A DIAGRAM OF AN UNUSED CORE ALLOCATION BLOCK LOOKS LIKE THIS:

```

*****
* LENGTH *      LENGTH (IN WORDS) OF THE BLOCK
*****
* FLINK *      POINTER TO NEXT BLOCK IN LIST
*****
* BLINK *      POINTER TO PREVIOUS BLOCK IN LIST
*****
* *
* UNUSED *      AVAILABLE FOR APL USE
* *
*****
* LENGTH *      LENGTH OF BLOCK (SAME AS LENGTH ABOVE)
*****

```

THE SPECIAL CHARACTER SET

WHEN A BLOCK OF CORE IS IN USE, THE LENGTH IS SET NEGATIVE (I.E. MOST SIGNIFICANT BIT IS SET). ALSO, THE LENGTH VALUE IS CONVERTED TO BYTES (BY SHIFTING LEFT) AND THEN CONVERTED BACK TO WORDS (BY SHIFTING RIGHT) MANY TIMES DURING THE EXECUTION OF THE APL-11 INTERPRETER.

WHEN THE I/D SPACE VERSION OF APL-11 WAS CREATED, THERE EXISTED THE POSSIBILITY THAT BIT 14 (NOTE: NUMBERING FROM 0-15) COULD BE SET DUE TO THE INCREASE IN SIZE OF WORKSPACE. THIS SITUATION COULD NOT HAPPEN IN THE ORIGINAL APL-11 INTERPRETER.

AS A CONSEQUENCE OF CONVERTING TO BYTES, BIT 15 COULD SOMETIMES BECOME SET PARTICULARLY WHEN THE WORKSPACE WAS VIRTUALLY EMPTY. THE CONVERSION BACK TO WORDS CREATED THE FOLLOWING PROBLEM:

THE DEC PROGRAMMERS WHO IMPLEMENTED APL OPTED TO USE THE ASR INSTRUCTION TO CONVERT FROM BYTES TO WORDS. UNFORTUNATELY, THE ASR INSTRUCTION LEAVES BIT 15 UNCHANGED AND THUS MISTAKENLY COULD INDICATE THAT THE CORE BLOCK IS IN USE WHEN IN REALITY IT IS FREE FOR USE. TO CORRECT THIS PROBLEM, ALL USES OF ASR INSTRUCTIONS WERE CHANGED TO ROR INSTRUCTIONS WHICH ROTATE THE CARRY BIT INTO BIT 15. IT SHOULD BE NOTED THAT THIS WAS A DIFFICULT PROBLEM TO DETECT BUT, ON THE OTHER HAND, WAS FAIRLY EASY TO CORRECT.

APL CHARACTER SET FOR THE VT220 FAMILY OF TERMINALS

THE VT200 FAMILY OF TERMINALS SUPPORTS DOWN-LINE LOADABLE CHARACTER SETS. THE MAIN OBSTACLE OF USING THIS FEATURE IS THAT THE CODE SEQUENCE USED TO GENERATE CUSTOM CHARACTER SETS IS DIFFICULT TO GENERATE FROM SCRATCH. TO OVERCOME THIS DIFFICULTY, I HAVE WRITTEN A PROGRAM CALLED COMPOSE THAT CAN BE USED TO EASILY DEVELOP CUSTOM CHARACTER SETS. THE PROGRAM DISPLAYS A CHARACTER CELL THAT YOU USE TO FORM YOUR CUSTOM CHARACTER BY TURNING OFF AND ON PIXELS WITHIN THE DISPLAYED CHARACTER CELL. AFTER THE CUSTOM CHARACTER SET (UP TO 95 INDIVIDUAL CHARACTERS) HAS BEEN DEFINED, THE PROGRAM STORES THE CHARACTER SET IN A FORTRAN DIRECT ACCESS FILE. THE PROGRAM ALSO WILL GENERATE THE CODE SEQUENCE NEEDED TO GENERATE THE CUSTOM CHARACTER SET ON THE VT220 FAMILY OF TERMINALS.

I USED THE PROGRAM TO GENERATE THE APL CHARACTER SET. IT SHOULD BE POSSIBLE TO USE THE CODE SEQUENCE LISTED BELOW TO GENERATE THE APL CHARACTER SET ON NON-RSX OPERATING SYSTEMS SIMPLY BY TYPING THE FILE AT YOUR TERMINAL PROVIDED THAT THE VT220 IS CONFIGURED TO BE A VT220 AND PERMIT USER DEFINED KEYS.

THERE IS ONE RESTRICTION THAT APPLIES. THE RESTRICTION IS THAT THE OVERSTRUCK APL CHARACTERS WILL BE DISPLAYED AS ONE OF THE TWO CHARACTERS INVOLVED IN THE OVERSTRIKE. FORTUNATELY, YOU CAN ENTER OVERSTRUCK CHARACTERS IN THE NORMAL FASHION ON THE VT220 AND THEY CAN BE PRINTED ON APL TERMINALS SUCH AS THE LA12 OR THE LA120 WITH THE APL CHARACTER SET OPTION.

THE SPECIAL CHARACTER SET

**ARTIFICIAL
INTELLIGENCE**



DECUS

ARTIFICIAL INTELLIGENCE SIC

Chairperson

Cheryl Jalbert
JCC
Granville, OH

Ass't Chairman

Don Rosenthal
Space Telescope Science Inst.
Baltimore, MD

Symposium Coordinator

David Slater
Institute for Defense Analysis
Alexandria, VA

Ass't Symposium Coordinator

Session Note Editor

Greg Parkinson
Cognitive Systems Inc.
New Haven, CT

Newsletter Editor

Terry Shannon
DEC Professional Magazine
Springhouse, PA

Newsletter Publisher

Bob Zeek
Pfizer Inc.
Groton, CT

Membership Coordinator

Pamela Vavra
KMS Fusion Inc.
Ann Arbor, MI

PSS Scheduling

Tom Viana

Store & Buttons

Sally Townsend
Inst. Defense Analysis
Alexandria, VA

Quality Control Chair

Dick Ciero
Harris Corp.
Palm Bay, FL

Quality Control

Carol Guyermelli

Site Coordinator, Anaheim

Chris Goodard

Volunteer Coordinator, Anaheim

Peter MacDonough
Tractor Inc.
California, MD

Members-at-Large

George Humfeld

Matt Matthews IV

Evaluation Research Corp.

King George, VA

DEC Counterpart

Art Beane
Digital Equipment Corporation
Hudson, MA

(THE (LINKED LIST))

The newsletter of the
DECUS Artificial Intelligence Special Interest Committee
... it's the real thing.

FROM THE EDITOR

Welcome to the third issue of (THE (LINKED LIST)) and the first issue of the DECUS SIGS COMBINED NEWSLETTERS. Now that the transition to a single monthly DECUS publication is complete, you can look forward to receiving your copy of (THE (LINKED LIST)) every month.

Some interesting developments have taken place within the DECUS AI group since the last issue of this newsletter and the Spring DECUS Symposium. Our group now has a well-defined structure, an expanded steering committee and two Digital counterparts. By the time you read this, our organization will very likely have a new name and a broader scope. Until now, AI interests and resources within DECUS have been dispersed among several SIGs and their respective working groups. After a joint working group meeting in New Orleans, we decided that this diffuse structure limited our ability to meet the needs of those DECUS members who have an interest in AI. We concluded that we could better serve these members, and the Society as a whole, if we reorganized as a discrete functional group which would serve as a focal point for AI activities within DECUS. Accordingly, we have applied for formal recognition as the DECUS Artificial Intelligence Special Interest Committee.

As of press time, our SIC status is still pending. Personally, I'm optimistic, but I'd rather not make any assumptions or predictions. For more information on the SIC issue, refer to "From The Chair" and the "AI SIC Proposal." These articles provide an overview of our efforts to achieve SIC status, explain the factors behind our decision to apply for recognition as a functional group, and summarize the benefits that this proposed reorganization will yield.

This issue features an article by Art Beane, one of our DEC counterparts from the AI Technology Center in Hudson. "AI At Digital" provides an introduction to the structure and purpose of the AITC, and is the first in a series of articles authored by DEC AI specialists. Also included this month is a listing of our steering committee and background sketches on several of its members; a brief list of suggested readings in AI; a book review, and an AI SIC Survey and Questionnaire. Please take the time to complete the questionnaire - your responses will help us determine what you need and would like from an AI SIC.

We welcome your letters, questions, suggestions and articles. Material submitted for publication in (THE (LINKED LIST)) should be addressed to:

Terry Shannon
P.O. Box 53
Spring House, PA 19477

AI-1

FROM THE CHAIR

Celebrate with us! We've taken the first steps toward becoming a SIG. At the New Orleans symposium, we took the big step of requesting SIC status. The SIG Council and the Management Council agreed, pending acceptance of our missions and goals statement and budget. Our editor has condensed the missions and goals statement so that we may present significant portions of it in this edition. The full statement and our budget have received wonderfully positive comment. By the time you read this newsletter, we hope to be the DECUS AI SIC.

Our progress is exciting. We've begun to forge a group of people into a functioning steering committee. The deadlines have been difficult for a new group to meet, but I confess to pride in our accomplishments. The newsletter, you have in your hands. We've been fortunate enough to have an experienced editor. When the abstracts for Anaheim appear, I urge you to look for our sessions. A number of people have contributed to that effort and prospects look excellent. Our pre-symposium seminar (PSS) abstracts will be out soon also. Consider attending one of them. AI, a general coverage of a number of topics in the field, is an adaptation of a seminar previously offered by LATSIG. I will be coordinating that offering with assistance from a number of others. Greg Parkinson, its original author, will be working with me. Don Rosenthal is working with Nancy Wogrin and others at DEC to create a new seminar on OPSS, DEC's language for expert systems. Dave Slater is offering the seminar he has given for the VAX SIG on LISP, the traditional language of AI. We're also working on session notes, a tape, and having great fun debating the issues concerning the store and buttons and mascots.

We offer heartfelt thanks to a number of supporters: Larry Jasmann, our mentor, Ted Bear, Bill Brindley, Jeff Killeen, our review committee, Steve Pacheko, Marg Knox, and Kathy Hornbach who, as Chairs of the three SIGs (DMS, VAX, and L&T) that have been sponsoring us, offered their support, the SIG Council, Symposium Committee, PSS committee, and DECUS staff for helping us find our way around as newcomers, Art Beane, our counterpart, Margaret Meehan and others at Digital, and the large number of others who have offered insights, congratulations, and encouragement.

I think that we're on the road to having an excellent SIG. I invite you to join us.

Cheryl Jalbert, AI SIC Chair
J C C
128 West Broadway
Granville, Ohio 43023-0381
(614)587-0157

WHO'S WHO ON THE AI STEERING COMMITTEE

The portion of the AI Steering Committee which has been in contact with the chair since the last symposium is listed on the following page. Refinements in this list are to be anticipated by the time of our first woods meeting. We've also included biographical information provided by several Steering Committee members to give you an opportunity to get to know who we are and what we do.

AI-2

DECUS AI STEERING COMMITTEE

Name	Position
Cheryl Jalbert	Chair
Don Rosenthal	Asst. Chair
David Slater	Symposia Rep.
Greg Parkinson	Asst. Symp. Rep., Session Notes Editor
Terry Shannon	Newsletter Editor
Bob Zeek	Newsletter Publisher
Pamela Vavra	Membership Coordinator
Tom Viana	PSS Scheduling
Sally Townsend	Store & Buttons
Dick Ciero	Quality Control Chr.
Carol Guyermelli	Quality Control
Chris Goodard	Site Coordinator, Anaheim
Peter MacDonough	Volunteer Coordinator, Anaheim
George Humfeld	Member At-Large
Matt Matthews IV	Member At-Large
Art Beane	DEC Counterpart

IN GREATER DETAIL...

Cheryl Jalbert, AI SIC Chair

Cheryl Jalbert is a Senior Consultant with J C C, a consulting firm specializing in training, VAX/VMS applications design, and corporate computer planning. Cheryl has a masters in Computer and Information Science from the Ohio State University. Her particular interests are expert systems, database support for knowledge engineering, and application design. The AI SIC is important to Cheryl as a way to expand her knowledge and experience in the exciting field of AI and as an opportunity to share ideas with interesting people. In addition to serving as AI SIC Chair, Cheryl is a founder and co-chair of the LAMA LUG.

Don Rosenthal, AI SIC Vice Chair

Don Rosenthal has been active in DECUS for a year and a half, and has held the position of AI coordinator for the Language and Tools SIG. Don is still active in the L&T SIG, but his position as AI SIC Vice Chair gives him the opportunity to share his knowledge of AI languages. Don works for the Space Telescope Science Institute, where he is responsible for incorporating AI techniques into ground support systems. His current project is the implementation of an OPS5-based expert system that will transform astronomers' proposals into a series of spacecraft activities. With the help of Nancy Wogrin of DEC's Educational Services, Don will present a pre-symposium seminar on OPS5 programming at the Fall 1985 Symposium. As Don is especially interested in the enhancement of the OPS5 language and development environment, DECUS members who use OPS5 are encouraged to contact him with their wishlist ideas, suggestions or pet peeves.

Terry Shannon, Newsletter Editor

Terry Shannon has been involved with computers and data processing for over 10 years and has used Digital hardware extensively. He is presently employed by Professional Press as technical editor of two DEC-specific magazines. In addition to writing articles on all aspects of Digital computing (including several on AI), Terry serves as the firm's VAX system manager and recently published his first book, a user's guide to VAX/VMS systems. A DECUS member since 1982, Terry serves on the Site Management Steering Committee, the Communications Executive Committee and the Support Services Committee.

Pam Vavra, Membership Coordinator

Pam Vavra has a B.S. in mathematics and psychology and has been employed in a programming, analysis or research capacity for the past 12 years in a variety of scientific research environments, most of which have involved digital image processing. A DECUS member since 1979, Pam's interest in AI began in 1973 when she attended a lecture by Marvin Minsky and found AI to be an ideal area to apply her expertise in math and psychology. Currently employed by KMS Fusion, Inc., Pam recently completed a NASA-sponsored study aimed at applying AI techniques to image processing problems. She is primarily concerned with the modeling of mental processes employed by scientists as they interpret film data; and welcomes the opportunity to exchange information with others who are involved in similar application areas.

Robert Zeek, Newsletter Publisher

Robert Zeek has worked in Data Base Management on mainframes, minicomputers, PDP's and Vaxes for over 9 years. He started with a Princeton based DBMS vendor and has worked at two major R&D centers doing data base applications and administration. He has also developed several decision support systems based upon various Data Management Systems. His current assignment at Pfizer, Inc. is to specify hardware and software for Office Automation and electronic archiving of scientific data. This has taken him deep into the field of Natural Language Indexing and, consequently, AI as a whole.

AI SIC PROPOSAL

Artificial intelligence is generating considerable interest among computing professionals in both the research and applications spheres. Topics such as expert systems, natural language interfaces, machine vision, robotics, and decision support systems intrigue many of us and are increasingly important in applications environments. DEC is a leading developer of AI software and systems, and the firm's 32-bit and 36-bit computers are the hardware of choice in AI research. Because of DEC's extensive involvement with AI, DECUS members and potential members have begun to expect to find AI resources within the DECUS organization.

Those with an interest in AI have been able to turn to DECUS for several years now. AI-oriented sessions were first presented by the VAX SIG at the Spring 1983 Symposium. The Spring 1984 Symposium, which featured five AI sessions, marked the first time that DEC sent a representative from the AI Technology Group to a major DECUS event. The interest in AI was even stronger at the Fall '84 Symposium. A member of DEC's AI Technology Group offered to serve as a counterpart to an AI group within DECUS. At that time, the establishment of a formally recognized AI group was unfeasible because those DECUS members who shared a common interest in AI were scattered among several SIGs. Even so these members were enthusiastic about maintaining contact with one another between symposia.

In New Orleans, there were eleven AI-related symposium sessions. As the symposium progressed, it became evident the concept of a discrete AI group enjoyed widespread support. Many DECUS leaders, including the SIG chairs whose SIGs would be most affected by such a group, offered encouragement that those people interested in an AI group request SIC status. Accordingly, the AI working groups of two SIGs met in a joint session and voted overwhelmingly to request SIC status. Steering committee officers were appointed and short term goals were discussed, and a subsequent open meeting generated suggestions and proposals for future symposium sessions.

As of this writing, DECUS recognition of an AI SIC is not a foregone conclusion. However, we have drafted a position paper and an initial budget, and we have begun to organize our resources to achieve SIC status. It has become evident that there are resources which are not being exploited for lack of an AI SIG. It's equally clear that within the general AI community there is a need for the type of group which we propose. Numerous groups discuss AI research ideas, but there are few resources for persons interested in real-world AI applications.

Therefore, we propose to form the DECUS Artificial Intelligence SIC as a precursor to an Artificial Intelligence SIG. Our goals include:

- o Providing an effective forum for the many changing issues that affect users developers, and managers of systems involving AI technology relevant to DEC products.
- o Facilitating the exchange of information and ideas in this sphere among DEC DECUS, and the external AI community.
- o Providing a focal point for those DECUS members who want to broaden their knowledge through exposure to this new and much discussed subfield of computer science.
- o Serving as a resource for DECUS members who face the challenge of implementing applications which employ aspects of AI technology.

A number of factors were analyzed and evaluated during the course of preparing this proposal. The establishment of any new group within DECUS is contingent upon several criteria. Significant among these are the availability of resources, the endorsement and support of DECUS leadership, the level of interest shown by the general DECUS membership for the topic or area the group intends to address, and the ability to deliver a viable product or service that will not duplicate the offerings or activities of existing groups within the Society.

We are confident that our proposed AI SIC fulfills these criteria. We have a growing membership list, an active steering committee and a cooperative DEC counterpart. We have sponsored useful, well-attended symposium sessions and presymposium seminars, and we plan to expand their number and scope in the future. Our leadership consists of qualified, eager people who have the talent, motivation and experience necessary to the success of a SIC. Our goals are supported by other DECUS leaders, and the DECUS membership as well.

A substantial and growing level of interest in AI exists among DECUS members. Although we have attempted to satisfy this interest through a newsletter, symposium sessions and presymposium seminars, a more formal approach is needed. As an AI SIC, we will foster communication among people creating AI applications, satisfy the intellectual curiosity of the general DECUS membership, and provide resources and information for the novice.

The provision of these services will not duplicate the activities of existing DECUS organizational units. A service group dedicated to the dissemination of information on AI applications would be unique both within DECUS and the computing community as a whole. There is a growing need for an organization devoted to the application of AI technology to real-world problems. Our group will fulfill this need by discussing AI products, user implementations, and the services offered by DEC's AI Technology Group. We believe that an AI SIC will have a positive impact on DECUS, and invite you to become a participant in our activities. Pending SIC approval, we will:

1. Expand our scope and begin to fulfill our mission by seeking further support from our counterparts, other people known to us within DEC's various AI groups, and those users we can identify who are active in AI projects and likely DECUS participants, but are not currently active.
2. Contribute high quality material to the newsletter.
3. Hold working meetings to further refine our definition and draft our charter.
4. Present high quality Pre-Symposium Seminars and Symposium Sessions.
5. Present a draft of the charter at the Spring 1986 Symposium in Dallas.
6. Continue our work on the newsletter, symposium sessions, PSSes, resource reviews, communication with DEC, and refinement of our charter.
7. Seek SIG status.

AI AT DIGITAL

By Art Beane
Manager, Product Management & Marketing
DEC AI Technology Group

Over the past few years, there has been an incredible amount of media hype around artificial intelligence. Some articles tout AI as the saviour of the industrial world, others complain that AI never keeps a promise. Obviously, the truth must be somewhere between. At Digital, artificial intelligence is considered to be a technology to be applied to solving certain kinds of complex problems. Over the next several months, we'll be writing articles that will demonstrate how to tell whether a program is amenable to AI techniques or not, how the techniques are used at Digital, and give suggestions to help you get started in AI.

These articles will be written by members of Digital's Artificial Intelligence Technology Center ("AITC"), located in Hudson, MA. This first article will give you some idea about the structure of the AITC. There are currently about 150 people in the center, working in four groups. As an indication of how much importance Digital places in AI technology, each group reports to a different vice president. All told, at least 10 of the Digital vice presidents have some AI responsibilities.

Intelligent Systems Technologies Group (IST)

IST is part of Digital's manufacturing organization, and is the founding group of the AITC. The group is chartered to develop internal applications using AI technology. The most publicized IST applications, XCON and XSEL (batch and interactive configurers, respectively), are just the tip of the iceberg. More than three dozen projects are underway in such areas as complex business management, inventory control, plant and shop floor scheduling and project management. IST is also responsible for in house training of application developers, sometimes called "knowledge engineers."

AI Applications (AIA)

As part of Digital's Field Service, AIA spends most of its time on diagnostic class applications. Notable among these is AI SPEAR, which detects intermittent failures in magnetic tape drives by evaluating the error log, rather than by running diagnostic programs. Here's a program that really believes in Murphy's Law: Diagnostics never detect failures when the problem is intermittent! Other application areas being worked on in AIA include automated crash dump analysis, trouble shooting tools for complex networks, and an expert system that assists TOPS-20 users to migrate to the DCL command language on the VAX.

AI Marketing (AIM)

Marketing is one of the most difficult tasks at Digital. AIM has the responsibility of not only convincing customers that Digital's experience in AI is proof that the technology is real, but also must help the strategic marketing groups (A.K.A. "Product Lines") fit AI into their plans. To meet these goals, AIM concentrates to a large extent on the applications to which AI is most appropriate. Using both third party and Digital developed tools, AIM provides the support and guidance that will help organizations use this technology most effectively.

AI Technology Group (AITG)

As part of Central Engineering, AITG has the charter to produce Digital developed AI tools. AITG is split into four subgroups, each dedicated to a piece of this goal. The LISP related product group is responsible for VAX LISP development, much as its name implies. In addition, members of this group serve on the Common LISP standards committee. A second product development group is responsible for all other products. Best known of these is OPS5, the high performance expert system development language. Advanced Development is the name of the third subgroup. A/D has a two-part job. The first is to perform the research needed for advanced features of existing and future products. The second is to serve as the incoming technology transfer point for AI. In this role, they monitor research projects in universities and national programs (such as MCC in the US) to determine which of these can be turned into products. The fourth subgroup in AITG is Product Management and Base Product Marketing. This group has the responsibility of managing product life cycles. In effect, the product manager has to look like an engineer to marketing groups and customers, and like a marketer to engineers. Product managers are responsible for making sure the product that gets developed meets the needs of the marketplace in functionality, price and timing.

To summarize, the AITC is Digital's central resource for AI applications and products. Over the next several months, various members of the AITC will show how Digital uses this technology to solve real-world problems. This is why we say, "Artificial Intelligence: Experience Makes It Real."

SUGGESTED READING LIST

There are a significant number of books devoted to AI and its various subdisciplines. One of our ongoing projects is the compilation of a bibliographic listing of AI literature. Ultimately, we plan to expand the scope of this reference service by including reviews or summaries of each title on the list. Although this effort is still in its preliminary stages, we have included an abbreviated book list to give you some suggestions for further reading. The content of each book is indicated by the following key: I - Introductory, R - Reference, RR - Research Review, T - Technical.

Artificial Intelligence: Making Machines Think, Peat (I)

Artificial Intelligence: The Search For The Perfect Machine, Stevens (I)

Artificial Intelligence and Robotics, Gevarter (I,T)

Building Expert Systems, Hayes-Roth et al (I,T)

Common LISP, Steele (T)

Engineering Intelligent Systems, Glorioso and Osorio (T)

Expert Systems, King (I)

Godel, Escher, Bach: The Eternal Golden Braid, Hofstadter (I)

Into The Heart Of The Mind, Rose (I)

LISP, Winston (I,T)

Machines Who Think, McCorduck (I)

Programming In PROLOG, Clocksin and Mellish (I,T)

Programming Expert Systems In OPS5, Brownston et al (I,T)

Readings In Artificial Intelligence, Webber and Nilsson (RR,T)

Robotics, Minsky (I)

Rule-Based Expert Systems, Buchanan and Shortliffe (RR,T)

The AI Business, Winston and Prendergrast, (I)

The Cognitive Computer, Schank (I)

The Fifth Generation, Feigenbaum and McCorduck (I)

The Handbook Of AI (3 volumes), Barr, Cohen and Feigenbaum (R)

BOOK REVIEW:

ARTIFICIAL INTELLIGENCE AND ROBOTICS: FIVE OVERVIEWS

Business/Technology Books
P.O. Box 19475
Sacramento, CA 95819
568 pages

Every time I visit the local bookstore, it seems that there's at least one new AI book displayed prominently on the science and technology bookshelf. While most of these books are interesting to read for diversion, they don't convey much technical information and are of little value to persons attempting to conduct an in-depth study of AI. Artificial Intelligence and Robotics: Five Overviews, by Dr. William Gevarter, is a welcome contrast to the current bookstore offerings. If you're looking for a comprehensive, technically oriented reference source on artificial intelligence and related topics, this book merits your consideration.

The book is structured as a five volume, seven section report. Collectively, the five volumes provide a through grounding in AI and its key application areas: robotics, expert systems, computer vision, and natural language processing. Each section details current developments and future trends in AI or one of its application areas, and each is supported by useful figures, tables, and references.

The book begins with a 90-page introduction to robotics. In this section, brief explanations of the key implementation issues in the field of industrial robotics are presented. After summarizing the current state of the art in this discipline, the author details research efforts underway in the United States and abroad. Interestingly enough, as the book points out, while Japan boasts some 70% of the world's robot population, it is the Europeans whose robotics research program is number one in terms of monetary investment.

Section Two deals with expert systems, the current standard bearers of AI. In 64 pages, most of the salient points brought forth in Hayes-Roth's 'Building Expert Systems' are summarized, leaving the reader armed with a good conceptual view of what expert systems are, how they work and what future research is needed to make these sophisticated programs more practical.

The following section is a 150-page review of computer vision. From the standpoint of AI, computer vision or image processing is still a discipline in its infancy. In general, it's one of the least understood and most complex human functions that we are attempting to model and mimic today. The text does a good job of explaining what is involved in image analysis and processing, and you'll come away with an appreciation of the difficulty involved in teaching a machine to distinguish a shadow from a rock, or an iron fence from a less substantial clump of reeds or grass.

Section Four addresses natural language processing. At 44 pages, this segment of the volume is somewhat skimpy, but it does provide an overview of the issues that form the crux of NLP: semantics, parsing and understanding. If you want to delve more deeply into this facet of AI, the section concludes with an extensive list of sources for further information.

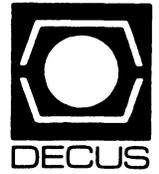
The last 220 pages of the book consist of Sections 5A through 5C, collectively referred to as Artificial Intelligence: An Overview. Section 5A introduces AI, explains its basic elements, history, application areas and projected future developments. As with several other sections of the book, this segment concludes with a useful glossary of AI terminology.

Section 5B surveys the fundamental application areas for AI. These are treated individually as expert systems, computer vision, natural language processing, speech recognition and understanding, speech synthesis and problem solving and planning. While some of the topic names may seem repetitious, the content of this section is by no means a rehash of the first four sections of the book. The author's emphasis of AI applications is most appropriate, for it is through these real-world applications that AI shows its practicality and profitability.

The book concludes with a section entitled "Basic Topics". Here, the author discusses AI and automation, knowledge representation schema, the problem solving aspects of AI and computational logic. This material is essential to more detailed study of AI techniques and implementation languages, particularly the logic-oriented PROLOG favored by the Japanese and a growing number of AI specialists in this country.

Artificial Intelligence and Robotics: Five Overviews will never make the bestseller list, but it's a valuable reference tool for the serious student or practitioner of AI. Dr. Gevarter has an extensive background in AI, and this is reflected throughout the book. I enjoyed reading the volume and was impressed by the wealth of information the author presents to his readers. While I question the logic of presenting introductory information in the final sections of the book, the fact that the material is understandable to a layman makes this title particularly suitable for the EDP specialist who wants to obtain a thorough grounding in AI fundamentals, techniques and applications.

- reviewed by Terry C. Shannon



BUSINESS APPLICATIONS

BUSINESS APPLICATIONS SIC STEERING COMMITTEE

Chairman

Stuart Lewis
Douglas Furniture
Bedford Park, IL

Library Representative

David Hittner
Projects Unlimited
Dayton, OH

Symposium Coordinator

Ronald Rubin
Sportmart
Niles, IL

CL SIG Liaison

Becky Burkes
Financial Insurance Consultants
Covington, LA

Asst. Symposium Coordinator

Bobbie Wiley
CEI Perry Nuclear Power Plant
Euclid, OH

DMS SIG Liaison

Joe Sciuto
Army Research Institute
Alexandria, VA

LRP and Marketing Coordinator

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

Members-at-Large

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

Marketing Asst.

George Dyer
Gallaudet College
Washington, DC

Robert Kayne
Gallaudet College
Washington, DC

Ray Evanson
Paragon Data Systems
Winona, MN

Communications Representative

Larry Eisenberg
L.H. Eisenberg Law Corp.
Encino, CA

Digital Counterparts

Sue Yarger
Digital Equipment Corporation
Merrimack, NH

Newsletter Editor

Thomas Byrne
L. Karp and Sons
Elk Grove Village, IL

Ray Arsenault
Digital Equipment Corporation
Merrimack, NH

Associate Newsletter Editor

Steve Simek
IRT
San Diego, CA

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

LETTER FROM THE EDITOR

Hello! Let me welcome you to the brand new Decus Newsletter. I'm sure you have already heard all you want to hear about the "new", "better" format. I'll just add that we should all benefit from the change. At a minimum, it will be cheaper for most of us!

But, that's not all that's new. You are reading a brand new section in that brand new newsletter. Welcome to the Business Application Special Interest Group Newsletter. I am exercising my "Editorial" rights in calling us a Group, we are still a Committee. Our strong leadership, headed by Stu Lewis, is working on that minor technicality.

Finally, we also have a new Editor. This is my first time ever doing anything remotely like trying to put together a newsletter. So please, be gentle with me.

What we are going to try to do in this newsletter is something a little different from what you expect to see in a Decus Newsletter.

Normally, you would read the PAGESWAPPER and see some nifty little (or big) DCL procedure or MACRO code telling you how to modify, automate, fix (pick one) your system to do strange and wonderful things. Or you read the WOMBAT EXAMINER to find out the latest clever work around for some un-documented feature in Datatreive. You get the idea.

I take my hat off to the technical geniuses who work out all the arcane secrets of our various systems. More than once, their brains and effort have made my life a little easier.

The problem is that not all of us are wizards. Our contributions to the newsletters in the past have been passive. We could only read about what others had done.

Those days are gone forever! EVERYONE gets to contribute to the Business Application SIG (Editorial license again) Newsletter! Our purpose is to share with others how we have used computers to come up with solutions to our business problems.

How has the use of computers made your company, job, or self more (less?) successful? What are you doing today that would be impossible/impractical without the computer? What can't you do today that you were told you could by the ever optimistic salesperson who sold you the latest "hot stuff"?

We don't intend to compete with the technical guys who fiddle the bits for us. We want to know what you were trying to get done that needed those bits fiddled in the first place. What are you using computers for? Why are you using computers instead of some other way of getting the job done?

What problems has the computer solved or created for you?

War stories and glorious adventures, we want to hear them all. This is your chance to contribute to that thing called Decus. Your ideas and experience can help others "get the job done".

We started out talking about new. We end up talking about old. The old lines... This is your newsletter... We need your articles to make it work... We need your participation... have never been truer than they are today.

Decus is an exciting organization. It is adapting to the new ways computers are made, sold, used. The new Business Application SIG is one of those adaptations. Welcome to that new Decus, that new SIG, that new newsletter, welcome from that new Editor. Interesting things are happening. You are making them happen. Let us know about them, we want to share the wonder.

BUSINESS APPLICATIONS SPECIAL INTEREST COMMITTEE

The Business Applications SIC serves the community of users, managers, and developers who are concerned with business application solutions to business problems.

The focus is on business computer systems with an integrated set of solutions rather than sets of independent hardware and software products.

Business application systems provide solutions in areas such as retailing, manufacturing, warehouse distribution, inventory control, finance, accounting, project management, business planning, and the like.

These business application systems are diverse in nature, but are integrated into a uniform environment by the designer and developer.

We address the needs of the designer and developer who want information concerning the proper makeup of a business application.

We address the needs of professionals who want to know the uses that technology-based solutions have in the business environment. These people are interested in the integration of diverse applications in order to effectively solve business problems.

We are interested in promoting communication between these designers, developers and professionals in order to promote the evolution of technology-based solutions that meet user needs.

We serve membership by addressing the issues involved in the use, management, and development of computer-based solutions.

WELCOME!

To the Business Applications SIC newsletter! We thank you for your interest in the SIC and in this newsletter! This premiere edition of the Combined Newsletter also marks the first edition of a newsletter for our newly formed committee. The SIC marked its first symposium in May 85 in New Orleans, where nearly thirty sessions pertaining to Business Applications were presented by the SIC. They were received by a responsive and enthusiastic audience. We are encouraged and pleased by the response to our new organization, and thank those of you who attended in New Orleans, those of you who stopped by our SIC suite to offer encouragement and support! A special thanks to the DECUS SIG Council and DECUS staff for their thoughtful support in the genesis of the Business Applications SIC. We look forward to seeing you again in Anaheim, and invite all to visit our sessions (we are now reviewing over seventy SIC sessions for Anaheim - a pleasing job!) and also invite you to visit our suite at the Disneyland Hotel this December. We will be detailing specifics on the sessions for Anaheim in future issues of this section of the Combined Newsletter. Be aware also that the Business Applications SIC is sponsoring two Pre-Symposium Seminars at the Fall Symposium. Details will follow in the Symposium Registration Kit that is forthcoming. Remember that Pre-Symposium Seminar registration is one of the best bargains in seminar programs available to you.

Stuart Lewis, Chairman
Ronald Rubin, Symposium
Representative

LEGAL PROBLEMS WITH SOFTWARE PROTECTION AND SOFTWARE AND HARDWARE WARRANTIES

Lawrence H. Eisenberg
L.H. Eisenberg Law Corp.

Introduction

Computer software - for applications and hardware - is the subject of considerable litigation in the United States, with damage awards now reaching millions of dollars. These awards, for the most part, are based upon a growing recognition of the need for honesty in representations by a hardware or software vendor to members of the public, following a rather long period of time where the vendor's warranty liability limitations were virtually a total shield.

Even though there is a definite and measurable trend toward holding vendors to a more strict accountability, decisions throughout the country are anything but uniform and what may appear to be settled in one state may be quite another story in another. Even in the same jurisdiction conflicting decisions are being reported so that it is exceedingly difficult to advise clients as to how to be protected in the event of litigation and how, if at all possible, to avoid litigation.

This discussion is GENERAL and any rules or advice given must be understood as not applying to any individual situation but to be illustrative of some of the problems and exposures existing in the big world of commerce.

YOU MUST CONSULT WITH AN ATTORNEY FOR SPECIFIC PROBLEMS OF ADVICE REGARDING ANY PERSONAL QUESTION YOU HAVE.

At this time, the cardinal rule with respect to protecting oneself may be expressed as mere hope or prayer, at least until the rules developed by both the courts and the legislatures and Congress become far more solidified than they are at this time.

If a VENDOR is engaged in several jurisdictions (e.g., across state lines), then the issues which relate to what may be allowed and what may not in any given situation not only may, but probably will, vary.

While the subject of this discussion is primarily that of SOFTWARE, many situations are identical with hardware - it's just that the representations may be different and the courts aren't used to "hardware" types of problems - yet. Indeed, not only the courts, but the computer industry itself has considerable difficulty in defining the difference between hardware and software, resulting in such inbetweens as "firmware" and "slushware"

Software Protection

Four types of possible protection for software:

- Copyright
- Patent
- Trade Secrets (contract) - Unfair Competition
- Plain secrecy

Copyright

Current law is poorly defined. In *Data Cash Systems, Inc. v. JSA Group, Inc.* [(N.D. Ill. 1979) 480 F. Supp. 1063, aff'd on other grounds (7th Cir 1980) 628 F.2d 1038] the trial Court found that a computer chess game, whose object code was unloaded from and copied to ROM, was not protected by copyright as the writings were not in human readable form. It is not entirely clear, however, whether the Court felt that the COPYRIGHT SYMBOL did not appear in human readable form upon the ROM, itself, or whether anything in object code is fair game. Whatever the Court felt,

the decision is, in this author's opinion, without any social redeeming value whatsoever and should be viewed with considerable caution. (Subsequent decisions in other jurisdictions have not followed this case.)

Yet, in *Tandy Corp. v. Personal Computers, Inc.* [(N.D. Calif. 1981)524 F.Supp. 171] Tandy (Radio Shack) complained that Personal Computers, Inc. had unloaded its TRS-80 Basic Interpreter from ROM and created a new ROM which was being used by Personal Computers, Inc. The U.S. District Court in California ruled that a cause of action was stated, recognizing that its decision was 180 degrees from the JSA decision. Although there has not been a large number of decisions on the subject since Tandy, it appears that the clear direction is toward Tandy and away from JSA. Finally, in an ITC decision [United States International Trade Commission], which is a federal agency acting to prevent unfair trade practices and competition with respect to import trade [June 26, 1981] ITC Docket No. 337-TA-87, the dispute involved a game known as GALAXIAN (an arcade video game) which originated in Japan. In view of the unlimited number of results which could be obtained from the game, the ITC decided that it simply was too difficult to decide whether there was any copyright violation or unfair trade practices arising out of the direct copy of the ROM.

COPYRIGHT LAW AMENDED. The U.S. Copyright laws have been amended recently with the stated purpose of providing some coverage for the ROM-type problem as well as general software protection problems. However, it remains to be seen whether the Courts will apply a reasonable approach to the subject or will continue with irrational rules of exception.

The real problems are quite complicated and involve a deep understanding of just what a copyright is. There is not enough space in this entire Newsletter to fairly discuss the true issues, although one may be able to get some idea upon the subject when we point out that a copyright is an absolute right to prevent another from copying an ORIGINAL work (generally a writing, but may be an art form). The key word is "ORIGINAL". However, a DESIGN or PROCESS traditionally is not subject to copyright protection. Therefore, one can see that it will be a long time before the courts will be able to resolve just what computer program is so original that it may be deemed subject to copyright protection, does not primarily involve a design or process AND does not involve too much copying of others' "sub-program" material.

And, as we have discussed above, the rules are so different among the various states, as well as among the different circuits of the United States Courts of Appeals. (Each circuit may, and often does, arrive at conflicting rules of law which will, until resolved by the United States Supreme Court, represent the rule of law to be applied among the states included within that specific circuit.)

Patent Law

As a general rule PATENTS will not protect software designs. Just accept that! (As with copyrights, this is a very complex area of law. Moreover, Congress has elected to proceed with providing protection under the copyright laws for most software program situations).

There is a very narrow exception for software design that requires specific hardware concepts, so that the entire matter (process and software design) may be acceptable as a PATENT.

Most software programs - including operating systems - will not meet the narrow definition. A generalization is TRY TO AVOID RELIANCE UPON PATENT LAW PROTECTION - ESPECIALLY SINCE IT MEANS DISCLOSING THE ENTIRE PROGRAM AND PROCESS IN ORDER TO GET THE PROTECTION.

Another consideration is the expense in securing PATENT protection, as well as the time required before the patent may be granted. Patent searches can cost a considerable sum of money, and there is no guarantee that a patent will withstand the scrutiny of a court proceeding, which would be required in order to enforce the

patent. Defenses available to a defendant in a patent case are extraordinary, including a claim that the patent is invalid since the "idea" is "obvious" from the existing art. REMEMBER - there are no reported cases where a patent has been held to protect a pure software program and only a very very few where a patent has been declared valid in connection with a software program (each of which involved a process).

Trade Secrets - Unfair Competition

Trade Secrets and many forms of unfair competition involve state laws and the rules vary with each state. A general rule is that if the program is treated as a secret within the company and that the specific program cannot be ascertained from casual observations, it probably is entitled to trade secret protection.

At the moment, trade secret law may provide the best protection available in the U.S. BUT, IF COPYRIGHT LAW IS FOUND TO BE APPLICABLE, THEN IT MAY BE DEEMED TO BE THE EXCLUSIVE REMEDY IN MOST, IF NOT ALL, CASES. The reason for this is that if Congress decides that software programs are entitled to copyright protection, then state laws, which might provide better or easier to prove protection, might be deemed to have been pre-empted by nothing less than the Constitution of the United States, which vests in Congress the exclusive power to legislate regarding copyrights.

Written contracts with employees to protect against piracy are suggested. Also, the doctrine (unfair competition and trade secrets) can be applied between vendor and vendee by a written contract. Implied agreements to maintain the trade secrets are available in most cases, but they are more difficult to prove. Be certain that the vendee knows that you claim a trade secret and that the vendee agrees to protect the secret. Thus, by way of example, a limited license to a potential vendee for the use of a software program (let's say for demonstration or field test purposes) may, if carefully drafted, afford the vendor protection against future use or distribution by the vendee of the program in violation of the terms of the license under state laws dealing with trade secrets and unfair competition.

An employment contract, wherein an employee agrees that software developed during the employee's tenure (whether on the job or off the job) may be valid. However such contracts require separate and valuable consideration. If offered to an employee prior to employment, then the act of employment generally is deemed to be adequate consideration. However, if such agreements, without additional or adequate consideration, particularly under threat of termination of the employment, the courts generally will disregard the contract and the employer could lose a valuable protection out of a sense of greed. (The profits which might be realized as a result of the employee's efforts, or it could be a cash payment or an increase in wages, etc.)

Keep in mind, however, that many states prohibit a restraint against an employee from competing with the employer in the event of termination (regardless of cause). Most states at least follow the Restatement of Torts upon the subject and, in any event and regardless of the specific language of an otherwise valid employment contract, limit such restraints (if otherwise valid under state law) to a reasonable time and a reasonable geographic locale.

Again, it is possible that federal law may be binding, even in the trade secret world, especially if the software issues should fall within patent or trademark law which may be exclusive with the federal government.

Violation of one's trade secret can result in an unfair competition claim, which may include punitive damages, if a competitor picks up your trade secrets. Further, the party disclosing the secret also is a target defendant.

In appropriate cases, improper use of a trade secret could constitute a violation of the federal anti-trust laws which can allow for treble damages and recovery of attorneys' fees, even if a patent or copyright is not involved.

To illustrate the difficulties, consider that many courts even struggle with whether decompiling a program, in order to ascertain the source code, and then recompiling it as your own can constitute a trade secret violation. Problem here is that protection is in the form of copyright, and common law unfair competition might be preempted by federal copyright law.

Plain Secrecy

The best possible protection is actual "secrecy". Keep your source files to yourself. If you have to disclose them, then be sure to have a trade secret agreement with your customer, but remember that you then have given up your secret, at least a little, and every little bit adds up to an eventual total loss.

If your object code is unloaded and then reloaded, don't be afraid to pursue the matter. The bigger they are, the harder the ultimate fall. YOU DON'T HAVE TO SPEND A FORTUNE, BUT YOU DO HAVE TO STAY IN AND FIGHT and it will be expensive.

Do include your copyright information in and on every program. This means in the code and upon every type of cover which can hold plain language. (Current international law does not permit use of the "(C)", but requires the c in a circle. However, present indications are that international conventions will be amended to permit the "(C)" indication within computer programs to be displayed on a screen.)

International copyright convention requires the following: Copyright (C) 1982 YOUR NAME BUT - for international protection, the "(C)" MUST BE A c IN A CIRCLE - the parentheses will not meet international requirements (at least not yet). U.S. Copyright protection may not require the copyright information, but don't chance it. Also, copyright material must be filed with the Government prior to filing an infringement suit. This means that you may be protected, just by indicating that you claim a copyright, and you do not have to register your software with the government in order to be protected, but if you bring a suit on the copyright then you must register the copyright. The principal difference is that you may not be entitled to the general and punitive damages provided for by the copyright laws for infringements taking place while your product was copyrighted.

Software Liability

Who is responsible for damages arising from failure of software (and firmware) to work as expected?

The rules vary from state to state. Generally, federal courts will apply local state rules - but often this is tempered with federal rules or independent thinking of the federal judges, especially where the particular state has not expressed itself clearly upon the issue.

In the early days of computers, a long line of cases upheld strict limited liability on the part of the manufacturer. Generally, these limitations were to the purchase price, only, and for a very short period of time.

Subsequent cases - especially more recent cases - are finding alternative theories to cut through the claimed limited liability shield of manufacturers. Very substantial damage awards are now being reported.

One may expect some states, such as those relying upon computer manufacturers, to sustain a manufacturer's limited liability; on the other hand, liberal states, such as California, disregard most limits on liability.

The most common theory for bypassing the limited liability or warranty is that of DECEIT, MISREPRESENTATION OR FRAUD, especially where substantial "consequential damages" exist but purport to be limited under a limited warranty - (i.e., harm arising as a result of a failure, such as payroll checks for double the amount due and all records as to who was paid are destroyed; inability to perform a contract, resulting in damages to the vendee, etc.).

MISREPRESENTATION may be NEGLIGENCE or INTENTIONAL. The only difference is

whether punitive damages will be awarded.

DECEIT and MISREPRESENTATION can include anything from date of delivery to capability of the equipment to what may be expected from the software. More importantly, the misrepresentations can be derived from brochures of the manufacturer, to statements made by a salesperson who may not be related in any manner to the original manufacturer.

MOST COMMON AREA FOR DECEIT and MISREPRESENTATION is where the software or the equipment, or both, cannot or does not perform within reasonable or express expectations of the user. The test, now, will be what a reasonable person in the position of the user may reasonably expect from the product. As systems keep moving to unsophisticated users, the expectations may be expected to increase and the knowledge of the user will decrease, presenting the vendor and manufacturer with more and more difficulties and duties.

WARRANTY LIMITATIONS do not protect against misrepresentation in most jurisdictions. The theory is that one cannot contract away his duty of good faith and fair dealing and misrepresentation is a violation of that duty. For a recent application of "misrepresentation" matters, see APPLICATIONS INC. v. HEWLETT-PACKARD CO (NO. 77 CIV. 5937 -S.D.N.Y. Oct. 20, 1980) 501 F.Supp. 129. The New York federal district court, applying California law, agreed that Hewlett-Packard's EXPRESS LIMITED WARRANTY precluded recovery of damages for consequential damages on the express warranty. However, although both New York and California adopted the Uniform Commercial Code, the general rules of fraud and misrepresentation (negligent and willful) were available as additional theories of recovery. THE COURT OBSERVED THAT EVIDENCE OF THE CONTENTS OF BROCHURES AS TO THE CAPABILITIES OF THE HEWLETT-PACKARD COMPUTERS COULD BE CONSIDERED WITH RESPECT TO THE REPRESENTATIONS ISSUES. (Our APL people should be interested in the case, as it involves the alleged inability of Hewlett-Packard's APL-3000 language to operate on its 3000 series small multi-user computers.)

FAILURE TO DISCLOSE AS MISREPRESENTATION. Failure of a vendor to disclose a known - or suspected - defect generally constitutes misrepresentation and generally is deemed WILLFULL MISREPRESENTATION. EXAMPLE: Vendor is advised by user of a system or program defect. Vendor fails to notify other users or new buyers of the defect. The failure to notify may be a breach of duty - even to old customers and even to customers who are known to the vendor, but who do not have service contracts. In a way, this is similar to a failure to recall known defects in automobiles.

STRICT LIABILITY. As computers become more complex, and as users become more unsophisticated, we may expect that computer programs, as well as computers themselves, will fall more and more into the area of law known as "Strict Liability", which means that the manufacturer and/or the vendor will be strictly liable to a user who uses the product in a manner intended and suffers a loss as a proximate result of having so used the computer or program. Strict liability may prevent the manufacturer or vendor from asserting, as a defense to a claim, that the user was negligent or contributed to the harm.

Protection Against Damage Claims

Thorough testing of programs for purpose intended, with good records of testing procedures and results, is one protection - especially valuable as a defense against claims of willful misrepresentation.

Avoid "puffing". Be certain that the software will do the job you claim that it will, and that it will do the job on the equipment upon which it is expected to run. Also, be certain that the software will do the job that a user reasonably may expect it to or be prepared to clearly state on the package what the limitations are.

Obtain written specifications from your VENDEE whenever possible. If the VENDEE can't provide specifications, the VENDOR can provide specifications, just

be sure to have VENDEE sign the specifications as representing his needs and document your meetings and discussions with your vendee.

Require the VENDEE to TEST the software to the VENDEE's expectations before the product is accepted. If possible, provide the VENDEE with a testing procedure and records to be maintained for the test.

SET FORTH EXPRESS LIMITED WARRANTY. Let the VENDEE know exactly what it is that is being represented. INCLUDE A STATEMENT IN LARGE LETTERS TO THE EFFECT THAT NO OTHER REPRESENTATIONS HAVE BEEN MADE OR ARE RELIED UPON AND THAT THE WARRANTY IS LIMITED. THIS IS IMPORTANT LANGUAGE - SEE YOUR LAWYER AND GET THE BEST POSSIBLE WORDING. The language may not protect you against suit, but it can limit the claims to be made against you and in some jurisdictions may be all that is required (subject to actual fraud).

QUALIFY YOUR WARRANTY by requiring that the VENDEE have made all required tests before final acceptance and that such testing is a condition of any warranty you may provide. Get your VENDEE's test documentation and retain it as part of your records.

REQUIRE - IN THE WARRANTY - THAT THE VENDEE MAINTAIN BACKUP COPIES OF ALL SOFTWARE AND DATA. While failure to have done so may not absolve the vendor of damage, it certainly will tend to limit the damages for failure of the vendee to have mitigated damages. FURTHER, IF THE REQUIREMENT TO MAINTAIN BACKUP COPIES IS INCLUDED AS PART OF THE CONTRACT BETWEEN THE PARTIES, THEN THERE MAY BE A BREACH OF CONTRACT DEFENSE AVAILABLE TO THE VENDOR AND, IN SOME JURISDICTIONS, AN ABSOLUTE DEFENSE TO NEGLIGENT MISREPRESENTATION.

TRY TO AVOID SELF-DESTRUCT SYSTEMS! Some vendors include time or use hooks which will, unless prevented by some specific input (such as a code word upon receipt of payment, etc.), self destruct the program or, even worse, the data stored by the user. Try to avoid this type of leverage as it only leads to severe damages to the user and can result in painful claims against the vendor. There are other remedies available to the vendor (including damages and, in appropriate cases, injunctive relief). Be absolutely certain that the user understands that if the programs should be transferred to another computer or cpu (and you have a hook to the cpu) that the user knows that it will be necessary to contact you to modify the program to run on the other computer. Failure to be available at any time for that purpose could result in very substantial damages. (Sure, the purpose of the hook is to prevent unauthorized copying for use on another computer, but what happens if the user's computer goes down some midnight and it is necessary to switch over to another backup system in the meantime?)

Variation of Laws Between States

Laws vary between the states. Some states limit the language which may be used in warranties and which may be used to limit liability. Most states have adopted the Uniform Commercial Code, but many apply the UCC in different ways. Further, some states apply broad restrictions upon warranty limitations in commercial transactions as well as in personal-use transactions.

If you are satisfied with the laws of your state, then try to have your contract specify that the laws of YOUR STATE will apply with respect to all aspects of the contract. In most cases this will result in your laws being applied - but not in all cases. If you rely heavily upon advertising in another state, or if the contacts are primarily in another state, it is likely that the laws of the latter state will be applied even though you have included a clause to the effect that your own state's laws will be applied.

If you are dealing with another state's vendor or vendee, keep in mind that the possibility of federal litigation always exists. While federal laws are supposed to follow the state's laws - in most instances - you can't always rely upon that.

Attorney's Fees Provisions

Litigation generally is very expensive in commercial cases. A "typical" commercial litigation in the Los Angeles area, for example, easily could cost over \$50,000.00 (and more likely more than \$100,000.00) through the trial. If the case is more than mere "typical" the cost cannot be estimated. (Fees of several millions of dollars in some complex computer litigation have been reported.)

In some types of cases (e.g., federal anti-trust, patent infringement) the law provides for the recovery of legal fees incurred BY THE PLAINTIFF. In other cases state or federal laws may provide for attorney's fees to the prevailing party.

The determination of whether a contract should provide for attorney's fees as an element of damages is not always an easy one to make. In some states if such a provision is made, even if it provides that only one party is entitled to such fees, the courts nonetheless will award fees to the PREVAILING PARTY. But, fees can be very high - even where damages are not very high. It is not particularly unusual to find a case where the award of attorney's fees exceeds the award for damages. Furthermore, you will not have any control over the fees to be incurred by the other side. The other side could retain the most expensive lawyers in town and you could be stuck with their fees, as well as your own, if you lose!

Summary

There is no clearly defined manner by which a vendor or a vendee can be fully protected with respect to problems with software. Presently, carefully worded contracts may resent the best protection. These contracts are not easy to draw and it is difficult to assume that any particular contract will be valid for more than a specific transaction.

At the present time, SOFTWARE PROTECTION (i.e., from theft) is mostly a matter of security, care and hope.

The only "clear" thing with respect to liability and protection with respect to SOFTWARE is that there is a desperate need for legislation upon the subject and a need for uniformity among the states.

In the meantime, the best advice available is to obtain the services of competent counsel, reduce your agreements to writing, and avoid "shortcuts" from the advice you are given. You should keep in mind that this particular field of law is very new and changing every day.

As stated by the Court in the Hewlett-Packard case discussed above: "This matter from the bright new frontier of small-scale computers ..." And that was only October 20, 1980!

If your attorney tells you that he can safely protect you, you might want to keep in mind the fairly well known JSA and Tandy Corp. cases discussed above. The results are precisely opposite!

Last, but not least, keep in mind the real desirability of obtaining product liability insurance. Even if it covers you for the costs of defense alone, it could be a bargain.

COMMERICAL
LANGUAGES
SIG

Sprechen sie DIBOL Parlez-vous BASIC
Hablo COBOL



COMMERCIAL LANGUAGES SIG

COMMERCIAL LANGUAGES SIG

Chairman

Ted A. Bear
Ramtek
Santa Clara, 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

Jim Wilson
Pfizer Inc. QC. Division
Terre Haute, IN

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

Cobol Working Group Members

Keith Batzel
Crowe, Chizek & Co.
South Bend, IN

Mary Anne Feerick
Greyhound Temporary Personnel
Lyndhurst, NJ

Bill Leroy
The Software House, Inc.
Atlanta, GA

Herbert J. Matthews IV
ManTech International Corp.
Alexandria, VA

Kenneth Shay
Melvin Simon & Associates
Indianapolis, IN

Jim Welborne
Crowe Chizek & Co.
South Bend, IN

DIBOL Working Group Members

Neil Baldridge
CompuShare
Lubbock, TX

Becky Burkes
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

David L. Wyse
Professional Business Software
Dayton, OH

Marty Zergiebel
The Software Gallery
Brookfield, CT

RPG Working Group Members

Keith Batzel
Crowe Chizek & Co.
South Bend, IN

Ted Bear
Ramtek
Santa Clara, CA

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

User's Guide

With this issue, the CL (Commercial Languages) SIG starts monthly publication of its newsletter, the Seal of Approval. The beginnings of today's CL SIG are chronicled by Beverly Welborne in her first contribution, The Birth of a SIG. CL SIG's involvement with standards will be highlighted through contributions from SIG members who also serve on various standards committees. This month's article in this area comes from Bruce Gaarder, a member of the X3J4 ANSI COBOL committee. RPG users are directed to articles by DEC's Shirley Stern and CL SIG's RPG Working Group Coordinator, Keith Batzel. Stupid Questions by Robert Fisher is a article aimed at BASIC programmers.

From CL SIG's first Symposium (Spring 85) comes Gene Land's magic in the It's Magic column.

And from Ted's keyboard comes More Languages Not Included in the Commercial Language SIG, a sequel to a highly successful DECUSCOPE article. Ted also serves as the Chair of the CL SIG.

Dear Language Sensitive Editor

Do you have questions or comments on the CL SIG or software products supported by the CL SIG? In future issues, replies to questions and/or comments submitted by readers will be published in this column. Address your questions and/or comments to:

Dear language sensitive editor
c/o Jim Wilson
Pfizer, Inc., Quality Control Division
P.O. Box 88
Terre Haute, IN. 47808

Past Wizards

All Wizards of the past from the BASIC, COBOL/RPG, and DIBOL SIGs are invited to submit their magic for publication in future issues. Submissions should include your name, address, and a daytime phone number as well as the symposium and SIG where you first presented your magic. Submissions should be sent to:

Jim Wilson
Pfizer, Inc., Quality Control Division
P.O. Box 88
Terre Haute, IN 47808

Birth of a SIG: By Beverly Welborne

If you once had a pink or blue foil wrapped chocolate cigar (it made a great souvenir for the kids and they probably ate it) or if you still have your airline size bottle of liquor tagged with a 12-12-84 date, then you were one of the nearly 200 new SIG members who made history by welcoming the "new arrival" at the Anaheim Fall Symposium: the brand new CL SIG.

It all started back in Las Vegas in October of 1983 when Jim Welborne, at that time Chair of the COBOL/RPG SIG, recognizing their common interest in business oriented languages, proposed to the BASIC SIG the possibility of joining together. Knowing that a better alignment with Digital's structure would enable them to have a better voice in the development and problem solving cycle and that higher levels of management would interface with a larger combined SIG, the two SIGS began negotiations. It was soon apparent that a welcome side effect would be a reduction in the expenses that each SIG incurred through duplication of effort.

Dan Esbensen, Chair of the BASIC SIG, was interested in the merge, but felt that there were several ramifications of a union: Would there be a loss of influence on the SIG Council? Would there be a reduced ability to serve the users needs? How would limits be placed on symposium sessions? These were questions that a Steering Committee could best address, so each SIG Chair consulted separately within his SIG and prepared to meet at the Cincinnati Spring Symposium in 1984.

By this time convinced that they had a good idea, the two SIGS approached DIBOL/BUSINESS SIG and APL SIG to encourage them to consider the merits of the issue. At the conclusion of the meeting a stalemate was reached with DIBOL/BUSINESS SIG and APL SIG voting against a merger and BASIC SIG and COBOL/RPG still in favor of joining together.

The situation was finally resolved during a combined Woods meeting in Boxborough, and, in the words of the former Chair of the DIBOL/BUSINESS SIG, Becky Burkes, "Because of the dynamic nature of DECUS and the Special Interest Groups, the high level languages (COBOL, BASIC, DIBOL and RPG) were able to combine to further benefit the membership. The Business portion of the former DIBOL/Business SIG formed their own Special Interest Committee, which will soon become a new Special Interest Group (Business Applications SIG)."

Keith Batzel, RPG Language Co-ordinator, and Jim Welborne, SIG Council Vice-Chair, well remember the turmoil involve in choosing a name for the new SIG. When "certain people from the COBOL group" suggested Common Business Oriented Languages SIG, with the acronym formed being the preferred name, the BASIC people countered that there was a "basic" misunderstanding here in the direction the new SIG was to take. Two of the more serious suggestions were Business Languages, rejected because this would not include those who use these languages within an academic environment, and Commercial Languages and Tools, which not only indicates business but also includes the commercial applications of these languages as used within the academic world.

A temporary steering committee (including many of those presently serving on the CL Steering Committee) was appointed to draft a statement of purpose, objectives and operating procedures
CL-2

```

;*****
;
; Print Negative Numbers Within "(12345)" Instead of 12345-
;
;*****
; Define Numbers
;*****
Record
      ,d8      ,12345678
      ,d8      ,00456789
      ,d8      ,12121212-
      ,d8      ,24325432
      ,d8      ,98796542-
      ,d8      ,87652031-
      ,d8      ,33344234
      ,d8      ,00298325-

;*****
; Redefine Numbers As An Array*
;*****
Record      ,x
      Numbers ,8d8

;*****
; Misc. Field Descriptions *
;*****
Record
      Idx      ,d2
      Mask1    ,a12      ,"ZZZZ,ZZZ.ZZ "
      Mask2    ,a12      ,"(((,((((("
      Alpha    ,a12

;*****
; Procedure Division *
;*****
Proc
;*****
; Open Terminal *
; Clear Pointer *
; Set Money Value To "(" *
;*****
      Open (1,I,"TT:")
      Clear Idx
      Xcall Money "(")

;*****
; Major Print Loop *
;*****
Next,      Incr Idx
      If Numbers(Idx).lt.0
          Then Alpha = Numbers(Idx)      ,Mask2
          Else Alpha = Numbers(Idx)      ,Mask1
      Writes (1,Alpha)
      If Idx.lt.8 Goto Next

;*****
; End Of Program Cleanup *
;*****
      Close 1
      End

```

CL-3

**It's
 Magic
 Submitted by
 Gene Lang
 Wilson-Hurd
 Wausau, WI**

to present to the SIG Council for approval. During Leadership Week in New Orleans in October, 1984, the charter was approved and the word "Tools" was deleted from the name to avoid confusion with the Languages and Tools SIG already in existence. After this initial approval, events moved rapidly at the symposium in Anaheim: the Management Council and Board met on Monday and Tuesday; on Wednesday, December 12, 1984, the new SIG was announced in the Update. Daily and all members were immediately invited to attend a reception to welcome the "new arrival" that evening.

As with any joyous occasion, especially one anticipated and labored over for so long a time, there was much cause for celebration, and so, a gala reception with all the trimmings was held in the CL Suite at the Anaheim Marriott. Pink and blue decorations festooned the halls and cigars and liquor bottle favors, served in a baby's bathtub, were distributed, followed by an elaborate buffet. To top the evening, Keith Batzel proposed an appropriate Champagne toast from atop his chair welcoming the "new arrival we delivered" and inviting each language represented to cheer in his/her language. What a polyglot group! If the success of a party is measured by the number of people who lean out of their hotel rooms to see what the heck that Tower of Babel is down the hall, then we had a very successful party!

Although the CL SIG is technically oriented in terms of dealing with computer questions, we take a holistic approach to the commercial shop and we deal with questions of cost effectiveness, efficiency, and long term usage benefits. This approach and the magnitude of the changes that have come with the merger have together attracted a growing number of new members and many of the expected benefits have materialized: there is better representation in the standards area; there is reduced overhead; the quality and quantity of the symposia and pre-symposia seminars have been maintained; the variety of the members has produced increased dynamics between users of various backgrounds; liaison between Languages and Tools and the CL SIG has been established.

Rocky Hayden, CL SIG at large steering committee member and facilitator at the later organizational meetings, has recently commented that the CL SIG "has proven a positive force in getting good positions in symposia scheduling...I think we are now structured more along Digital's structure which should mean a more active interface with DEC." So when in Anaheim we raised our glasses to toast the "new arrival," in a larger sense we were toasting each other and our multiplicity of strengths and talents and our ability to give life to our ideas. For after all, DECUS is DEC and US.

CL-4

VAX Users: Get Rid of Your System/3 or System/34 By Shirley Stern DEC

Are you a VAX user who has held on to a System/3 or System/34 to run your RPG II applications?

Now the Commercial Languages SIG and Digital can help you move these RPG II applications to VAX. The SIG has scheduled several sessions for the Fall 1985 DECUS symposium to be held in Anaheim, California the week of December 9.

RPG II is the second most popular commercial programming language used in data processing applications. Developed by IBM in the days of the 1401, RPG became very popular with the introduction of the System/3 and System/34. And because of its ability to manipulate large databases easily and generate numerous reports quickly, RPG II applications are found in many record keeping and accounting functions. Both small businesses and departments of larger companies have kept System/3s and System/34s with RPG II to meet these needs.

Digital has been offering VAX RPG II for the entire VAX family since March 1984; the product has acquired a reputation for quality from a loyal base of users. The Commercial Languages SIG has responded to these users by designating this Fall 1985 symposium as the one for emphasizing the importance of RPG II to VAX users.

Several sessions are being offered by both users and Digital. User sessions will discuss subjects such as structured RPG II programs and a cost analysis of migrating to VAX. Digital sessions will include: advanced features of VAX RPG II (including a super-productive editing environment) and tools and techniques for RPG migration. Digital has recently announced Migration Services -- a new, cost-effective method for customers to convert from IBM. Sessions are suitable for both technical and managerial members.

So plan to attend the Fall Symposium and make your reservations early. There are many other exciting sessions planned as well as the RPG II ones to make this symposium the best one yet.

RPG Working Group News

Hi!

Will you be in Anaheim for the DECUS Symposia in December, 1985? The RPG Working Group of the Commercial Languages SIG needs your support to help make this symposia a milestone event in the DECUS involvement with RPG and other related products from Digital. Will you be a session chair or a campground "counselor"?

Last year at Anaheim we had a strong showing of RPG sessions and interest from the attendees. This year we have two user given sessions planned in addition to the Digital sessions, a magic session, and a Q & A. Let's keep the momentum going.

Bring your session ideas for the next symposia to the "Joy of RPG" magic session. If RPG is a language of choice in your commercial shop then share with others the benefits you have gained by its use by putting on a magic act.

Also, if you are working with RPG Version 2 or any other RPG related field test please contact me. This working group and the SIG in general want to work with you to demonstrate our interest in RPG and all of the Commercial Languages from DEC.

The momentum for symposia is in place. Your presence and help is needed there. But, there are also non-symposia activities such as articles for this newsletter. Please call me at (219) 232-3992 or write to the editor if you can be an active DECUS member in the RPG or other working groups of the Commercial Languages Special Interest Group.

Keith N. Batzel
RPG Working Group Coordinator

ANSI COBOL Report

Submitted by: Bruce L. Gaarder
Commercial Languages SIG ANSI X3J4 Representative

The question that I am most frequently asked about the work of the ANSI COBOL Committee is: When will the new standard be adopted? I am happy to report that it might happen before the end of the year.

A brief summary of what has happened in the meetings since the last report was published in the COBOL SIG newsletter:

1. The X3J4 committee studied all of the 2,000 public review comments on the first draft proposed standard and made many changes to accommodate the users. Travelers Insurance and the Data Processing Management Association mounted a form letter campaign protesting the changes which would be hardest on IBM shops, especially those still running COBOL-68 programs. About 1,000 of the comments received were of the form letter variety.

2. The original draft was then withdrawn and a second draft was published in June, 1983. This was a procedural step, and does not mean that we forgot about the original comments, it just lets us focus on the comments on the draft standard as revised. Public comments were again sought and about 900 were received. Once again, about half were form letters. All comments were reviewed, including a twenty page submission from Czechoslovakia.

3. Public comments were invited on the substantive changes (non-editorial) made to the second draft in February, 1984 and about 225 were received. The committee has since dealt with all of these and not made any further substantive changes. Most comments that we have received in response to our answers to the 225 are of the flavor "The problems remaining are not of such great concern to warrant delaying the standard any longer."

4. The committee is currently voting on whether to send the draft proposed standard to X3 for it to vote on forwarding to the Bureau of Standards Review for adoption. This is a procedural step which considers whether the committee has done all that it can to satisfy objections and whether proper procedures were followed.

5. The International Standards Organization is proceeding with voting on adopting the draft proposed COBOL standard as an ISO standard, and we are trying very hard to keep the two the same.

6. The committee has also developed a draft proposed COBOL interface to the draft proposed ANSI Network Database Language and published it in a COBOL Information Bulletin (CIB) for public comment. This is somewhat controversial, because many people believe that no vendor will implement this database standard when relational seems to be the coming thing.

7. The committee will be studying the draft proposed ANSI Data Dictionary standard, and I will be looking at how it compares to DEC's CDD. The draft proposed ANSI Structured Query Language standard and its interface with COBOL is also being studied.

8. We are currently considering how to make the revision process take less time (it has been 11 years since the last standard was published.) One likely approach is that of adding modules which are upwardly compatible from the existing standard, such as the VALIDATE concept. This approach has already been approved by ISO.

9. Many of the new features have shown up in newer DEC compilers and have been well received for the productivity increases that they yield. A list of the four items which still are of concern to the greatest number of vendors/users follows:

- a. When a receiving item is a variable length item and contains the object of the OCCURS DEPENDING ON phrase, the maximum length of the item will be used. (This is of most concern to IBM users who have coded an OCCURS DEPENDING ON after another in the same record, which would be wiped out. This is prohibited by the '74 standard, but IBM still allows it.)
- b. The ALPHABETIC test includes upper case and lower case letters and the space, rather than upper case letters and the space. (DEC's compilers already do this.)
- c. Within the VARYING ... AFTER phrase of the PERFORM statement, identifier-2 is augmented before identifier-5 is set. (This was previously undefined, and it is believed that this will enhance program portability.)
- d. Want complete upward compatibility, language conversion programs, and flagging of statements which might produce different results under the new standard. (Generally not possible, since the last two are vendor-specific, and the first would preclude clarifying ambiguous and undefined items, since different vendors might have interpreted the item

?
Stupid Questions
?

by Robert Fisher

In the August '83 issue of BASIC SIG. Shel Jones shared an anecdote about the need all systems have for fundamental maintenance (p.3). A similar experience of mine involved a user who each Monday would call and ask, "My computer's broken, could you help me?" After my first week I learned she did not have a Mini, Micro, or a PC; nor a VAX, PDP 11, or a DecMate; no, what she had was a VT100 with the power switch off. So often unwanted questions reveal stupid problems bordering on the marvelous.

Digital's BASIC supplies an equally stupid question with its INPUT statement. With repeated use of this command one becomes more and more driven to ask, "Why can't that insipid question mark be suppressed?" Applying INPUT to an error handler which requires a response from the keyboard (as in ". hit return to continue ?") provokes many questions. What is the purpose of the question mark? How is it to be interpreted? Is the process asking a question or merely thinking out loud?

Clearly, the INPUT statement creates an embarrassing situation. If I know "hit return to continue?" doesn't make any sense, how can I explain it to my clients? In Microsoft Basic this confusion can be avoided by suppressing the question mark with a simple semi-colon at the end of the command:

```
INPUT " .. hit return to continue ";DUMS;<---**
```

Question: as we approach the end of the eighties, should a two bit, el cheapo home computer out perform our own very friendly VAX? Answer: Digital should follow the trend of "New-conservatism" and get Big Brother off all our backs by allowing us to be responsible for our own question marks. I don't need Digital supplied punctuation on my input routines, thank you very much.

After calmly discussing our question mark we see that it beckons the elegant programmer to experimentation. (By "elegant" I mean those well rounded individuals who want to talk to their users in a sensible manner.) For convenient reference, allow one method of avoiding DEC's foolish question mark to be executing a separate command to print out our prompt. We will then request a GET statement on lun 0:

```
PRINT " .. hit return to continue .. ";  
WAIT 60%  
GET #0  
WAIT 0%
```

So far, so good; though "not supported by Digital for terminal format files" the GET will accept the line terminator without the programmer even needing to set up a

buffer. The message has been acknowledged and the program can continue. Now let us expand upon this construct to receive meaningful input from the screen, as in the typical inquiry "Are you sure you want to delete this?":

```
PRINT "Verify delete (Y): ";  
WAIT 60%  
GET #0  
WAIT 0%  
MOVE FROM #0, GO$=1  
RETURN IF GO$<>"Y"
```

Here we have an effective route around the question mark roadblock; we can tell the operator what to do without the rude intrusion of that puny piece of punctuation (DEC's question mark). The road we've taken displays a more direct message for the user to read and is consequently less confusing. Unfortunately, though, our detour is not without problems because using the GET on lun 0 can end up on a treacherous stretch of dirt road. The air is clear and the ride is smooth while accepting character data, but things become a bit rough when one asks for integer input.

To illustrate, suppose we need a quantity for an order, and we choose to follow this path:

```
PRINT "# of items: ";  
WAIT 60%  
GET #0  
WAIT 0%  
MOVE FROM #0, QUANTITY%
```

Seems okay, right? Well, I thought so too. But when it came time to travel on this highway I found myself routinely fixing a flat tire. If you respond here with a null input (just hit return), the value in QUANTITY% will be 2573. Our little country road will happily transform 0 (type 0, then return) into 658736. Figure 1 displays more routes I took during this vacation. Fellow travelers beware: after several tire changes, I found my MOVE statement to be causing all the blowouts, and what follows is my report to the travel club.

The MOVE loads the buffer character by character and then assigns that bit map to the integer variable. Our result is not a binary representation of a number but rather an integer longword filled with four bytes of ASCII code; i.e., our result is a QUANTITY% of garbage. The number stored in QUANTITY% is purely a function of the ASCII code; type in a letter or two, don't worry, the MOVE will still give us a number; anything may be typed because only the character string for the first four bytes is going to be loaded. Add to this that the buffer (if the number of characters input is less than 4) will contain a carriage return-line feed sequence, and we must conclude that the whole procedure is completely unreliable (but very, very interesting) Hence, in this scenario 2573 is a longword loaded with two high order nulls and a low order line feed, carriage return (see Figure 2 for more examples) BASIC

treats the TT: as any other file and simply assigns the bits to a storage location. In VAX Basic, version 2.2, it is not possible to avoid using the INPUT statement and also read integer information directly from the terminal. Using a GET statement with a MOVE on lun 0 to accept integer values will always lead to a corrupted data set.

To use BASIC for inputting an integer one must receive the data as character, and then use a VAL(EDIT\$(BUF\$,-1\$)) combination. The TRM\$() function is not an option here because it doesn't trim the CR and LF. Now we have a method to receive all types of data without the noisy intrusion of many stupid questions conveniently supplied by Digital Equipment Corporation. Here's another example of DEC going out of their way to meet the needs of their customers, but wouldn't life be easier if we had some method to suppress that foolish, insipid, crudely embarrassing, socialistic, blatantly passe', gnat-like, leave-me-alone-I-don't-want-any, utterly meaningless and very, very STUPID little question mark?

?

Robert Fisher

Display Program 1

```

10  ON ERROR GOTO 200

15  CZ = CTRLC
    PRINT \PRINT
    INPUT ' Show Mask. ';mm$
    PRINT \PRINT
    MZ = 0Z
    MZ = -1Z                                IF MM$<>'N'

50  PRINT 'NO_STUPID_QUESTIONS>';
    GET #0
    MOVE FROM #0, HAZ

    MASK$ = ''
    BBZ = 1Z
    FOR IZ = 30Z TO 0Z STEP -1Z
        BRZ = + 1Z
        BB$ = "0"
        BB$ = "1"          IF HAZ AND (2Z**IZ)
        MASK$ = MASK$ + BB$
        IF BBZ = 8Z      THEN
            BRZ = 0Z
            MASK$ = MASK$ + ' '
        END IF
    NEXT IZ
    BBZ = "0"
    BB$ = "1"
    MASK$ = BB$ + MASK$      IF HAZ<0Z

    PRINT ' This is the value loaded: 'haZ;' -- '
    PRINT '          ';MASK$ IF MZ
    PRINT
    HAZ = 0Z
    GOTO 50

200  IF ERR=11 THEN RESUME 300
    ELSE IF ERR=28 THEN RESUME 15
    ELSE PRINT 'ERROR ';ERR,ERL      \RESUME 300

300  END

```

Display Program 2

```

10  ON ERROR GO TO 32000

    DECLARE LONG  TT_CHN,      ! TERMINAL OPEN CHANNEL      &
                  EDIT_MASK   ! EDIT MASK

    TT_CHN       = 57Z
    EDIT_MASK    = 4Z + 32 + 64
                  ! 1 TRIM PARITY BIT
                  ! 2 DISCARD ALL SPACES, TABS
                  ! 4 DISCARD EXCESS CHARACTERS:
                  !   CR,LF,FF,ESC,RUBOUT,NULL
                  ! 8 DISCARD LEADING SPACES, TABS
                  ! 16 REDUCE SPACES, TABS TO 1 SPACE
                  ! 32 CONVERT LOWER TO UPPER CASE
                  ! 64 CONVERT [ TO (, AND ] TO )
                  ! 128 DISCARD TRAILING SPACES, TABS
                  ! 256 DON'T ALTER CHARTERS IN QUOTES

    OPEN 'SYS$INPUT:' FOR INPUT AS FILE TT_CHN

50  PRINT ' Show Mask ' ;      ! PRINT THE PROMPT
    INPUT LINE #TT_CHN, MM$    ! GET THE INPUT
    MM$ = EDIT$(MM$, EDIT_MASK) ! EDIT THE INPUT

100 Z$ = INSTR(1Z, MM$, CHR$(8Z)) ! SEARCH FOR BACKSPACES
    IF ZZ <> 0Z THEN              ! REMOVE BACKSPACES
        MM$ = LEFT(MM$, ZZ - 2Z) + RIGHT(MM$, ZZ + 1Z)
        GO TO 100                ! GO BACK FOR MORE
    END IF

200 ST$ = EDIT$(MM$, 2Z)        ! REMOVE ALL SPACES
    POINTZ = INSTR(1Z, ST$, '--')
    ST$ = '--' + LEFT(ST$, POINTZ - 1Z) IF POINTZ = LEN(ST$)
        ! MOVE THE MINUS SIGN IF IT
        ! IS ON THE END
    HAZ = VAL(ST$)              ! CONVERT STRING TO REAL

300 PRINT ' This is the value loaded: 'haZ;' -- '
    PRINT '      ');HASK$
    PRINT
    GOTO 50

32000 IF ERL = 200 THEN
        PRINT ERT$(ERR);' CONVERTING <<' MM$; '>>'
        RESUME 200
    END IF

    PRINT 'UNKNOWN ERROR ';ERT$(ERR);' AT ';ERL
    RESUME 32767

32767 END

```

CL-13

MORE FOOLISHNESS

We have told you about C-, DOGO, FIFTH, LAIDBACK, LITHP, REGAN, RENE, SARTRE, SIMPLE, SLOBOL and VALGOL. Just when you thought it was safe to go back to your terminal, we have NEW LANGUAGES II.

More
Languages not included in the Commercial Language SIG
or the Languages and Tools SIG.

by

Ted A. Bear
In the heart of Silicon Valley

and

A Usually Reliable Source
Digital Equipment Corporation
Somewhere in New England

BEAN COUNT 1,2,3

BEAN COUNT 1,2,3 is an artificial intelligence language for accounts. The language has one command BOTTOM LINE. All one needs to do is supply the bottom line and BEAN COUNT 1,2,3 makes all of the other numbers fit.

CLOSE ENOUGH

This language is replacing ADA in many sites because it is CLOSE ENOUGH for government work. Commands in CLOSE ENOUGH are obscure and verbose.

The military version contains SECRET CODE (some so TOP SECRET that no one can use them except for teen age hackers who gain illegal access) while the CIA version contains SECRET AGENTS.

FUN-4-ALL

FUN-4-ALL was developed by Dr. Fred Rogers (of Mister Roger's Neighborhood) with the main idea to keep the language slow and simple. You must wear a sweater and blue deck shoes to program in this language.

CL-14

This language is so friendly that commands are called suggestions. If you input a incorrect suggestion you might get a response like this:

```
$ HELO
  Hello boys and girls.
  I see that you entered a suggestion.
  I am sorry, but I can not recognize HELO.
  Could you have meant HELLO?
  Can you say HELLO?
$
```

Some other suggestions are:

```
TOMORROW
FUN-4-ALL sings TOMORROW, because that is when you will get your
results.
```

```
TROLLEY
An input suggestion that takes your input into the land of make
believe.
```

This language is also very courtious, all input/output devices are address as Mr. or Miss. (Mr. DRB1:). A FUN-4-ALL user would never address anyone or anything by first name.

FUN-4-ALL will not run on Friday the 13th.

RIP

RIP is a stable product designed for programmers who are dead or have have moved up into non-technical management (which is about the same thing.)

RIP contains only termination commands.

```
STOP, END, FINISH, TERMINATE, PUT_TO_SLEEP, GONE_AWAY,
BIG_SLEEP, HIBERNATE, GREAT_REWARD, LAYOFF, FIRED,
NO_LONER_WITH_US, HAPPY_HUNTING_GROUND
```

(Editor's Note: If you have any suggestions for future programming languages that may not be included in either the CL or L&T SIGs, send your suggestions to:

```
Jim Wilson
Pfizer, Inc., Quality Control Division
P.O. Box 88
Terre Haute, IN. 47808)
```



D A A R C

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

Introduction

If you've been watching DECUS for awhile, you may have noticed the disappearance of the LABS SIG. It has been replaced by DAARC. DAARC stands for Data Acquisition, Analysis, Research, and Control. This article should, once and forever, dispell any doubts as to just what DAARC is all about. The change was made to encompass the variety of interests of the users, who, though practicing their crafts on a variety of levels, have similiar problems, desires, and interests. In short, DAARC hopes to meet the needs of anyone who uses the computer to collect data, scrutinize data, or control processes (whether this be a giant paper mill or an instrument in the laboratory).

Data Acquisition

Problems encountered in data acquisition are varied and yet can be very similiar. Whether one is acquiring data from many sensors in a large industrial process or from a few in a laboratory experiment. Many considerations are parallel, such as collection rate, noise, availability of a signal, and others. DAARC provides a forum for discussion of these problems as well as an interface to Digital for an exchange of information regarding user needs in this area. Sessions at DECUS symposia are one forum, but this publication is yet another!

Research

Ask anyone who is employed in a research laboratory and they will probably tell you that their environment is unique. Add a computer and it is unequalled. DAARC provides a forum for research and laboratory types to exchange ideas and needs. It also provides an interface to Digital for their Laboratory Data Products line. Another area upon which DAARC is focusing on includes LIMS (Laboratory Information Management System). As Digital and other vendors produce more software to meet the needs of the laboratory, sessions to exchange information are held at the DECUS symposia.

Analysis

No matter what operating system you use, or no matter how you have collected your data, sooner (or sometimes instantly) or later, some analysis of the data must occur. It is DAARC's hope to provide a forum for the exchange of analytical techniques to aid in the analysis. Whether mathematical or statistical (or just plain common sense), all should be able to be presented and discussed freely. Note that DAARC provides an "operating system free" forum thus divorcing the application from the operating system.

DAARC also sponsors sessions on statistical analysis software packages, such as RS-1. Users of this system can readily share ideas, experiences, and needs.

Control

There are an increasing number of users involved in process control and industrial automation. These users, too, need a forum for exchange. DAARC sees to provide a forum through sessions scheduled at DECUS symposia and provides an interface to Digital concerning products in these areas.

Making it all work

To make all this happen, DAARC has a steering committee which plans the direction that the SIG takes. One of the Steering Committee members takes charge of finding and scheduling sessions for the symposia while others, each from a particular area of interest, are responsible for this publication, session notes, and interfacing with the rest of DECUS. The Steering Committee also serves as a source of information for you, the DECUS member. In most cases, if one of them cannot answer your questions, they can, at least, point you toward someone who can.

But the real force behind DAARC or any SIG, is you the DECUS member. Now that some light has been shed on DAARC, if you would like to be involved, here are some suggestions:

- 1) Remember- There is more to DECUS than the Symposia!
This publication should serve as a timely forum for a lively exchange. DAARC's editor eagerly awaits your input. Long technical articles would be great- but short comments on issues can also prove stimulating.
- 2) If you would like to make a presentation or be involved in a panel or whatever you can do at a symposia, let us know. Any Steering Committee member will be glad to help you.
- 3) If you are interested in becoming part of DAARC's leadership, please contact any Steering Committee member.

One Final Note

Just rember that, "It's always DAARCEst just before the dawn". Here's to the dawn of a new era in increased exchange of information in DAARC (and DECUS) via this publication.

DMS

DMS

D

M

S



DMS

DMS

DMS

Chairman

Steve Pacheco
Ship Analytics
North Stonington, CT

Chairman Elect.

Joseph F. Sciuto
Army Research Institute
Alexandria, VA

Symposium Coordinator

Keith Hare
JCC
Granville, OH

Symposium Coordinator

Barbara Mann
TRW
Redondo Beach, CA

Session Note Editor

Mark Morgan
Farm Credit Banks
Springfield, MA

RMS Working Group Coordinator

Allen Jay Bennett
Lear Siegler Rapistan
Grand Rapids, MI

Forms Working Group

Debbie Kennedy
Land Bank National DP Center
Omaha, Nebraska

MIS Working Group

Sandy Krueger
Key Financial Systems, Inc.
Pine Brook, NJ

Database Working Group

Jim Perkins
PSC, Inc.
Shelburne, VT

ADE Working Group

Penny Peticolas

Working Group Coordinator

CDD Chair/Prod Group Coordinator

Alan Schultz
Land Bank National DP Center
Omaha, NE

Forms Working Group

Nancy Torok
Credit Suisse Bank
New York, NY

ADE Working Group

William H. Irving
FLB of Sacramento
Sacramento, CA

AI SIC Liaison

D. Slater
Institute for Defense Analysis
Alexandria, VA

Datatrieve Liaison

John Schutt
J.R. Simplot Company
Boise, ID

Black Book

David B. Turner
Korn/Kerry International
Los Angeles, CA

DEC Counterpart

Shirley Schneider
Digital Equipment Corporation
Nashua, NH

Member-At-Large

Julie Llewellyn
United Technologies
Colorado Springs, CO

Member-At-Large

Doug Dickey
GTE Government Systems
Rockville, MD

Member-At-Large

Richard Arndt
Cognos Corporation
Houston, TX

Member-At-Large

Larry W. Hicks
Relational Database Services
Kernersville, NC

SIG Chair's Message

I would like to welcome all readers to this new publication. For those of you who have received DECUS Newsletters in the past you will recognize this as a new format; for those of you who are new to DECUS Newsletter publications, the Special Interest Groups look upon this new newsletter format as both a challenge and an opportunity. The OPPORTUNITY is to provide another communications mechanism between you, THE USERS, and DECUS.

A Special Interest Group is composed of people, each having his own particular interest. As a group we have a particular set of goals and I believe that it is important that there is an understanding of those goals. The main goal of the DMS SIG is COMMUNICATIONS. We wish to provide the opportunity to exchange information on both technical and non-technical topics. A simple statement of our goals is:

The interest areas of those involved in the Data Management Systems SIG are varied, but relate to the use of data management tools to provide access to data. In general we are concerned with access to substantial amounts of data, or problems relating to protection, integrity and remote retrieval of the stored data.

This statement provides a great deal of latitude, opportunity, and sometimes problems. The DMS SIG works closely with other SIGs to accomplish the main goal of COMMUNICATIONS. The communications we wish to foster are:

User to User, User to Digital, Digital to User

Users of Digital Equipment Computer Systems have many tools available, those produced by Digital as well as those produced by others. It is important to everyone that each of us get our jobs done in the most effective way possible. If DECUS can provide the means by which discussion with other users or Digital can aid in accomplishing a task or improving the general understanding of a topic, then we have succeeded.

The Data Management Systems SIG has put the majority of its effort into symposia related activities. These activities have been received well and have provided a great deal of opportunity for people to interact. There have been many very good sessions presented at past symposia, as well as excellent dialog among users and Digital in the DMS Campgrounds. I believe though, that DECUS is at a crossroads of change, and the DMS SIG must now evaluate where its efforts are placed and how best to provide communications. We WILL be a regular part of this new Newsletter format; and we welcome comments and questions from YOU the readers. We are looking forward to suggestions, articles and a Q&A column. Remember that communications requires a two way channel, both ends must speak as well as listen.

The Data Management Systems SIG will continue to discuss any and

all topics relating to the very broad area of Data Management. This includes both Digital and non-Digital hardware and software, as well as how data management relates to the successful running of a business. The "WHY" of data management is as important as the "HOW" or "WITH WHAT" issues.

For the upcoming Fall Symposium we have a series of exciting sessions as well as the Campground. We have scheduled both technical and non-technical sessions, and I am sure there will be a number of adhoc sessions scheduled during the week. The presentors include users, people from Digital, and other companies which supply hardware and software. Our campground will be open from 9:00 AM until early evening Monday thru Thursday. We always have a good group of both Digital and non-Digital people onhand to discuss almost any topic.

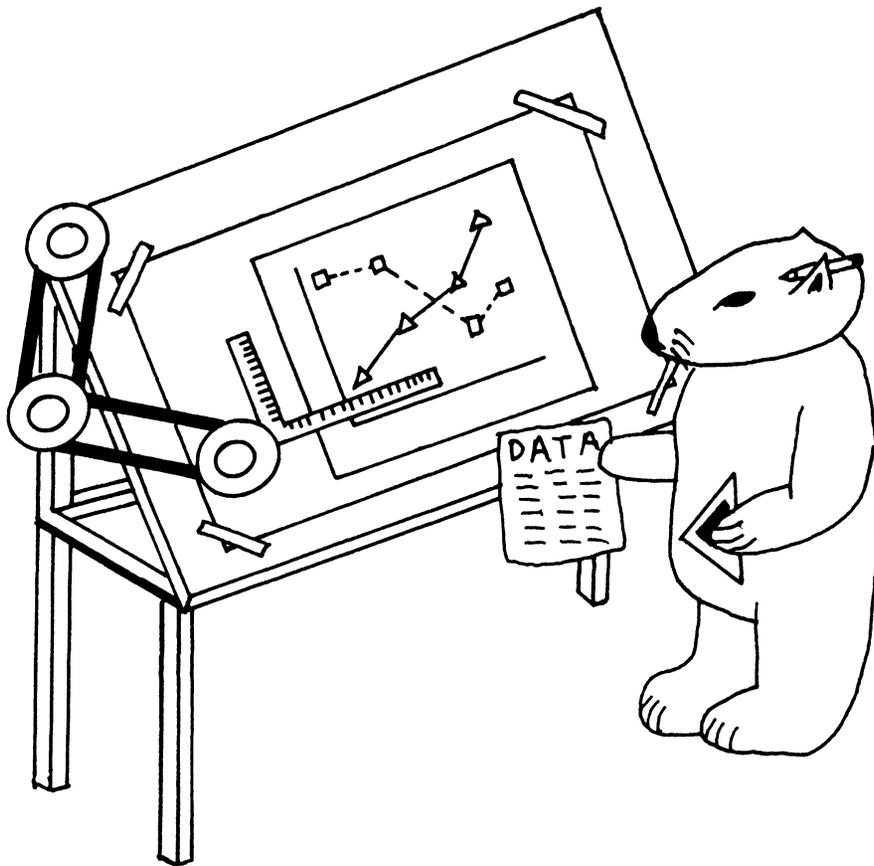
To conclude my message in this first issue of the new DECUS Publication I would ask you to consider this as another medium of communications. Send us your thoughts, questions, and concerns. We will do our best to publish them and get answers. If you would like to talk to me directly my office phone number is (203) 535-3092. We of the DMS SIG do want to hear from YOU the users.

Stephen P. Pacheco
DMS SIG Chair

The Wombat

EXAMINER

"Increases the Circulation of Anyone in America"



DTR & Graphics

DATATRIEVE Special Interest Group - Officers

7-21-85

SIG Chairman	Larry Jasmann	U. S. Coast Guard	10067 Marshall Pond Rd.	Burke, VA 22015	202-426-2344
Communications					
Committee Rep.	Elaine McWilliams	Boeing Company	P.O. Box 24346 M/S 9c55	Seattle, WA 98124	206-773-0102
Newsletter Editor	Joe H. Gallagher	Cleveland Clinic Foundation	9500 Euclid Ave.	Cleveland, OH 44106	216-444-2551
Newsletter Assoc. Editor	Steve Cordiviola	Kentucky Geological Survey	311 Breckinridge Hall	Lexington, KY 40506	606-257-5863
Newsletter Assoc. Editor	Donld Stern	Warner Lambert Company	10 Webster Road	Milford, CT 06460	203-878-9351
Artist and Media	Bart Z. Lederman		2572 E. 22nd St.	Brooklyn, NY 11235	212-250-2300
Symposia					
Symposia Committee Rep.	Cris Wool	E.I. duPont Eng. Dept.	Louviers Bldg.	Wilmington, DE 19898	302-366-4610
Symposia Assitant	Diane Pimney	Computer Sciences Corp.	443 Inyokern Road	Ridgecrest, CA 93555	619-375-4610
Pre-Symposia Seminars	Dana Schwartz		15719 Millbrook Lane	Laurel, MD 20707	301-859-6277
Symposium Notes	Elaine McWilliams	Boeing Company	P.O. Box 24346 M/S 9c55	Seattle, WA 98124	206-773-0102
Library Committee Rep.	Kathy Tamer	Univ. of Texas Med School	6431 Fannin RM 5.020	Houston, TX 77030	713-792-5566
Hardware & Black Book	Bert Roseberry	COGD8 (dt)	500 Camp Street	New Orleans, LA 70130	504-589-4934
Special Effects	Alex L. Lamb	U. S. Army	16 Wake Road	Eatontown, NJ 07724	201-532-3318x1843
Special Consultant	Phil Naecker	J M Montgomery Engineers	250 N. Madison Ave	Pasadena, CA 91101	818-796-9141

DATATRIEVE Masters List

3-15-85

RSX & PO/S	Bart Z. Lederman		2572 E. 22nd Street	Brooklyn, NY 11235	212-250-2300
RSX & VMS	Joe Kelly	Waymon Gordon	Worchester Road	North Grafton, MA 01536	617-839-4411x5480
IAS & VMS	Chuck Watson	Battelle Northwest		Richland, WA 99352	509-376-2227
VMS	Larry Jasmann	U. S. Coast Guard	10067 Marshall Pond Rd.	Burke, VA 20015	202-426-2344
VMS	James Swanger	G. D. Searle & Co.	4901 Searle Parkway	Skokie, IL 60077	312-982-7430
VMS	Darrell Eade	Naval Undersea Water E. S.		Hayport, WA	206-396-2501
VMS	Chris Wool	E. I. duPont, Engineering Dept.	Louviers Bldg.	Wilmington, DE 19898	302-366-4610
VMS	Dick Azzi	Motorola SPS	Box 2953 MS: P116	Phoenix, AZ 85062	602-244-4316

DTRI

Contributions

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

Joe H. Gallagher, Ph. D.
DATATRIEVE Newsletter Editor
Manager, Information Center
Cleveland Clinic Foundation
9500 Euclid Avenue
Cleveland, Ohio 44106

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 SPRs submitted to Digital or other information for members of the DATATRIEVE SIG. Machine readable input is highly desirable and machine-to-machine transfer of material is preferred, but most anything legible will be considered. However, this newsletter is not a forum for job and/or head hunting, nor is commercialism appropriate.

Table of Contents

Volume 7, Number 1

- 2 Chairman's Corner
 - 3 From the Editor's Pen
 - 4 Article - Horizontal Bargraphs Without Graphics Devices
 - 9 Previews of Coming Attractions
 - 10 Article - VAX-DATATRIEVE Graphics Internals
 - 30 Article - Modifying VAX-DTR Plots
 - 35 Article - Additional DATATRIEVE Plots
-
-

About the Cover

The section cover for this first issue of the combined DECUS Newsletters was drawn by Bart Lederman. Graphics is the theme for this issue.

Chairman's Corner

Larry Jasmann, U. S. Coast Guard, Burke, VA

Greetings. My name is Larry Jasmann, and I am the chairman of DATATRIEVE SIG. The DATATRIEVE SIG exists to serve the informational needs of DECUS members who use or have an interest in DATATRIEVE. Additionally, the SIG has recently decided to change its name and focus slightly to include Fourth Generation Languages which deal with information systems. The process of doing that has met with initial approval of higher levels of DECUS management, and should be in effect by the Anaheim symposia. With this addition, the SIG will also be looking at products, both DEC and third party, which are used primarily for user interface with Data bases, and decision support. The extent to which the DTR/4GL SIG will expand to include other products will depend on the expressed needs, desires and energy of the membership of the SIG. If you are interested in this area, let me know, or come see us at the Anaheim symposia.

Let me introduce you to some of the services provided by the SIG.

First and foremost, DATATRIEVE SIG serves as a communication path in three areas; user to user, user to DEC and DEC to user. All of our activities focus in some way on one of these three critical communication paths. Here are some of our activities:

Symposia: At symposia, DTR SIG usually sponsors about 30 sessions covering DTR topics at all levels, from the highly technical, to basic descriptive material. We also have a campground where you can meet with other users, DEC developers, and members of the SIG steering committee. We have terminals and a PRO in the campground, and a lot of problems get solved there. We produce a wish list during the Symposia which is our way of telling DEC what we want to see added to the product. The high point of the week is Wombat Magic, a special session where the best (and sometimes the funniest) of DATATRIEVE comes out. DATATRIEVE SIG has always been known as a SIG where things are fun, but we also don't let that get in the way of getting lots of good, constructive work done. Our activities at the Symposia follow that theme.

Newsletter: We publish a regular section in the DECUS Newsletter. The intent is to reach, particularly those members of the SIG who do not get to the symposia, with as much good material as possible. Our issues, this one is typical, tend to be filled with lots of good ideas on how to use DATATRIEVE more effectively.

Hot Line: We always publish a list of people [DATATRIEVE Masters List on the back of the section cover] willing to accept calls about DATATRIEVE. The value of someone else listening to your current problem BEFORE you spend a week beating your head against a wall is incalculable.

Pre-Symposia Seminars: We always teach at least two seminars on the Sunday before the Symposia. The intent is to provide a day of intensive training by teams of real experts to boost your productivity with DATATRIEVE.

With the inclusion of other 4GL's our list of deliverables will grow (this assumes that we will find people interested enough to participate in those areas). The one thing that MUST be emphasized is that the basic spirit of the SIG will NEVER go away. We wish to remain a friendly, somewhat spontaneous group which can rapidly and effectively help each other. The Great Wombat isn't going to go away, he is just going to diversify a bit.

The theme of this issue is Graphics, and it is an area of DATATRIEVE that excites me a lot. DATATRIEVE graphics are a very powerful way to make good sense out of massive amounts of data with very simple, easily remembered commands. Enjoy!

From the Editor's Pen

Joe H. Gallagher, Cleveland Clinic Foundation, Cleveland, OH

Alphonse Karr said, "The more it changes, the more it's the same thing." (What he really said was "Plus ça change, plus c'est la même chose.") When looking back through some old copies of the Wombat Examiner to get some references for articles in this issue, I ran across issues of Volume 9 of DECUSCOPE from 1969. For those of you who were not active members of DECUS at that time, DECUSCOPE was then a combined monthly technical journal of DECUS. It seems that Karr was right; we have gone full circle. Combined newsletters are a big opportunity for specialty areas to reach a wider audience. The success of this bold undertaking will, however, depend on the readership. I encourage you to subscribe or re-new your subscription, and to submit material for publication to this and other newsletters. Articles are, of course, most desired, but Hint and Kinks, unsolved problems, and other items of interest to the DATATRIEVE community of users is welcomed.

With a more frequent publication schedule, it will be possible to disseminate information of a more timely nature. There will sections devoted to current problems, bug fixes, etc. We will continue to try to solicit and produce quality articles while maintaining the usual Wishlist, Wombat Magic, Hints and Kinks, and News sections. A current SIG officers list and a DATATRIEVE Masters List will be published monthly and will appear on the back of the section cover. In additions, a new section "Previews of Coming Attractions" will be added to let users know what to expect in future issues.

The monthly publication schedule means that there are now twelve dead-lines per year instead of just four. However, I am pleased to announce that Steve Cordiviola of the Kentucky Geological Survey and Don Stern of Warner Lambert Company have agreed to serve as Associate Editors of the Wombat Examiner. I could not be more delighted to get their excellent help.

The theme of the Wombat Examiner section of this first issue in the Combined DECUS Newsletter is DATATRIEVE Graphics. There was considerable interest in DATATRIEVE graphics at the recent New Orleans Symposium. Four articles results from that interest. It is my expectation that there will be future issues of the Wombat Examiner devoted to other specific themes.

Horizontal Bargraphs Without Graphics Devices

Bart Z. Lederman, Brooklyn, NY

In a previous article [Bar Graphs in DATATRIEVE, Wombat Examiner, Volume 5, Number 2, pp. 32-44] I showed how to obtain vertical bargraphs in any version of DATATRIEVE, on standard terminals and hardcopy printers. This article will show two ways horizontal bargraphs may be obtained. As in the previous article, these graphs are not as fancy as those obtained with VAX-DTR plots, but they don't require any graphics hardware, and will work on VAX, DEC-20, PDP-11, and PRO systems. The second method also demonstrates a method of accessing individual records within an inner list (an OCCURS clause) and the value of looking at information in more than one way.

The first method was shown to me by Ed Sweeney, and is very easy. You simply define a table similar to this:

```
DEFINE TABLE BAR_TAB USING BEGIN
"0" : " ",
"1" : "*",
"2" : "**",
"3" : "***",
"4" : "****",
"5" : "*****",
"6" : "*****",
. . .
END TABLE
```

and so on until it reaches the maximum number you wish to graph (or the maximum width of your paper). The right hand side can be filled with whatever printing character you wish, or you might get fancy and put in escape sequences to put your terminal into reverse video, etc. To graph a value you simply print it via this table. This method is certainly easy, and you can concatenate several variables into one string. It has the disadvantage that it will only graph variables which are defined as "USAGE DISPLAY" (it can't graph INTEGER or other COMP types), and it won't interpolate fractional values. Also, in DATATRIEVE-11, loading large tables use up pool space. Even so, it does have it's uses.

A more versatile method of graphing data is to define a temporary storage file and two domains to access the data in this file.

```
!
! The first domain allows access to individual characters in a bar.
!
DELETE H_STRING_REC;
DEFINE RECORD H_STRING_REC
01 H_STRING_REC.
    03 VALUE PIC 999.
    03 STRNG OCCURS 50 TIMES.      ! The width of the widest bar.
    06 CHAR PIC X. ;
```

```

DELETE H_STRING;
DEFINE DOMAIN H_STRING
  USING H_STRING_REC ON HBAR.SEQ;

!
! The second domain accesses the bar as a single field.
!
DELETE H_BAR_REC;
DEFINE RECORD H_BAR_REC
01 H_BAR_REC.
  03 VALUE PIC 999 EDIT_STRING ZZ9.
  03 BAR PIC X(50) QUERY_HEADER " ".
;
DELETE H_BAR;
DEFINE DOMAIN H_BAR
  USING H_BAR_REC ON HBAR.SEQ;    ! NOTE: SAME FILE NAME FOR BOTH DOMAINS !!

```

In the above example, I have chosen a bar 50 spaces wide, because that happens to fit well on an 80 column screen, but it can be any value you wish. Note that the number of OCCURS in the first domain must match the size of BAR in the second domain, and that both domains use the same file. For these examples, I am going to define a third domain just to hold some sample data and demonstrate the bargraphs, but in actual use the data could come from any domain.

```

!
! Third domain just stores data for sample demonstration.
!
DELETE DATA_REC;
DEFINE RECORD DATA_REC
01 DATA_REC.
  03 VALUE PIC 999 EDIT_STRING ZZ9.
;

```

All that is needed now is the procedure which reads data from a given domain (in this example, the domain name is DATA), and creates a bargraph.

```

!
! Create a new Horizontal Bar Graph
!
DELETE MAKE_H_BAR;
DEFINE PROCEDURE MAKE_H_BAR
DEFINE FILE FOR H_BAR          ! start with an empty domain

!
! Remember to purge old HBAR.SEQ files,
! or use the SUPERCEDE qualifier on the DEFINE FILE command.
!
READY H_BAR WRITE
READY DATA                    ! source of the data to graph
DECLARE BLANK PIC X(50).       ! same size as bars
BLANK = "                      " ! 50 blanks

```

```

FOR DATA STORE H_BAR USING BEGIN
    VALUE = VALUE      ! store the value to be graphed
    BAR = BLANK        ! put in an empty bar.
END
!
! Clean up pool
!
RELEASE BLANK
FINISH

!
! We now have a domain which contains one record for each value to be
! graphed, and a blank line. The next section will use the other domain
! to run down the bar one character at a time and fill in the bar.
!
READY H_STRING MODIFY
DECLARE INC PIC 999.
DECLARE TMP PIC 999.
INC = *."Maximum value of this graph" ! the maximum value of the right edge
INC = INC / 50                        ! the size of each of the 50 steps
FOR H_STRING BEGIN                    ! for each value to be graphed
    TMP = INC                          ! start at the first step
    FOR STRNG MODIFY USING BEGIN      ! this moves us across the bar
!
! If the present value is less than the value to be graphed, change
! the blank space to a printing character.
!
        IF TMP LE VALUE CHAR = "*"
        TMP = TMP + INC                ! increment to next column
    END
END
RELEASE TMP
RELEASE INC
FINISH
END PROCEDURE

```

The first half of the procedure just copies data out of whatever domain the data is in, and fills in a new temporary file. The DEFINE FILE command creates an empty file faster than finding and erasing an old domain, but it does create new files on disk, so you should purge them regularly, or use the SUPERCEDE qualifier on the command. Also, you can make this procedure run a little faster by using the ALLOCATION qualifier to set the size of the HBAR.SEQ file. When I was developing this procedure I did not supersede old files so I could look at them and see what happened when something went wrong, and you won't know what size the file will be until you run the procedure with your data. After you decide what data to graph, you will be able to go back and set the allocation to fit your application. Note that the data is in a sequential file (no keys). In this particular application, the data is always written and read through in the same order from the beginning of the file, so this is a rare case where an indexed file isn't needed (it would be slower and use more disk space in this particular application). The second half of the procedure readies the domain with the other definition which allows access to each character in the bar. The temporary variable INC is set to the maximum value to be graphed. In this example, the user is prompted for a value and I

have entered 50 for the graphs printed below, but it could also be set automatically to a fixed value or a MAXIMUM derived from the domain holding the data. INC is then divided by the number of columns in the bar (in this case, this is also 50) to obtain the amount by which the columns increase in value from left to right. TMP starts at the value of one column, and increases by one increment for each column. Having the "FOR STRNG" loop within the "FOR H_STRING" loop is the method by which you can pass through an OCCURS clause within each data record. While TMP is less than the value to be graphed, the blank space is replaced with a printing character (I used an "*", but it can be anything), otherwise the space is left blank. I have defined INC and TMP to be 999 because it fits my data, but you can define them to have whatever range is necessary. They can have places to the right of the decimal place if needed for your data.

This is all there is to it; the domain is how ready to print.

```
READY H_BAR
PRINT H_BAR
```

VALUE

```

1 *
5 *****
10 *****
15 *****
20 *****
30 *****
35 *****
37 *****
40 *****
49 *****
50 *****

```

This is all that is necessary, but it can be made to look a little nicer with a few extra statements in the report writer. This example only formats the bars; you can add your own statements to print maximums or averages, etc.

```
READY H_BAR
REPORT H_BAR ON HBAR.RPT
SET COLUMNS PAGE=80
SET REPORT_NAME="Horizontal Bar Graphs"
PRINT COL 10, VALUE, SPACE 1, "|", SPACE 1, BAR
END_REPORT
```

VALUE

```

1 | *
5 | *****
10 | *****
15 | *****
20 | *****
30 | *****
35 | *****
37 | *****
40 | *****
49 | *****
50 | *****
    
```

This is the minimum possible graph. You can easily make it nicer by printing a scale, or printing maximum or average values at the bottom, etc.

The above example was set to be as universal as possible. If you are graphing data in a known format, you can customize the procedure somewhat. For example, if you know the data will always be in the range of zero to fifty, you can fix the scale. In the procedure MAKE_H_BAR, change the value of BLANK from

```
BLANK = " " ! 50 blanks
```

to

```
BLANK = " | | | | |"
```

This will fill in an empty vertical grid rather than all blank spaces. Also, rather than prompting for the value of INC, you can set it equal to 1. As you know that there are 50 columns with a maximum scale of 50, so each column must have an increment of one. Otherwise, the procedure does not change. This domain would be printed with the following report:

```

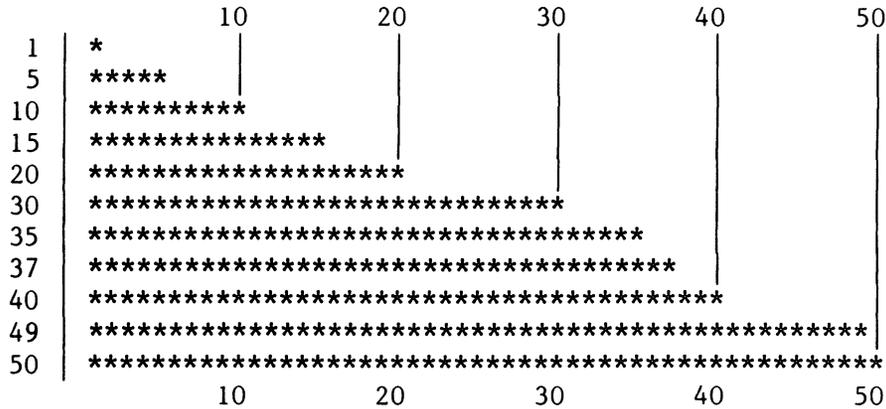
DELETE GRAPH H BAR;
DEFINE PROCEDURE GRAPH_H_BAR
READY H_BAR
REPORT H_BAR
SET COLUMNS PAGE=80
SET REPORT_NAME="Horizontal Bar Graphs"
AT TOP OF REPORT PRINT REPORT_HEADER, SKIP, COL 17,
" 10 20 30 40 50"
PRINT COL 10, VALUE, COL 15, "|", COL 17, BAR
AT BOTTOM OF REPORT PRINT COL 17,
" 10 20 30 40 50"
END_REPORT
END_PROCEDURE
    
```

When this is reported, it will look like:

Horizontal Bar Graphs

18-Jun-1985

Page 1



One may easily go on from here to make the graphs suit a particular application once the basic operation is understood. Don't be afraid to experiment on your own.

Users of DATATRIEVE-11 V3.0, V3.1, and Pro-DTR are cautioned about the use of DATATRIEVE keywords as field names. When preparing this article for publication, I fell into the trap of using a keyword as a field name. The original name for the list in H_STRING was "STRING". This worked fine in DATATRIEVE-11 V2.0 and V2.4, and works in all current versions of VAX-DATATRIEVE, but starting with DTR-11 V3.0, "STRING" is a keyword. I don't know what it is used for [Editor's note: STRING is used within the VAX-DATATRIEVE graphic language. See the next article.], but I had to change my field name to "STRNG" to avoid the conflict. This one had the telephone support center running around for a day or so before they found it.

Previews of Coming Attractions

Some of the articles which will appear in future issues of the Wombat Examiner are:

- o Part 2 of the Wombat Magic Session from the New Orleans Symposium. Part 1 appeared in Volume 6, Number 3.
- o An article by Don Stern on "Accessing SYSUAF.DAT and QUOTA.SYS with VAX DATATRIEVE" that will be of interest to all VAX system managers.
- o A transcription of the Fourth General Language Panel at New Orleans Symposium. Speakers were Katherine Hornback (Languages and Tools SIG Chair), Chuck Duncan (Digital), and Larry Jasmann (DATATRIEVE SIG Chair). This session contained some lively discussion.

VAX-DATATRIEVE Graphics Internals

Joe H. Gallagher, Cleveland Clinic Foundation, Cleveland, OH

Introduction

One of the most important characteristics of DATATRIEVE besides its productivity advantages over third generation languages, is its ability to be modified and extended to meet the needs of users. Kirk Searle of Digital made a presentation yesterday afternoon (Customizing VAX DATATRIEVE) on how VAX-DATATRIEVE could be customized to change its apparent functionality by modifying its internal tables, lists, and domains. Larry Jasmann and Phil Naecker this morning in an impromptu, fill in session (Using the DATATRIEVE Call Interface) provided information on how to add apparent functionality by using the call interface. With the call interface, it is possible to add what appears to be new statements in the DATATRIEVE language. And this afternoon, Andy Schneider of Digital will show how the features and functionality of DATATRIEVE may be used in a distributed environment (DATATRIEVE Distributed Manipulation Facility). In addition, users may add new functions to DATATRIEVE with additions to the MACRO linkage file, DTR.FND, to user written and VAX library routines that Phil Naecker described. Perhaps more examples of this type will be given at the Wombat Magic session on Thursday evening. In none of these situation has the internal code of DATATRIEVE been changed as would be required if a completely new statement or command been added to DATATRIEVE such as a statement like

FIND C IN A UNION B

That is a whole new idea and would certainly require internal code to be written.

There is an area of DATATRIEVE where users can write new code in a somewhat PASCAL-like language, which can add functionality which is fundamentally new to DATATRIEVE. I am, of course, referring to the internals of the DATATRIEVE plot library which is the topic of this presentation.

In the previous presentation (Using VAX DATATRIEVE Graphics), Kirk Searle described the use of the plot library from the outside. I wish to describe it from the inside and how one would go about adding to it and extending it. This talk is really in three parts. In order to be able to extend the library one has to understand its structure. The first part of the talk will be spent in looking at the structure of the plot library and how the pieces interrelate to each other. If one is going to make changes in a group of library routines, it is very important to understand how and where to make changes so the linkages between the routines won't get broken. The second part of the talk is an introduction to the plot language. In the third part, some new plots are presented.

Structure and Contents of the Plots Dictionary

Let's look in a slightly different way at those routines which Kirk Searle previous described. The plot routines are contained in the common data dictionary node, CDD\$TOP.DTR\$LIB.PLOTS . The entire contents of that node may be gotten with the EXTRACT ALL command in DATATRIEVE. The plot routines fall into five categories: documented top level plots which start a new plot; documented top level plots which change and/or continue a plot; documented top level plots which modify (without substantive change) a plot; undocumented top level plots; and utility level plots. All routines are listed below and the undocumented and utility routines are briefly described.

Documented Top Level Plots which Start a New Plot

BAR	BAR_AVERAGE	DATE_LOGY	DATE_Y
HISTO	LOGX_LOGY	LOGX_Y	MULTI_BAR
MULTI_BAR_GROUP	MULTI_LINE	MULTI_LR	MULTI_SHADE
PIE	RAW_BAR	RAW_PIE	STACKED_BAR
VALUE_PIE	WOMBAT	X_LOGY	X_Y

Documented Top Level Plots which Change and Continue a Plot

NEXT_BAR	SORT_BAR
----------	----------

Documented Top Level Plots which Modify (without substantive change) a Plot

BIG	CONNECT	CROSS_HATCH	HARDCOPY
LR	PAUSE	RE_PAINT	SHADE

Undocumented Top Level Plots

DATE_FREQ	frequency of occurrence vs. date; arg1=date.
HATCH	exactly same as CROSS_HATCH; no arguments.
OUTSTANDING	works a little like "PLOT DATE_FREQ . . .; PLOT SHADE"; arg1=date, arg2=date.
X_FREQ	frequency of occurrence vs. X; arg1=x-value.
STACKED	** This appears to be old code from some previous version of the plot routines. It appears, currently, to be unreachable code. There are no routines which call it. It also appears not to work - at all. **

Utility Level Plots

HOUSEKEEP	Plot for entering and exiting graphics mode on the terminal. Manages differences between VT240/1 and VT125.
LABEL	Utility plot routine with 14 entry points which draws boundary box, scales X and Y, linear, log, and date arrays, computes linear regressions, displays scale lines, and performs several other internal utility functions.
LEGEND	Utility plot routine with 4 entry points which manages LEGENDS.
MAP	Basic routine to manage intensity, luminance, hue, and saturation. Used by the PALETTE procedure to adjust colors.
MONITOR	Test routine to check wiring of color monitor on VT125 or VT241.

DRAW_BAR	Basic BAR plotting routine that does the real work for BAR, BAR_AVERAGE, HISTO, and RAW_BAR.
DRAW_PIE	Basic PIE plotting routine that does the real work for PIE, RAW_PIE, and VALUE_PIE.
X_LABEL	Basic X-axis labeling routine that works for MULTI_BAR, MULTI_BAR_GROUP, MULTI_SHADE, and STACKED_BAR

As will be described in detail in the second part of this presentation, plots within the plot library are all called in a manner very similar to multiple entry subroutines in FORTRAN. Most top level plots have three or more entry points. However, WOMBAT has only one, but LABEL has fifteen.

In order to investigate which plot routine calls which other plot routines (more specifically which entry point of which routine calls which entry point of another or the same routine), I extracted the entire plot library from the dictionary, removed the procedure PALETTE with the editor, and converted all "REDEFINE PLOT" to "REDEFINE_PLOT" also with the editor. I then used a small BASIC program which keyed off of the tokens "REDEFINE_PLOT", "ENTRY", and "PLOT" to create the pairs of routine entry points - calling entry point with called entry point. The entry points are indicated by appending an underline character and the number of the entry point to the name of the plot routine. Thus, HOUSEKEEPING_2 refers to the entry point 2 in the HOUSEKEEPING plot routine. I then use DATATRIEVE to create a view of four domains (actually the same domain with itself three more times) in which the called entry point is matched with another calling entry point to create the entire tree of references through the plot library. Multiple references to the same linkage (duplicates) were removed and the output report was slightly re-formatted with the editor.

The "CALLING" linkages within the plot library are listed in the table "DOWN the Calling Tree". In this table, one can find all routines in alphabetic order which call other routines. Below each routine is the routines which it calls. The order in which routines are called is not preserved; they are listed in alphabetic order.

Inverting the calling tree with the DATATRIEVE view produces the "CALLED" linkages within the plot library. These are listed in the table "UP the Called Tree". In this table, one can find all routines in alphabetic order which are called by other routines. Below each routine is the routines which it is called by. It is from this table that one can determine which routines might be affected by a change in a particular routine.

When support for VT240 and VT241 terminals was added in Version 3.0 of VAX-DATATRIVE, the method of performing I/O for graphics instructions was changed. With the VT125 it was possible to write directly to the terminal. Because the graphics planes and characters planes of the VT125 were independent; command prompting, command input, and graphics output were unaffected by each other. However, with the VT240/1 terminal in which the graphic and character displays were intertwined, the DATATRIEVE plot library had to be changed to write all graphics instructions into internal buffers so that the VT24X terminal could be cleared and the image re-displayed one or more times. These buffers are called SEGMENTS; there are eight of them numbered zero to seven. There are three ways of referencing these segments - CLEAR, OUTPUT, and SET. The meaning of these references will be explained in the second part of the presentation. The table "PLOTNAME\SEGMENTS" lists

DOWN the Calling Tree

BAR 2	DATE LOGY 2	HISTO 2	LOGX Y 2
DRAW BAR 0	HOUSEKEEP 2	DRAW BAR 0	HOUSEKEEP 2
DRAW BAR 2	LABEL 5	DRAW BAR 2	LABEL 3
HOUSEKEEP 0	LABEL 11	HOUSEKEEP 0	LABEL 10
HOUSEKEEP 2	LABEL 6	HOUSEKEEP 2	LABEL 11
LABEL 0	LABEL 14	LABEL 0	LABEL 4
LABEL 13	LABEL 8	LABEL 13	LABEL 12
LABEL 15	DATE Y 0	LABEL 15	LABEL 14
LABEL 3	HOUSEKEEP 0	LABEL 3	LABEL 8
LABEL 10	LABEL 0	LABEL 10	LR 0
LABEL 11	LABEL 13	LABEL 11	HOUSEKEEP 2
LABEL 8	DATE Y 2	LABEL 8	HOUSEKEEP 4
BAR AVERAGE 2	HOUSEKEEP 2	LABEL 0	LABEL 7
DRAW BAR 0	LABEL 3	LABEL 13	LABEL 13
DRAW BAR 2	LABEL 10	LABEL 2	MAP 2
HOUSEKEEP 0	LABEL 11	LABEL 10	HOUSEKEEP 2
HOUSEKEEP 2	LABEL 6	LABEL 12	MONITOR 0
LABEL 0	LABEL 14	LABEL 14	HOUSEKEEP 0
LABEL 13	LABEL 8	LABEL 3	HOUSEKEEP 2
LABEL 15	DRAW BAR 2	LABEL 10	MULTI BAR 0
LABEL 3	HOUSEKEEP 0	LABEL 11	HOUSEKEEP 0
LABEL 10	HOUSEKEEP 2	LABEL 4	LABEL 0
LABEL 11	LABEL 0	LABEL 12	LABEL 13
LABEL 8	LABEL 13	LABEL 14	MULTI BAR 2
BIG 0	LABEL 15	LABEL 5	HOUSEKEEP 2
HOUSEKEEP 4	LABEL 3	LABEL 11	LABEL 10
CONNECT 0	LABEL 10	LABEL 6	LABEL 3
HOUSEKEEP 2	LABEL 11	LABEL 14	LABEL 10
HOUSEKEEP 4	LABEL 8	LABEL 7	LABEL 11
LABEL 9	DRAW BAR 3	LABEL 13	LABEL 8
CROSS HATCH 0	DRAW BAR 2	LEGEND 0	LEGEND 3
HOUSEKEEP 2	HOUSEKEEP 0	HOUSEKEEP 2	LEGEND 5
DATE FREQ 0	HOUSEKEEP 2	LEGEND 3	X LABEL 0
HOUSEKEEP 0	LABEL 0	LEGEND 5	MULTI BAR GROUP 0
LABEL 0	LABEL 13	LEGEND 4	HOUSEKEEP 0
LABEL 13	LABEL 15	LEGEND 5	LABEL 0
DATE FREQ 2	LABEL 3	LOGX LOGY 0	LABEL 13
HOUSEKEEP 2	LABEL 10	HOUSEKEEP 0	MULTI BAR GROUP 2
LABEL 3	LABEL 11	LABEL 0	HOUSEKEEP 2
LABEL 10	LABEL 8	LABEL 13	LABEL 10
LABEL 11	DRAW PIE 0	LOGX LOGY 2	LABEL 3
LABEL 6	HOUSEKEEP 0	HOUSEKEEP 2	LABEL 10
LABEL 14	HOUSEKEEP 2	LABEL 4	LABEL 11
LABEL 8	HARDCOPY 0	LABEL 12	LABEL 8
DATE LOGY 0	HOUSEKEEP 2	LABEL 14	LEGEND 3
HOUSEKEEP 0	HOUSEKEEP 4	LABEL 5	LEGEND 5
LABEL 0	HATCH 0	LABEL 11	X LABEL 0
LABEL 13	HOUSEKEEP 2	LABEL 8	MULTI LINE 0
		LOGX Y 0	HOUSEKEEP 0
		HOUSEKEEP 0	LABEL 0
		LABEL 0	LABEL 13
		LABEL 13	

MULTI LINE 2	NEXT BAR 0	SORT BAR 0	X LOGY 0
HOUSEKEEP 2	DRAW BAR 2	DRAW BAR 3	HOUSEKEEP 0
LABEL 2	HOUSEKEEP 0	DRAW BAR 2	LABEL 0
LABEL 10	HOUSEKEEP 2	HOUSEKEEP 0	LABEL 13
LABEL 12	LABEL 0	HOUSEKEEP 2	X LOGY 2
LABEL 14	LABEL 13	LABEL 0	HOUSEKEEP 2
LABEL 3	LABEL 15	LABEL 15	LABEL 2
LABEL 10	LABEL 3	LABEL 3	LABEL 10
LABEL 11	LABEL 10	LABEL 8	LABEL 12
LEGEND 4	LABEL 11	STACKED 2	LABEL 14
LEGEND 5	LABEL 8	HOUSEKEEP 0	LABEL 5
MULTI LINE 3	OUTSTANDING 0	HOUSEKEEP 2	LABEL 11
LABEL 8	HOUSEKEEP 0	LABEL 0	LABEL 8
MULTI LINE 3	LABEL 0	LABEL 13	X Y 0
LABEL 8	LABEL 13	LABEL 3	HOUSEKEEP 0
MULTI LR 0	OUTSTANDING 2	LABEL 10	LABEL 0
HOUSEKEEP 0	HOUSEKEEP 2	LABEL 11	LABEL 13
LABEL 0	LABEL 3	LEGEND 0	X Y 2
LABEL 13	LABEL 10	HOUSEKEEP 2	HOUSEKEEP 2
MULTI LR 2	LABEL 11	STACKED 3	LABEL 2
HOUSEKEEP 2	LABEL 6	STACKED BAR 0	LABEL 10
LABEL 2	LABEL 14	HOUSEKEEP 0	LABEL 12
LABEL 10	LABEL 8	LABEL 0	LABEL 14
LABEL 12	PIE 2	LABEL 13	LABEL 3
LABEL 14	DRAW PIE 0	STACKED BAR 2	LABEL 10
LABEL 3	HOUSEKEEP 0	HOUSEKEEP 2	LABEL 11
LABEL 10	HOUSEKEEP 2	LABEL 3	LABEL 8
LABEL 11	RAW BAR 2	LABEL 10	
LEGEND 4	DRAW BAR 0	LABEL 11	
LEGEND 5	DRAW BAR 2	LABEL 8	
MULTI LR 3	HOUSEKEEP 0	LEGEND 3	
LABEL 7	HOUSEKEEP 2	LEGEND 5	
LABEL 13	LABEL 0	X LABEL 0	
LABEL 8	LABEL 13	VALUE PIE 2	
MULTI LR 3	LABEL 15	DRAW PIE 0	
LABEL 7	LABEL 3	HOUSEKEEP 0	
LABEL 13	LABEL 10	HOUSEKEEP 2	
LABEL 8	LABEL 11	WOMBAT 0	
MULTI SHADE 0	LABEL 8	HOUSEKEEP 0	
HOUSEKEEP 0	RAW PIE 2	HOUSEKEEP 2	
LABEL 0	DRAW PIE 0	X FREQ 0	
LABEL 13	HOUSEKEEP 0	HOUSEKEEP 0	
MULTI SHADE 2	HOUSEKEEP 2	LABEL 0	
HOUSEKEEP 2	RE PAINT 0	LABEL 13	
LABEL 3	HOUSEKEEP 4	X FREQ 2	
LABEL 10	RE PAINT 2	HOUSEKEEP 2	
LABEL 11	HOUSEKEEP 2	LABEL 2	
LABEL 8	SHADE 0	LABEL 10	
LEGEND 3	HOUSEKEEP 2	LABEL 12	
LEGEND 5	HOUSEKEEP 4	LABEL 14	
X LABEL 0	LABEL 9	LABEL 3	
		LABEL 10	
		LABEL 11	
		LABEL 8	

UP the Called Tree

DRAW BAR 0
 BAR 2
 BAR AVERAGE 2
 HISTO 2
 RAW BAR 2
 HOUSEKEEP 0
 DATE_FREQ 0
 DATE_LOGY 0
 DATE_Y 0
 DRAW BAR 2
 BAR 2
 BAR AVERAGE 2
 DRAW BAR 3
 SORT BAR 0
 HISTO 2
 NEXT BAR 0
 RAW BAR 2
 DRAW PIE 0
 PIE 2
 RAW PIE 2
 VALUE PIE 2
 LOGX_LOGY 0
 LOGX_Y 0
 MONITOR 0
 MULTI BAR 0
 MULTI BAR GROUP 0
 MULTI LINE 0
 MULTI LR 0
 MULTI SHADE 0
 OUTSTANDING 0
 STACKED 2
 STACKED BAR 0
 WOMBAT 0
 X_FREQ 0
 X_LOGY 0
 X_Y 0

HOUSEKEEP 2
 CONNECT 0
 CROSS HATCH 0
 DATE_FREQ 2
 DATE_LOGY 2
 DATE_Y 2
 DRAW BAR 2
 BAR 2
 BAR AVERAGE 2
 DRAW BAR 3
 SORT BAR 0
 HISTO 2
 NEXT BAR 0
 RAW BAR 2
 DRAW PIE 0
 PIE 2
 RAW PIE 2
 VALUE PIE 2
 HARDCOPY 0
 HATCH 0
 LEGEND 0
 STACKED 2
 LOGX_LOGY 2
 LOGX_Y 2
 LR 0
 MAP 2
 MONITOR 0
 MULTI BAR 2
 MULTI BAR GROUP 2
 MULTI LINE 2
 MULTI LR 2
 MULTI SHADE 2
 OUTSTANDING 2
 RE_PAINT 2
 SHADE 0
 STACKED 2
 STACKED BAR 2
 WOMBAT 0
 X_FREQ 2
 X_LOGY 2
 X_Y 2
 HOUSEKEEP 4
 BIG 0
 CONNECT 0
 HARDCOPY 0
 LR 0
 RE_PAINT 0
 SHADE 0
 LABEL 0
 DRAW BAR 2
 DRAW BAR 3
 SORT BAR 0

LABEL 10
 LABEL 2
 MULTI LINE 2
 MULTI LR 2
 X_FREQ 2
 X_LOGY 2
 X_Y 2
 LABEL 3
 DATE_FREQ 2
 DATE_Y 2
 DRAW BAR 2
 BAR 2
 BAR AVERAGE 2
 DRAW BAR 3
 HISTO 2
 NEXT BAR 0
 RAW BAR 2
 LOGX_Y 2
 MULTI BAR 2
 MULTI BAR GROUP 2
 MULTI LINE 2
 MULTI LR 2
 MULTI SHADE 2
 OUTSTANDING 2
 STACKED 2
 STACKED BAR 2
 X_FREQ 2
 X_Y 2
 MULTI BAR 2
 MULTI BAR GROUP 2

LABEL 11
 LABEL 3
 DATE_FREQ 2
 DATE_Y 2
 DRAW BAR 2
 BAR 2
 BAR AVERAGE 2
 DRAW BAR 3
 HISTO 2
 NEXT BAR 0
 RAW BAR 2
 LOGX_Y 2
 MULTI BAR 2
 MULTI BAR GROUP 2
 MULTI LINE 2
 MULTI LR 2
 MULTI SHADE 2
 OUTSTANDING 2
 STACKED 2
 STACKED BAR 2
 X_FREQ 2
 X_Y 2
 LABEL 5
 DATE_LOGY 2
 LOGX_LOGY 2
 X_LOGY 2
 LABEL 12
 LABEL 2
 MULTI LINE 2
 MULTI LR 2
 X_FREQ 2
 X_LOGY 2
 X_Y 2
 LABEL 4
 LOGX_LOGY 2
 LOGX_Y 2

LABEL 13	LABEL 13 (continued)	LABEL 15 (continued)	LABEL 8 (continued)
LABEL 0	LABEL 7	HISTO 2	MULTI LR 3
DATE_FREQ 0	LR 0	NEXT BAR 0	MULTI LR 2
DATE LOGY 0	MULTI LR 3	RAW BAR 2	MULTI SHADE 2
DATE Y 0	MULTI LR 2	LABEL 3	OUTSTANDING 2
DRAW BAR 2	LABEL 14	DRAW BAR 2	STACKED BAR 2
BAR 2	LABEL 2	DRAW BAR 3	X_FREQ 2
BAR AVERAGE 2	MULTI LINE 2	SORT BAR 0	X LOGY 2
DRAW BAR 3	MULTI LR 2	LABEL 8	X Y 2
HISTO 2	X_FREQ 2	DATE_FREQ 2	LABEL 9
NEXT BAR 0	X LOGY 2	DATE LOGY 2	CONNECT 0
RAW BAR 2	X Y 2	DATE Y 2	SHADE 0
LOGX LOGY 0	LABEL 4	DRAW BAR 2	LEGEND 5
LOGX Y 0	LOGX LOGY 2	BAR 2	LEGEND 3
MULTI BAR 0	LOGX Y 2	BAR AVERAGE 2	MULTI BAR 2
MULTI BAR GROUP 0	LABEL 6	DRAW BAR 3	MULTI BAR GROUP 2
MULTI LINE 0	DATE_FREQ 2	SORT BAR 0	MULTI SHADE 2
MULTI LR 0	DATE LOGY 2	HISTO 2	STACKED BAR 2
MULTI SHADE 0	DATE Y 2	NEXT BAR 0	LEGEND 4
OUTSTANDING 0	OUTSTANDING 2	RAW BAR 2	MULTI LINE 2
STACKED 2	LABEL 15	LOGX LOGY 2	MULTI LR 2
STACKED BAR 0	DRAW BAR 2	LOGX Y 2	STACKED 3
X_FREQ 0	BAR 2	MULTI BAR 2	STACKED 2
X LOGY 0	BAR AVERAGE 2	MULTI BAR GROUP 2	X LABEL 0
X Y 0	DRAW BAR 3	MULTI LINE 3	MULTI BAR 2
	SORT BAR 0	MULTI LINE 2	MULTI BAR GROUP 2
			MULTI SHADE 2
			STACKED BAR 2

all plot library entry points which make a reference to a segment. With each is listed the segment number which is referenced and the method by which it is referenced. Most top level plot routines make reference to one or more segments within their entry points 0 and 2. Notice that HOUSEKEEP_0 clears all segments and all top level routines call HOUSEKEEP_0.

The table "Segments Access by Plots" inverts the "PLOTNAME\SEGMENTS" table to show which segments are manipulated by which plots. It is from the information in these two tables that one can see how changes in such plot routines as HOUKEEPING, HARDCOPY, and BIG will affect other routines.

\SEGMENTS PLOTNAME\	CLEAR							OUTPUT							SET									
	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
BIG_0					4								4	5	6								4	
CONNECT_0																							1	
CROSS_HATCH_0	0				4				0	1		3	4		6		0						4	
DATE_FREQ_0																							1	
DATE_FREQ_2										1													1	
DATE_LOGY_0																							1	
DATE_LOGY_2																							1	
DATE_Y_0																							1	
DATE_Y_2																							1	
DRAW_BAR_2																							1	
DRAW_PIE_0					3				0	1							0	1					5	
HARDCOPY_0						4								4	5	6							4	
HATCH_0	0				4				0	1		3	4		6		0						4	
HOUSEKEEP_0	0	1	2	3	4	5	6	7						4	6								4	6
HOUSEKEEP_1					4									4									4	
HOUSEKEEP_2								7								7								7
HOUSEKEEP_3					4									4									4	
HOUSEKEEP_4					4				0	1		3	4		6								4	
LEGEND_0					4									4	6								4	
LEGEND_5											2	3									2	3		
LOGX_LOGY_0																						1		
LOGX_LOGY_2										1												1		
LOGX_Y_0																						1		
LOGX_Y_2										1												1		
LR_0																						1		
MAP_0					4																		4	
MAP_2														4										
MONITOR_0		1								1												1		
MULTI_BAR_0									0						6		0	1						
MULTI_BAR_2										1														
MULTI_BAR_GROUP_0									0						6		0	1						
MULTI_BAR_GROUP_2										1														
MULTI_LINE_0									0								0	1						
MULTI_LINE_2										1														
MULTI_LR_0									0								0	1						
MULTI_LR_2										1														
MULTI_SHADE_0									0						6		0	1						
MULTI_SHADE_2										1														
OUTSTANDING_0																						1		
OUTSTANDING_2										1														
PAUSE_0					4									4									4	
SHADE_0																						1		
STACKED_2										1												1		
STACKED_BAR_0									0						6		0	1						
STACKED_BAR_2										1														
WOMBAT_0									0	1							0	1	2					
X_FREQ_0																						1		
X_FREQ_2										1														
X_LOGY_0																						1		
X_LOGY_2										1														
X_Y_0																						1		
X_Y_2										1														

Segments Access by Plots

Segment 0

CROSS_HATCH_0
 DRAW_PIE_0
 HATCH_0
 HOUSEKEEP_0
 HOUSEKEEP_4
 MULTI_BAR_0
 MULTI_BAR_GROUP_0
 MULTI_LINE_0
 MULTI_LR_0
 MULTI_SHADE_0
 STACKED_BAR_0
 WOMBAT_0

Segment 1

CONNECT_0
 CROSS_HATCH_0
 DATE_FREQ_0
 DATE_FREQ_2
 DATE_LOGY_0
 DATE_LOGY_2
 DATE_Y_0
 DATE_Y_2
 DRAW_BAR_2
 DRAW_PIE_0
 HATCH_0
 HOUSEKEEP_0
 HOUSEKEEP_4
 LOGX_LOGY_0
 LOGX_LOGY_2
 LOGX_Y_0
 LOGX_Y_2
 LR_0
 MONITOR_0
 MULTI_BAR_0
 MULTI_BAR_2
 MULTI_BAR_GROUP_0
 MULTI_BAR_GROUP_2
 MULTI_LINE_0
 MULTI_LINE_2
 MULTI_LR_0
 MULTI_LR_2
 MULTI_SHADE_0
 MULTI_SHADE_2
 OUTSTANDING_0
 OUTSTANDING_2
 SHADE_0
 STACKED_2
 STACKED_BAR_0
 STACKED_BAR_2
 WOMBAT_0
 X_FREQ_0
 X_FREQ_2

Segment 1 (continued)

X_LOGY_0
 X_LOGY_2
 X_Y_0
 X_Y_2

Segment 2

HOUSEKEEP_0
 LEGEND_5
 WOMBAT_0

Segment 3

CROSS_HATCH_0
 DRAW_PIE_0
 HATCH_0
 HOUSEKEEP_0
 HOUSEKEEP_4
 LEGEND_5

Segment 4

BIG_0
 CROSS_HATCH_0
 HARDCOPY_0
 HATCH_0
 HOUSEKEEP_0
 HOUSEKEEP_1
 HOUSEKEEP_3
 HOUSEKEEP_4
 LEGEND_0
 MAP_0
 MAP_2
 PAUSE_0

Segment 5

BIG_0
 DRAW_PIE_0
 HARDCOPY_0
 HOUSEKEEP_0

Segment 6

BIG_0
 CROSS_HATCH_0
 HARDCOPY_0
 HATCH_0
 HOUSEKEEP_0
 HOUSEKEEP_4
 LEGEND_0
 MULTI_BAR_0
 MULTI_BAR_GROUP_0
 MULTI_SHADE_0
 STACKED_BAR_0

Segment 7

HOUSEKEEP_0
 HOUSEKEEP_2

Plot Language

Plot Definition:

A plot definition has the following structure:

```
DEFINE PLOT plotname
DECLARE TYPE-SPECIFICATION date_variable [,data_variable...]
. . .
ENTRY 0 [{P1[:TYPE[:COMMENT]] [,Pn[:TYPE[:COMMENT]]}]
  BEGIN
    . . .
    plot statements
    . . .
  END
. . .
ENTRY N [{P1[:TYPE[:COMMENT]] . . . ]
  BEGIN
    . . .
  END
END_PLOT
```

The plot begins with a "DEFINE PLOT" and ends with an "END-PLOT".

Declarations Statement and Data Typing:

There are three data types allow with the plot package. These are REAL, DATE, and STRING. The data type REAL is the default data type. Thus, if a variable is to be of the DATA or STRING data type, it must be declared in a DECLARE statement or have its type specified within the entry point statement. In addition the plot package allows for dynamic arrays. That is, array variables may be allocated on the fly as needed. To specify that a variable is an array, then the declaration modifier VECTOR is added to the data declaration statement. Vectors can be indexed by literal integers (even though there is not really an integer data type) or by REALs. The reals are rounded to the nearest integer before being used as an index. Thus for an N element vector, the valid indexes I are $0.5 \leq I < N+0.5$.

Entry Points:

Unlike any other DATATRIEVE objects, plots have multiple entry points. These entry points have parameters which are specified in a manner very similar to PASCAL that allows a flexible but more carefully specified interface with ordinary DATATRIEVE statements. The form of an entry statement is

```
ENTRY N [{P1[:TYPE[:COMMENT-STRING]] [,Pn[:TYPE[:COMMENT-STRING]]}]
```

where N is an integer literal, P1 though Pn are variable names, TYPE is one of the valid data types (REAL, DATE, and STRING), and COMMENT-STRING is a literal string while documents variable being passed. Parameters are, in effect, declared in the entry point with the TYPE modifier. The default data type is REAL, so if the TYPE modifier is left off, the parameter is assumed to be REAL.

Entry point 0 is used to initialize the plot. It is called once to initialize the plot and to pass in the labels associated with all of the parameters. This entry point specifies the number of parameters of the plot, and all of the parameters are STRING type. In general, this entry point clears the necessary plot segment buffers, clears the screen, and sets up size and location of the plots and any special plotting parameters like shading or cross-hatching.

Entry point 1 is called once for each set of parameters that are to be managed in the plot. Thus if there are 23 points to be plot, entry point 1 would be used 23 times. In general, data is moved from the user's record stream into internal arrays. Usually there is no actual output within this entry point. Data is just gathered up.

Entry point 2 is called only, once and it has no parameters. Final calculations are done, scaling of the plot is passed to various routines within LABEL, the actual plot is written into any appropriate PLOT SEGMENT buffer, and the control is passed off to HOUSEKEEPING to display the plot.

Most top level plots have only three entry points 0, 1, and 2. That is, plots which are called with the DATATRIEVE PLOT statement observe the conventions of entry points 0, 1, and 2. A single PLOT statement at DATATRIEVE command level will execute entry 0 once, entry 1 multiple time, and entry 2 one.

Top level plots which are called by the user may make calls to other utility plots. These calls from within plot definitions to other plots are of the form

```
PLOT UTILITY_PLOTNAME ENTRY_NUMBER [ ... argument list ... ]
```

Utility plots, in contract to top level plots, do not observe the convention of entry points 0, 1, and 2. However, many utility plots do not have an entry 1 or 2. When utility plots are called with a specify entry point reference, only the specified entry is executed. No other plot routines are executed unless specifically called.

Statements:

The statements with the PLOT Package appear similar to the statements with DATATRIEVE. However, they are, in some ways, very different. Statements in the PLOT Package are a highly structure language just as is DATATRIEVE. Thus, there is no GOTO-like statement or need for one. There are no statements in the PLOT language other than those described below and the "DEFINE PLOT", "END-PLOT", "DECLARE", and "ENTRY" statements.

Assignment Statement:

The assignment statement takes the usual form. However, it is possible to have multiple assignment statements on a single line as in the following example:

```
X = 23.5  
Y = 27.8    PSTRING = 'NUMBER'
```

Compound Statement:

Compound statements are exactly the same as in DATATRIVE. The BEGIN-END form the boundaries of the compound statement.

```
BEGIN
  X = X + 1
  Y = X / 3
END
```

IF-THEN-ELSE Statement:

Although the IF-THEN-ELSE statement within the PLOT Package may appear to be the same as in DATATRIVE, it is not. The difference is that an ELSE may begin a statement line. Thus it is possible to easily create n-way branching logic without the use of BEGIN-END statements as illustrated in the following examples:

```
IF X < 5 THEN
BEGIN
  X = X + 1
  Y = X * X
END
ELSE Y = X * 3
```

or

```
IF X = 1 THEN
  PRINT "X equals 1"
ELSE IF X = 2 THEN
  PRINT "X equals 2"
ELSE IF X = 3 THEN
  PRINT "X equals 3"
ELSE
  PRINT "X equals other"
```

PRINT Statement:

Since the support of VT240 and VT241 terminals, ReGIS code is not transferred directly from the modules of the PLOT Package to the terminal hardware, but the ReGIS instructions are first placed in output buffers (known as SEGMENTS) since it may be required to re-execute the graphics instructions. Thus, ReGIS code is saved in internal segments (buffers). The PRINT statement moves data into these buffers. There are four special "functions" which control the use of these buffers. They are:

RELEASE_SEGMENT	- initialize all segments
CLEAR_SEGMENT N	- initialize segment N
SET_SEGMENT N	- cause all output from the PRINT statements to go into segment N
OUTPUT_SEGMENT N[,N2[,N3 . . .]]	- transfer the contents of segments N, N2, N3, etc to the display terminal. That is, actually perform the output.

Thus, the PRINT statement does not do output directly, but moves data into buffers. There are several special symbols or declared global variables which have meaning only within the PLOT Package. These are:

\$	- special symbol for <ESC>
CLRSCR	- clear screen escape sequence
ENTER_REGIS(1)	- enter ReGIS mode escape sequence
EXIT_REGIS	- exit ReGIS mode escape sequence
RESTORE_VT125	- special ReGIS code sequence which sets the four (black and white) intensity levels to 0, 33, 66, and 100%.

A PRINT statement may contains one or more of the special symbols, one or more special graphics functions (described later), or strings. Some examples of PRINT statement are:

```
PRINT $,`<`,CLRSCR,ENTER_REGIS(1),`S(m0(10)1(133)2(166)3(1100))`
PRINT `P`,LXY(X_AXIS+1,Y_POS),`V(W(P4I(0)))`,RX(LINE_LEN)
```

PLOT Statement:

Other utility plot may be called from within a plot definition. To call these utility plots, one must specify the entry point as well as the calling arguments. Example of these utility plot calls are:

```
PLOT LABEL 2 (X_MIN, Y_MIN, X_VECTOR)
PLOT HOUSEKEEP 0
```

SORT Statement:

Although the SORT statement looks like a functions, it is a statement which performs a sorting operations on the specified array or arrays. Upon return from this statement, the arrays have been re-ordered in their same storage locations. The SORT statement may optionally take the modified DESCENDING or DESC. The SORT statement will sort on the first array but will re-order all arrays specified. SORT takes one to four arrays as arguments. Examples are:

```
DECLARE VECTOR X, Y1, Y2, Y3
DECLARE VECTOR DATE

SORT (X, Y1, Y2, Y3)
SORT DESCENDING (DATE, Y1)
```

INCREMENT Statement:

Since the PLOT Package contains arrays (which DATATRIEVE doesn't have), a DO-LOOP mechanism (which DATATRIEVE doesn't really need) is implemented. The INCREMENT statement takes on two forms. The first form is

INCR loop-variable FROM start-variable TO stop-variable

and

INCR loop-variable OVER array-name

This second form is equivalent to

INCR loop-variable FROM 1 TO SIZE(array-name)

Examples of the INCREMENT statement are:

```
INCR INDEX OVER VALUES
VALUES(INDEX) = VALUES(INDEX)/COUNT(INDEX)
```

and

```
INCR I FROM Y_ORIGIN TO TOP - 1
INCR J FROM 1 TO 9
  BEGIN
    POINT = J * 10 ** I
    . . .
  END
```

WHILE Statement:

A WHILE statement is also implemented although it is used only once within the PLOT Package in LABEL 6. The syntax of the WHILE statement is the same as in DATATRIEVE. An example is:

```
WHILE DATE_COM(SCRATCH_VALUE) < MAX_VALUE
  BEGIN
    . . .
  END
```

Operators and Expressions:

The usual set of arithmetic operators are implemented (+, -, *, and /). In addition, exponentiation is implemented in the form "**". There is, however unlike DATATRIEVE, no string concatenation operators within the PLOT Package.

Many Boolean relations operators have been implemented. However, they are, by no means, a complete and exhaustive set. The operators (AND, OR, <, >, EQ, LT, LE, GE, GT, and NE) exist. Notable by its absence is "NOT", but that doesn't seem to create any problems.

Functions:

The PLOT Package is rich in internally defined functions. User defined functions and the DATATRIEVE functions like FN\$FUNCTION are not accessible. The functions internal to the PLOT Package are:

CENTER - CENTER takes four arguments. A REAL value giving an X coordinate. A REAL value giving a Y coordinate. A STRING whose upper left corner is to be placed at the coordinate. And the size of the characters to be displayed (in pixels). CENTER returns a string representing the ReGIS commands to display the string.

CVT - CVT takes a REAL number as a parameter and returns it as a string.

COS - COS takes a REAL number in degrees and returns the REAL cosine.

INT - INT takes a REAL number and return a REAL number which has the value of the nearest integer.

LENGTH - LENGTH takes a STRING and return a REAL number which is the length of the string.

LOG - LOG takes a REAL number and returns the REAL natural log of the number.

LX - LX takes one REAL X coordinate and returns the string representing the absolute ReGIS coordinates [X].

LXY - LXY takes the REAL X and Y coordinates and returns the string representing the absolute ReGIS coordinates [X,Y].

LY - LY takes one REAL Y coordinate and returns the string representing the absolute ReGIS coordinate [,Y].

MAX - MAX takes an array as a parameter and returns the maximum value of its elements.

MIN - MIN takes an array as a parameter and returns the minimum value of its elements.

QUOTE - QUOTE takes a STRING as a parameter and returns the string in a form for inserting it into a ReGIS command.

RX - RX takes one REAL X coordinate and returns the string representing the relative ReGIS coordinates [+X].

RXY - RXY takes two REAL X and Y coordinate and returns the string representing the relative ReGIS coordinates [+X,+Y].

RY - RY takes one REAL Y coordinate and returns the string representing the relative ReGIS coordinate [, +Y].

SEARCH - SEARCH takes a string, and an array of strings as parameters and returns the index of the string in the array, or 0 if it is not found in the array.

SIN - SIN takes a REAL number in degrees and returns the REAL sine.

- SIZE - SIZE takes an array as a parameter and returns the number of elements in the array.
- SQRT - SQRT takes a REAL number as a parameters and returns the REAL square root. Even though this functions is available, it is not now used. But it still works! Wonder what other functions are available but are just not used?
- TXY - TXY takes two parameters which are the X (column) and Y (row) coordinates. TXY returns the escape sequence which will position the cursor to that location on the 80 column by 24 line screen.

Some New Plots

When Version 3.0 of DATATRIEVE was release last fall, the plot HARDCOPY was changed. In versions 2.X, HARDCOPY only printed the plot and the legend was not plotted unless specifically requested. With Version 3.0, HARDCOPY was changed so that the LEGEND was always copied -- even if the LEGEND was blank. This caused a lot of blank space after PIE, X_Y, and other kinds of plots. In order the maintain some compatibility with some of my command files, it was necessary to create a plot which worked like the Version 2.X HARDCOPY. I called the plot HARDCOPY_WITHOUT_LEGEND. The changes from the Version 3.0 of HARDCOPY are shown in bold and marked at the right-hand margin with "<--".

<pre> PLOT HARDCOPY ENTRY 0 BEGIN PLOT HOUSEKEEP 4 OUTPUT_SEGMENT 5 CLEAR_SEGMENT 4 SET_SEGMENT 4 PRINT 'S(H)S(E)P[100,200]@BS(H)' OUTPUT_SEGMENT 5,4,6 PLOT HOUSEKEEP 2 END END_PLOT </pre>	<pre> PLOT HARDCOPY_WITHOUT_LEGEND <-- ENTRY 0 BEGIN PLOT HOUSEKEEP 4 OUTPUT_SEGMENT 5 CLEAR_SEGMENT 4 SET_SEGMENT 4 PRINT 'S(H)' <-- OUTPUT_SEGMENT 5,4,6 PLOT HOUSEKEEP 2 END END_PLOT </pre>
--	--

In my system, I keep track of the number of disk blocks which are used by each account on each day. Very often I need to have a graph of the total disk blocks used by a group of users in a department or division. I would, therefore, like to have a plot of date vs the sum of y. The plots X_FREQ and DATE_FREQ can easily be modified to make plots X_SUM and DATE_SUM. In a similar way, BAR_AVERAGE could be changed make a plot BAR_SUM. DATE_SUM can be created from DATE_FREQ by making changes in only five lines of plot code. The changes are noted in bold and marked on the right-hand margin with "<--". The use of DATE_SUM is illustrated in Figures 1, 2, and 3.

```

PLOT DATE_FREQ                                PLOT DATE_SUM                                <--
.
.
ENTRY 0 (X_LABEL : STRING)                    ENTRY 0 (X_LABEL:STRING,Y_DL:STRING) <-
BEGIN
. . .
Y_LABEL = 'NUMBER'                            Y_LABEL = 'SUM'                                <--
. . .
END
ENTRY 1 (X : DATE)                            ENTRY 1 (X : DATE , Y)                        <--
BEGIN
INDEX = SEARCH (X, DATES)                    INDEX = SEARCH (X, DATES)
IF INDEX EQ 0                                IF INDEX EQ 0
BEGIN
INDEX = SIZE (DATES) + 1                    INDEX = SIZE (DATES) + 1
DATES (INDEX) = X                            DATES (INDEX) = X
END
FREQS(INDEX)=FREQS(INDEX) + 1                FREQS(INDEX)=FREQS(INDEX) + Y <--
END
ENTRY 2                                        ENTRY 2
BEGIN
. . .
END
END_PLOT                                    END_PLOT

```

In our medical environment it is very often necessary to display an X_Y scattergraph of more than one group of patients. By comparison with the plots X_Y and MULTI_LINE, a new plot X_Y_GROUP is created which has as its call arguments the x coordinate, the y coordinate, and a string indicating group membership. Because of the restrictions of LEGEND entry points 4 and 5, the number of groups is limited to three. All points belonging to the fourth, fifth, and higher groups are plotted as members of the third group.

```

DELETE X_Y_GROUP;
REDEFINE_PLOT X_Y_GROUP
DECLARE X_AXIS, Y_AXIS, X_LENGTH, Y_LENGTH, X_MAX, X_MIN, Y_MIN, Y_MAX
DECLARE X_POS, Y_POS, I, J, WIDTH
DECLARE VECTOR XS, YS, ZS, Y_MX,COLOR
DECLARE STRING VECTOR GROUP_VALUE, CHR
ENTRY 0 (X_LABEL : STRING, Y_LABEL : STRING, GROUP_LABEL : STRING)
BEGIN
PLOT HOUSEKEEP 0
SET_SEGMENT 0
PRINT 'L(A2)'
PRINT 'L"0"00181818FF181818'           ! cross
PRINT 'L"1"003C42818181423C'         ! circle
PRINT 'L"2"0081422418244281'         ! X
OUTPUT_SEGMENT 0
CHR(1) = '0' CHR(2) = '1' CHR(3) = '2'
COLOR(1) = 1 COLOR(2) = 1 COLOR(3) = 1
SET_SEGMENT 1
X_AXIS = 100
Y_AXIS = 360

```

```

X_LENGTH = 600
Y_LENGTH = 350
PLOT LABEL 0 (X_AXIS, Y_AXIS, X_LENGTH, Y_LENGTH, X_LABEL, Y_LABEL)
END
ENTRY 1 (X : REAL , Y : REAL , G : STRING)
BEGIN
  XS (SIZE (XS) + 1 ) = X
  YS (SIZE (YS) + 1 ) = Y
  I = SEARCH(G, GROUP_VALUE)
  IF I EQ 0 THEN BEGIN
    GROUP_VALUE(SIZE(GROUP_VALUE) + 1) = G
    I = SIZE(GROUP_VALUE)
    IF I GT 3 THEN I = 3          ! RESTRICTION DUE TO LEGEND
  END
  ZS (SIZE(ZS) + 1) = I
END
! Print scatter plot
ENTRY 2
BEGIN
  X_MIN = MIN (XS)
  X_MAX = MAX (XS)
  Y_MAX = MAX (YS)
  Y_MIN = MIN (YS)
  IF Y_MIN > 0
    THEN Y_MIN = 0
  PLOT LABEL 2 (X_MIN, X_MAX, XS)
  PLOT LABEL 3 (Y_MIN, Y_MAX)
  PLOT LABEL 8 (YS)
!
  PRINT 'T(BA2S[8,16])'
  INCR I OVER XS
    PRINT 'P', LX(XS(I)-4,YS(I)-8),'T',QUOTE(CHR(ZS(I)))
  PRINT 'T(E)'
  OUTPUT_SEGMENT 1
!
  WIDTH = X_LENGTH/30
  Y_MX(30) = 0
  INCR I OVER XS
  BEGIN
    J = ((XS(I) - X_AXIS) / WIDTH) + 1
    Y_MX(J) = 1000
    IF Y_MX(J) GT YS(I) THEN
      Y_MX(J) = YS(I)
    END
  INCR I OVER Y_MX
    IF (I NE 1) AND (Y_MX(I) EQ 0) THEN
      Y_MX(I) = Y_MX(I - 1)
  PLOT LEGEND 4 (X_AXIS, Y_AXIS, X_LENGTH, Y_LENGTH, WIDTH, Y_MX,
    CHR, COLOR, GROUP_VALUE)
  PLOT HOUSEKEEP 2
  END
END_PLOT

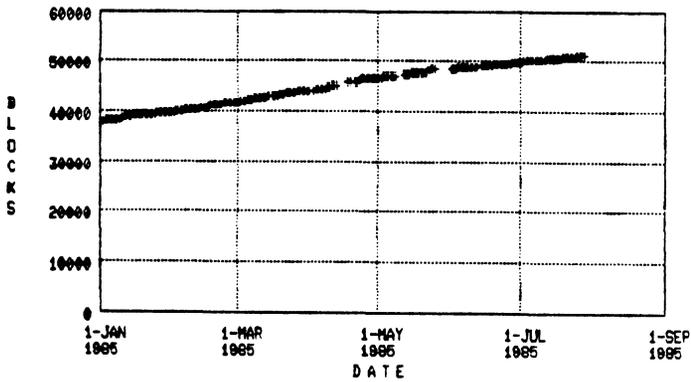
```

What I have given is three new plots; what I would really like to give you is the courage to go in and create your own plots. Keep in mind that the plot library is undocumented. That means, Digital makes no promises as to how DATATRIEVE graphics will be implemented in the future. If some new graphical device appears on the scene, it's likely that the internal structure of the plot library will change as it did in the case of VT240 support. If you create new plots, you are really on your own. But isn't the freedom great?

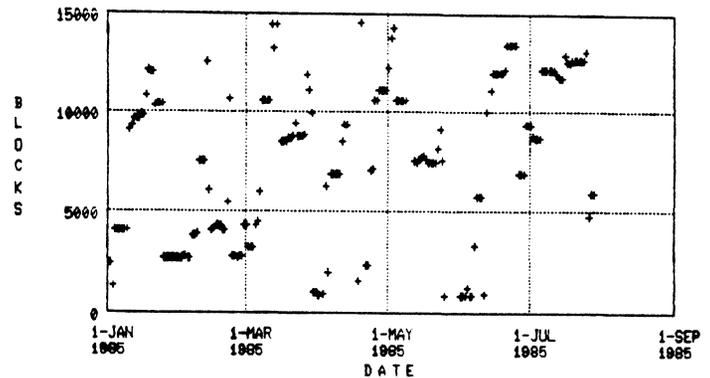
The news plot DATE_SUM is show in Figure 3. For comparison, Figures 1 and 2 shows the data which combines in Figure 3. Figure 4 is the new plot X_Y_GROUP. The plot HARDCOPY_WITHOUT_LEGEND can not be easily illustrated since it is the absence of a blank legend which is of interest.

Example Plots

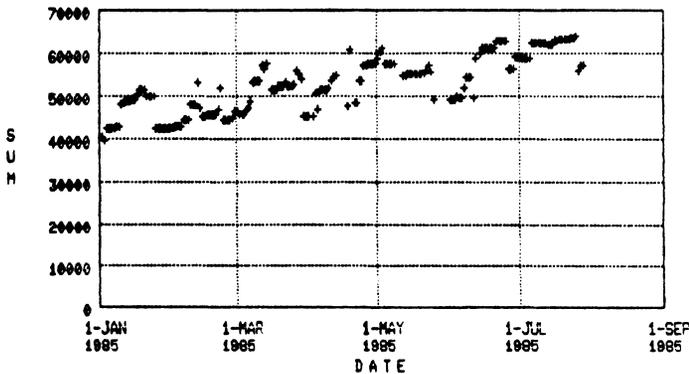
find diskuse with acc="USER1"
plot date_y all date, blocks
Figure 1.



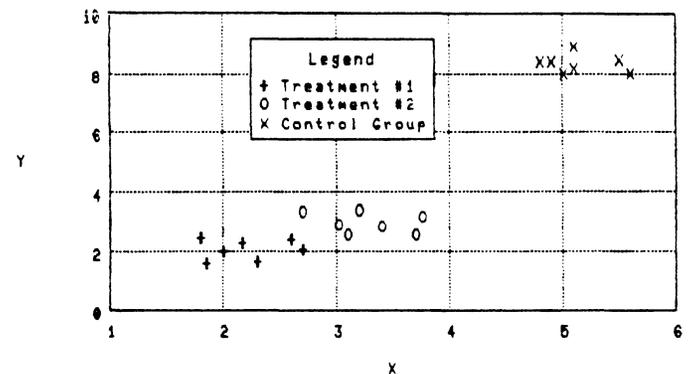
find diskuse with acc="USER2"
plot date_y all date, blocks
Figure 2.



find diskuse with acc="USER1","USER2"
plot date_sum all date, blocks
Figure 3.



plot x_y_group all x, y, group
Figure 4.



Conclusion

The VAX-DATATRIEVE PLOT Package contains significant programming tools. If one is willing to spend a little time becoming familiar with the PLOT Package and the PLOT language, then modifications to existing plots and addition of new plots may be accomplished with relative ease.

Acknowledgments

I would like to gratefully acknowledge the work of Don Becker which appeared in an article "How to Write Plots in DTR" in the Wombat Examiner, Volume 5, Number 3, pages 20-31. It is from this first article published on the plot language that much of the information in the second part of this presentation was taken. I would also like to acknowledge the assistance of Andy Schneider of Digital. In several conversations with Andy over the past year, I gained enough insight into the plot library that I begin to understand it in some meaningful way.

Discussion

[The following are excerpts from the question and answers which followed the presentation. Only a small part of the discussion is summarized here.]

Doug Wegscheid, Whirlpool Corporation: Is it true that if you make a change to one of your plots after you have done a SET PLOTS, you have to get out of DATATRIEVE and get back in for your changes to take effect?

Gallagher: Plots are just like any other dictionary object. Since there is no RELEASE PLOT-NAME command, you do have to exit and re-enter DATATRIEVE for your editing changes to take effect.

Jane Hesler, Medical College of Virginia: When a new version of DTR gets installed, what happens to plots in the plot library?

Gallagher: They all get blown away. The way to avoid this is to keep a source code copy of all your plots in a separate account. Then re-install these plots after DTR has been update. Another method is to keep a copy of the plot library on another node of the CDD which would not be affected by a DTR update. Your own dictionary node is one possibility.

Eric Mathew, Teradyne: Have you tried using other DATATRIEVE statements within a DEFINE PLOT?

Gallagher: Yes, I have tried a few. The statements I have given are apparently the only legal statements with the plot language.

Mathew: One comment on ReGIS code, I also played with the HARDCOPY plot, and I had to buy a manual.

Gallagher: A word of caution about ReGIS code, the implementation of ReGIS code on a VT125, VT240, GIGI, Rainbow emulator, DECmate emulator, and PRO emulator is slight different. There's ReGIS code and then there's ReGIS code.

Doug Wegscheid: It's very important to make a complete copy of everything in the plot library.

Gallagher: It is very important in the plot development stage that you do not do your development on the production plot library. Users don't like that.

Unknown female questioner: Is there a way to get your plots to the pen plotter?

Gallagher: The pen plotter, of course, must be able to handle ReGIS code. For a simple plot use the construct PLOT PLOT-NAME on PLOTFILE. Then transfer the file to the pen plotter. However, the plot library routines are based on the assumption that you are plotting to a ReGIS terminal with a hardcopy device.

Brian Lockery, ITT: I have a VT125 and an LA50 hardcopy device. I have a procedure that prints out about 50 plots in a row. How do I get a form feed between the hardcopy plots?

Gallagher: This is a particular nasty problem which has to do with the timing between the computer, VT125, and LA hardcopy device. The computer must be prevented from issuing the sequence "<ESC>[5i<FF><ESC>[4i" before the LA device physically finishes making the hardcopy. This problem is discussed and solved in the Wombat Examiner, Volume 5, Number 4, pages 42-44.

Modifying VAX-DTR Plots

Bart Z. Lederman, Brooklyn, NY

I had some data which I wanted to plot in VAX-DATATRIEVE. It happened that the data was best displayed with the X (horizontal) axis linear, and the Y (vertical) axis Logarithmic. In addition, I wanted several variables plotted together. I found that DATATRIEVE would plot the proper axis (X_LOGY), or several variables (MULTI_LINE), but not multiple lines with Log Y. I was able to add this plot relatively easily by comparing plots X_Y and X_LOGY, and transposing the differences into a copy of MULTI_LINE.

The first step in adding a plot is to extract the old plot used as a template, and then rename it so it won't replace the original (people might still be using it, and they'll get upset if it disappears). I called my version MULTI_LOGY, and it is as follows:

```
!
!   This plot is used the same way that MULTI_LINE is, except that
!   the Y axis values are logs rather than linear.
!
DELETE MULTI_LOGY;
REDEFINE PLOT MULTI_LOGY
DECLARE X_REF, X_LENGTH, X_MIN, X_MAX
DECLARE Y_REF, Y_LENGTH, Y_MIN_VALUE, Y_MAX_VALUE, D_LABEL
DECLARE I, J, K, N, COUNT, MX, WIDTH
DECLARE VECTOR X, Y1, Y2, Y3, Y_MIN, Y_MAX, Y_MX, COLOR
DECLARE STRING VECTOR Y_LABEL, CHR
ENTRY 0 (X_LABEL : STRING,
        LABEL_1 : STRING,
        LABEL_2 : STRING,
        LABEL_3 : STRING)
BEGIN
  PLOT HOUSEKEEP 0
  SET_SEGMENT 0
  PRINT `L(A2)`
  PRINT `L"5"00FF81818181FF`
  PRINT `L"6"0018244281422418`
  PRINT `L"0"00FF814242242418`
  OUTPUT_SEGMENT 0
  SET_SEGMENT 1
  X_REF = 100   Y_REF = 360   X_LENGTH = 600   Y_LENGTH = 350
  PLOT LABEL 0 (X_REF, Y_REF, X_LENGTH, Y_LENGTH, X_LABEL, D_LABEL)
  CHR(1) = `6`   COLOR(1) = 1   Y_LABEL(1) = LABEL_1
  CHR(2) = `5`   COLOR(2) = 2   Y_LABEL(2) = LABEL_2
  CHR(3) = `0`   COLOR(3) = 3   Y_LABEL(3) = LABEL_3
  INCR I OVER Y_LABEL
  IF LENGTH(Y_LABEL(I)) NE 0 THEN
    COUNT = COUNT + 1
END
```

```

ENTRY 1 (X_VALUE, Y1_VALUE, Y2_VALUE, Y3_VALUE)
BEGIN
  X(SIZE(X)+1) = X_VALUE
  Y1(SIZE(Y1)+1) = LOG (Y1_VALUE)      ! change here for log value
  Y2(SIZE(Y2)+1) = LOG (Y2_VALUE)      ! change here for log value
  Y3(SIZE(Y3)+1) = LOG (Y3_VALUE)      ! change here for log value
END
ENTRY 2
BEGIN
  Y_MIN(1) = MIN(Y1)   Y_MIN(2) = MIN(Y2)   Y_MIN(3) = MIN(Y3)
  Y_MAX(1) = MAX(Y1)   Y_MAX(2) = MAX(Y2)   Y_MAX(3) = MAX(Y3)
  Y_MIN_VALUE = MIN(Y_MIN)
  Y_MAX_VALUE = MAX(Y_MAX)
  PLOT LABEL 5 (Y_MIN_VALUE, Y_MAX_VALUE) ! change LABEL 3 to LABEL 5
  X_MIN = MIN(X)
  X_MAX = MAX(X)
  PLOT LABEL 2 (X_MIN, X_MAX, X)
  SORT(X, Y1, Y2, Y3)
  PRINT ^T(BA2S[8,16])^
  IF COUNT GE 1 THEN
    PLOT MULTI LOGY 3 (Y1)
  IF COUNT GE 2 THEN
    PLOT MULTI LOGY 3 (Y2)
  IF COUNT GE 3 THEN
    PLOT MULTI LOGY 3 (Y3)
  PRINT ^T(E)^
  WIDTH = X_LENGTH / 30
  Y_MX(30) = 0
  INCR I OVER X
  BEGIN
    Y_MIN(1) = Y1(I)   Y_MIN(2) = Y2(I)   Y_MIN(3) = Y3(I)
    MX = MIN(Y_MIN)
    J = ((X(I) - X_REF) / WIDTH) + 1
    Y_MX(J) = 1000
    IF Y_MX(J) GT MX THEN
      Y_MX(J) = MX
  END
  INCR I OVER Y_MX
  IF (I NE 1) AND (Y_MX(I) EQ 0) THEN
    Y_MX(I) = Y_MX(I - 1)
  OUTPUT SEGMENT 1
  PLOT LEGEND 4 (X_REF, Y_REF, X_LENGTH, Y_LENGTH, WIDTH, Y_MX,
    CHR, COLOR, Y_LABEL)
  PLOT HOUSEKEEP 2
END
ENTRY 3 (Y : VECTOR)
BEGIN
  PLOT LABEL 8 (Y)
  N = N + 1
  PRINT ^P^, LXY(X(1), Y(1)), ^W(I^, CVT(COLOR(N)), ^)^
  INCR I OVER X
  PRINT ^V^, LXY(X(I), Y(I))
  PRINT ^W(R)^

```

```

    INCR I OVER X
      PRINT 'p', LXY(X(I)-4, Y(I)-10), 't', QUOTE(CHR(N))
    PRINT 'w(v)'
  END
END_PLOT

```

One change not immediately obvious is in the next to last PRINT statement: the literals 'p' and 't' are in lower case. I found this to be necessary for the graph to come out properly. The LABEL ENTRY change from 3 to 5 was also found to be necessary for the Y-axis to be labeled in the normal Log scale; otherwise, the exponent of the scale value is printed linearly (this might be useful to some people). If you want to use this plot, you will probably find it easier to extract MULTI_LINE and edit it rather than typing in the entire plot as printed here.

I also made an almost identical plot, MULTI_LOGY_LR, which is used in the same manner as MULTI_LOGY. The only difference is that the points are not connected. In stead, a linear regression line (best fit to the data) is drawn for each Y-axis variable.

```

!
!   This is used exactly the same way as MULTI_LINE or MULTI_LOGY
!   that in addition to marking the data points it draws the linear
!   regression line.
!
DELETE MULTI_LOGY_LR;
REDEFINE PLOT MULTI_LOGY_LR
DECLARE X_REF, X_LENGTH, X_MIN, X_MAX
DECLARE Y_REF, Y_LENGTH, Y_MIN_VALUE, Y_MAX_VALUE, D_LABEL
DECLARE I, J, K, N, COUNT, MX, WIDTH
DECLARE VECTOR X, Y1, Y2, Y3, Y_MIN, Y_MAX, Y_MX, COLOR
DECLARE STRING VECTOR Y_LABEL, CHR
ENTRY 0 (X_LABEL : STRING,
        LABEL_1 : STRING,
        LABEL_2 : STRING,
        LABEL_3 : STRING)
BEGIN
  PLOT HOUSEKEEP 0
  SET_SEGMENT 0
  PRINT 'L(A2)'
  PRINT 'L"5"00FF8181818181FF'
  PRINT 'L"6"0018244281422418'
  PRINT 'L"0"00FF814242242418'
  OUTPUT_SEGMENT 0
  SET_SEGMENT 1
  X_REF = 100    Y_REF = 360    X_LENGTH = 600    Y_LENGTH = 350
  PLOT LABEL 0 (X_REF, Y_REF, X_LENGTH, Y_LENGTH, X_LABEL, D_LABEL)
  CHR(1) = '6'    COLOR(1) = 1    Y_LABEL(1) = LABEL_1
  CHR(2) = '5'    COLOR(2) = 2    Y_LABEL(2) = LABEL_2
  CHR(3) = '0'    COLOR(3) = 3    Y_LABEL(3) = LABEL_3
  INCR I OVER Y_LABEL
  IF LENGTH(Y_LABEL(I)) NE 0 THEN
    COUNT = COUNT + 1
END

```

```

ENTRY 1 (X_VALUE, Y1_VALUE, Y2_VALUE, Y3_VALUE)
BEGIN
  X(SIZE(X)+1) = X_VALUE
  Y1(SIZE(Y1)+1) = LOG (Y1_VALUE)      ! change here for LOG value
  Y2(SIZE(Y2)+1) = LOG (Y2_VALUE)      ! change here for LOG value
  Y3(SIZE(Y3)+1) = LOG (Y3_VALUE)      ! change here for LOG value
END
ENTRY 2
BEGIN
  Y_MIN(1) = MIN(Y1)   Y_MIN(2) = MIN(Y2)   Y_MIN(3) = MIN(Y3)
  Y_MAX(1) = MAX(Y1)   Y_MAX(2) = MAX(Y2)   Y_MAX(3) = MAX(Y3)
  Y_MIN_VALUE = MIN(Y_MIN)
  Y_MAX_VALUE = MAX(Y_MAX)
  PLOT LABEL 5 (Y_MIN_VALUE, Y_MAX_VALUE) ! change LABEL 3 to LABEL 5
  X_MIN = MIN(X)
  X_MAX = MAX(X)
  PLOT LABEL 2 (X_MIN, X_MAX, X)
  PRINT 'T(BA2S[8,16])'
  N = 0
  IF COUNT GE 1 THEN
    PLOT MULTI LOGY_LR 3 (Y1)
  IF COUNT GE 2 THEN
    PLOT MULTI LOGY_LR 3 (Y2)
  IF COUNT GE 3 THEN
    PLOT MULTI LOGY_LR 3 (Y3)
  PRINT 'T(E)'
  WIDTH = X_LENGTH / 30
  Y_MX(30) = 0
  INCR I OVER X
  BEGIN
    Y_MIN(1) = Y1(I)   Y_MIN(2) = Y2(I)   Y_MIN(3) = Y3(I)
    MX = MIN(Y_MIN)
    J = ((X(I) - X_REF) / WIDTH) + 1
    Y_MX(J) = 1000
    IF Y_MX(J) GT MX THEN
      Y_MX(J) = MX
  END
  INCR I OVER Y_MX
  IF (I NE 1) AND (Y_MX(I) EQ 0) THEN
    Y_MX(I) = Y_MX(I - 1)
  OUTPUT_SEGMENT 1
  PLOT LEGEND 4 (X_REF, Y_REF, X_LENGTH, Y_LENGTH, WIDTH, Y_MX,
    CHR, COLOR, Y_LABEL)
  PLOT HOUSEKEEP 2
END
ENTRY 3 (ARRAY : VECTOR)
BEGIN
  PLOT LABEL 8 (ARRAY)
  N = N + 1
  PRINT 'W(RI', CVT(COLOR(N)), ')'
  PLOT LABEL 7
  INCR I OVER X
  BEGIN
    PRINT 'p', LXY(X(I)-4, ARRAY(I)-10), 't', QUOTE(CHR(N))
  END

```

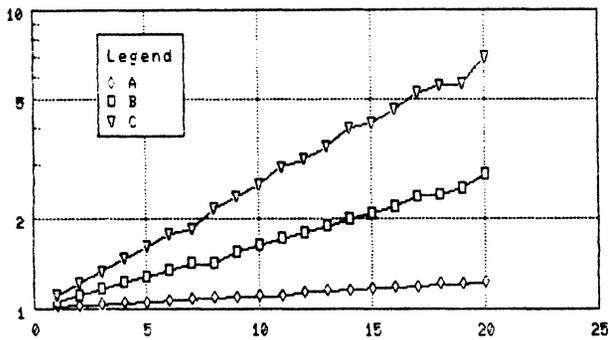
```

PRINT 'W(V)'
END
END_PLOT

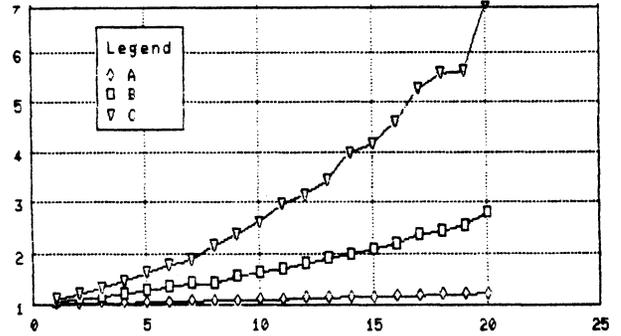
```

Attached are examples of both plots using some sample data. The MULTI_LOGY plot is shown in Figure 1 and the MULTI_LOGY_LR plot is given in Figure 3. I have also included the same data plotted with MULTI_LINE and MULTI_LR for comparison in Figures 2 and 4. The difference is most obvious at the top right-hand end of the "C" variable line (the topmost data line). It can be seen that a logarithmic graph fits this data better than a linear graph.

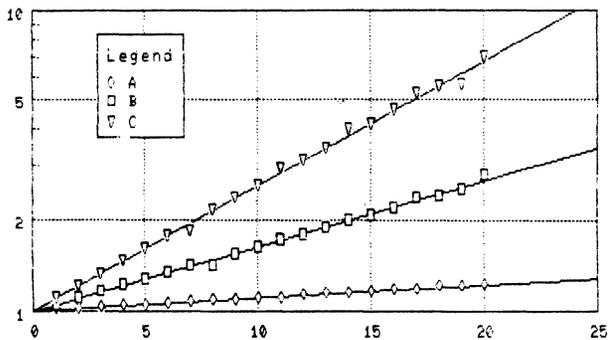
Example of PLOT MULTI_LOGY
Figure 1.



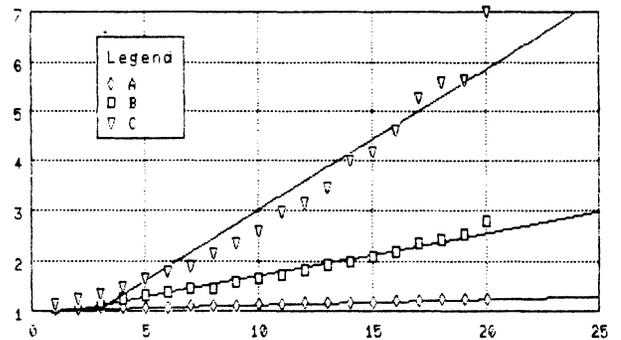
Comparison with PLOT MULTI_LINE
Figure 2.



Example of PLOT MULTI_LOGY_LR
Figure 3.



Comparison with PLOT MULTI_LR
Figure 4.



Additional DATATRIEVE Plots

Donald E. Stern, Jr., Warner Lambert Company, Milford, CT

At the recent DECUS symposium in New Orleans, Joe Gallagher presented an excellent talk on the DATATRIEVE Plotting Language internals. At that time Joe indicated that some of the customized plots which he and others had developed would be given in the Wombat Examiner. Below are some customized plots which have been generated at this location.

First, several of our users had complained that the linear regression plotting routine supplied by Digital did print the regression statistics (slope, intercept, and correlation coefficient.) The following plot was developed which combines the features of the DEC supplied plots X_Y and LR; it produces the scatter diagram, the regression line, and prints the regression statistics as well.

```
DELETE LINREG;
DEFINE PLOT LINREG
!
!This plotting routine is based upon the DEC distributed code for
!the plot X_Y. In this implementation a regression line is plotted
!automatically and the regression coefficients are displayed.
!
DECLARE X_AXIS, Y_AXIS, X_LENGTH, Y_LENGTH, X_MAX, X_MIN, Y_MIN, Y_MAX
DECLARE X_POS, Y_POS, I
DECLARE VECTOR XS, YS
DECLARE REAL SUM_X, SUM_Y, SUM_X_Y, SUM_X_SQ, SUM_Y_SQ, R_SQ, R,
              SP_XY, SP_X, SP_Y, N, MEAN, SLOPE, INTERCEPT
ENTRY 0 (X_LABEL : STRING, Y_LABEL : STRING)
  BEGIN
    PLOT HOUSEKEEP 0
    SET SEGMENT 1
    X_AXIS = 100
    Y_AXIS = 360
    X_LENGTH = 600
    Y_LENGTH = 350
    PLOT LABEL 0 (X_AXIS, Y_AXIS, X_LENGTH, Y_LENGTH, X_LABEL, Y_LABEL)
    SUM_X = 0
    SUM_Y = 0
    SUM_X_SQ = 0
    SUM_Y_SQ = 0
    SUM_X_Y = 0
  END
ENTRY 1 (X : REAL : "horizontal coordinate", Y : REAL : "vertical coordinate")
  BEGIN
    XS (SIZE (XS) + 1 ) = X
    YS (SIZE (YS) + 1 ) = Y
    SUM_X = SUM_X + X
    SUM_Y = SUM_Y + Y
    SUM_X_SQ = SUM_X_SQ + X**2
```

```

        SUM_Y_SQ = SUM_Y SQ + Y**2
        SUM_X_Y = SUM_X_Y + X * Y
    END
! Print scatter plot
ENTRY 2
    BEGIN
        N = SIZE (XS)
        MEAN = SUM_X / N
        SLOPE = (SUM_X_Y - MEAN * SUM_Y) / (SUM_X_SQ - MEAN * SUM_X)
        INTERCEPT = (SUM_Y - SLOPE * SUM_X) / N
        SP_XY = SUM_X_Y - (SUM_X * SUM_Y) / N
        SP_X = SUM_X_SQ - (SUM_X * SUM_X / N)
        SP_Y = SUM_Y_SQ - (SUM_Y * SUM_Y / N)
        R_SQ = (SP_XY * SP_XY) / (SP_X * SP_Y)
        R = SQRT(R_SQ)
!
        X_MIN = MIN (XS)
        X_MAX = MAX (XS)
        Y_MAX = MAX (YS)
        Y_MIN = MIN (YS)
        IF Y_MIN > 0
            THEN Y_MIN = 0
        PLOT LABEL 2 (X_MIN, X_MAX, XS)
        PLOT LABEL 3 (Y_MIN, Y_MAX)
        PLOT LABEL 8 (YS)
        INCR I OVER XS
            PRINT CENTER (XS(I), YS(I)-9, '+', 9)
!
!Print the regression statistics
        PRINT 'P', LXY(20, 393),
            'T(s1)', QUOTE('Slope = '),
            'T(s1)', QUOTE( CVT(SLOPE) )
        PRINT 'P', LXY(20, 415),
            'T(s1)', QUOTE('Intercept = '),
            'T(s1)', QUOTE( CVT(INTERCEPT) )
        PRINT 'P', LXY(20, 437),
            'T(s1)', QUOTE('Cor. Coef. = '),
            'T(s1)', QUOTE( CVT(R) )
!
        PLOT LABEL 7
        OUTPUT_SEGMENT 1
        PLOT HOUSEKEEP 2
    END
END_PLOT

```

In another instance, a user was generating a large number of records (512) containing x,y data. Using the plot X_Y, and CONNECT produced a very busy graph because of the 512 "+" markers used to plot individual points. In order to clean up the display, a new plot X_Y_CON was created which combined the features of the DEC supplied plots into a single plot which outputs only a line connecting the data points. The code for this plots follows.

```

DELETE X_Y_CON;
DEFINE PLOT X_Y_CON
!
!This routine is based upon DEC code for the plots X_Y and CONNECT. It
!plots data using connected lines rather than plotting lines between
!data marker symbols as the combination of the two DEC supplied plots does.
!
DECLARE X_AXIS, Y_AXIS, X_LENGTH, Y_LENGTH, X_MAX, X_MIN, Y_MIN, Y_MAX
DECLARE X_POS, Y_POS, I
DECLARE VECTOR XS, YS
ENTRY 0 (X_LABEL : STRING, Y_LABEL : STRING)
  BEGIN
    PLOT HOUSEKEEP 0
    SET_SEGMENT 1
    X_AXIS = 100
    Y_AXIS = 360
    X_LENGTH = 600
    Y_LENGTH = 350
    PLOT LABEL 0 (X_AXIS, Y_AXIS, X_LENGTH, Y_LENGTH, X_LABEL, Y_LABEL)
  END
ENTRY 1 (X : REAL : "horizontal coordinate", Y : REAL : "vertical coordinate")
  BEGIN
    XS (SIZE (XS) + 1 ) = X
    YS (SIZE (YS) + 1 ) = Y
  END
! Print scatter plot
ENTRY 2
  BEGIN
    X_MIN = MIN (XS)
    X_MAX = MAX (XS)
    Y_MAX = MAX (YS)
    Y_MIN = MIN (YS)
    IF Y_MIN > 0
      THEN Y_MIN = 0
    PLOT LABEL 2 (X_MIN, X_MAX, XS)
    PLOT LABEL 3 (Y_MIN, Y_MAX)
    PLOT LABEL 8 (YS)
    PLOT LABEL 9(XS,YS,Y_AXIS)
    SORT (XS,YS)
    PRINT 'P', LXY (XS(1), YS(1)), 'V'
    INCR I OVER XS
      PRINT LXY(XS(I), YS(I))
    OUTPUT_SEGMENT 1
    PLOT HOUSEKEEP 2
  END
END_PLOT

```

At our location, several users have LA50 printers connected to the printer ports of their terminals. One of the problems with the DEC supplied HARDCOPY plot is that the printed output is not centered on 8 1/2 x 11 tractor feed paper, it is shifted to the left. The following plot was created in order to center the plot on the paper.

```

DELETE HARDCOPY_CENTERED;
!
!This plotting routine shifts hardcopy output to the right in order to
!center it on 8 1/2 x 11 tractor feed paper on an LA50 printer
!
REDEFINE PLOT HARDCOPY_CENTERED
ENTRY 0
  BEGIN
    PLOT HOUSEKEEP 4
    OUTPUT_SEGMENT 5
    CLEAR_SEGMENT 4
    SET_SEGMENT 4
    PRINT ^S(H(P[175,0]))S(E)P[100,200]@BS(H)^
    OUTPUT_SEGMENT 5,4,6
    PLOT HOUSEKEEP 2
  END
END_PLOT

```

The final plot given is one created by one of our users, T. Hession. Basically, he wanted to be able to create plots similar to those created by the DEC supplied MULTI_LINE routine but to use the DATE data type. The following plot was created to meet that need.

```

DELETE DATE_MULTI_LINE;
!
! Modification of MULTI_LINE to create DATE_MULTI_LINE
!
REDEFINE PLOT DATE_MULTI_LINE
DECLARE X_REF, X_LENGTH, X_MIN, X_MAX
DECLARE Y_REF, Y_LENGTH, Y_MIN_VALUE, Y_MAX_VALUE, D_LABEL
DECLARE I,J,K,N, COUNT, MX, WIDTH
DECLARE VECTOR DATES, Y1, Y2, Y3, Y_MIN, Y_MAX, Y_MX, COLOR
DECLARE STRING VECTOR Y_LABEL, CHR
ENTRY 0 (X_LABEL : STRING,
        LABEL_1 : STRING,
        LABEL_2 : STRING,
        LABEL_3 : STRING)
BEGIN
  PLOT HOUSEKEEP 0
  SET_SEGMENT 0
  PRINT ^S(MO(L0)1(L33)2(L66)3(L99))^
  PRINT ^S(MO(AD)1(AR)2(AC)3(AY))^
  PRINT ^L(A2)^
  PRINT ^L"5"00FF8181818181FF^
  PRINT ^L"6"0018244281422418^
  PRINT ^L"0"00FF814242242418^
  OUTPUT_SEGMENT 0
  SET_SEGMENT 1
  X_REF = 100      Y_REF = 360      X_LENGTH = 600      Y_LENGTH = 350
  PLOT LABEL 0 (X_REF, Y_REF, X_LENGTH, Y_LENGTH, X_LABEL, D_LABEL)
  CHR(1) = ^6^    COLOR(1) = 1    Y_LABEL(1) = LABEL_1
  CHR(2) = ^5^    COLOR(2) = 2    Y_LABEL(2) = LABEL_2
  CHR(3) = ^0^    COLOR(3) = 1    Y_LABEL(3) = LABEL_3

```

```

      INCR I OVER Y_LABEL
        IF LENGTH(Y_LABEL(I)) NE 0 THEN
          COUNT = COUNT + 1
END
ENTRY 1 (X : DATE, Y1_VALUE, Y2_VALUE, Y3_VALUE)
BEGIN
  DATES (SIZE (DATES) + 1 ) = X
  Y1(SIZE(Y1)+1) = Y1_VALUE
  Y2(SIZE(Y2)+1) = Y2_VALUE
  Y3(SIZE(Y3)+1) = Y3_VALUE
END
ENTRY 2
BEGIN
  Y_MIN(1) = MIN(Y1)   Y_MIN(2) = MIN(Y2)   Y_MIN(3) = MIN(Y3)
  Y_MAX(1) = MAX(Y1)   Y_MAX(2) = MAX(Y2)   Y_MAX(3) = MAX(Y3)
  Y_MIN_VALUE = MIN(Y_MIN)
  Y_MAX_VALUE = MAX(Y_MAX)
  PLOT LABEL 3 (Y_MIN_VALUE, Y_MAX_VALUE)
  X_MIN = MIN (DATES)
  X_MAX = MAX (DATES)
  PLOT LABEL 6 (X_MIN, X_MAX, DATES)
  SORT(DATES, Y1, Y2, Y3)
  PRINT 'T(BA2S[8,16])'
  IF COUNT GE 1 THEN
    PLOT DATE_MULTI_LINE 3 (Y1)
  IF COUNT GE 2 THEN
    PLOT DATE_MULTI_LINE 3 (Y2)
  IF COUNT GE 3 THEN
    PLOT DATE_MULTI_LINE 3 (Y3)
  PRINT 'T(E)'
  WIDTH = X_LENGTH / 30
  Y_MX(30) = 0
  INCR I OVER DATES
  BEGIN
    Y_MIN(1) = Y1(I)   Y_MIN(2) = Y2(I)   Y_MIN(3) = Y3(I)
    MX = MIN(Y_MIN)
    J = ((DATES(I) - X_REF) / WIDTH) + 1
    Y_MX(J) = 1000
    IF Y_MX(J) GT MX THEN
      Y_MX(J) = MX
  END
  INCR I OVER Y_MX
  IF (I NE 1) AND (Y_MX(I) EQ 0) THEN
    Y_MX(I) = Y_MX(I - 1)
  OUTPUT_SEGMENT 1
  PLOT LEGEND 4 (X_REF, Y_REF, X_LENGTH, Y_LENGTH, WIDTH, Y_MX,
    CHR, COLOR, Y_LABEL)
  PLOT HOUSEKEEP 2
END
ENTRY 3 (Y : VECTOR)
BEGIN
  PLOT LABEL 8 (Y)
  N = N + 1
  PRINT 'P', LXY(DATES(1), Y(1)), 'W(I', CVT(COLOR(N)), ')

```

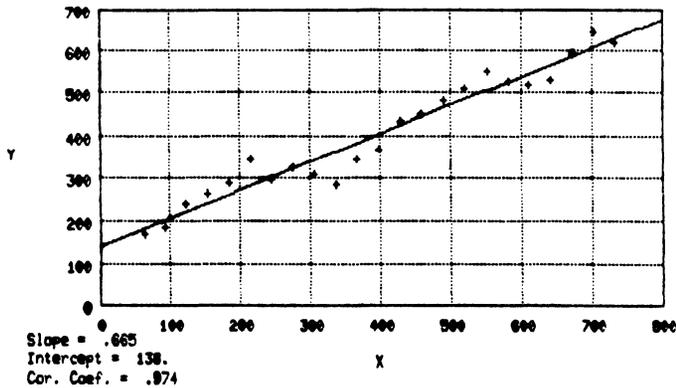
```

INCR I OVER DATES
PRINT 'V', LX(DATES(I), Y(I))
PRINT 'W(R)'
INCR I OVER DATES
PRINT 'P', LX(DATES(I)-4, Y(I)-10), 'T', QUOTE(CHR(N))
PRINT 'W(V)'
END
END_PLOT

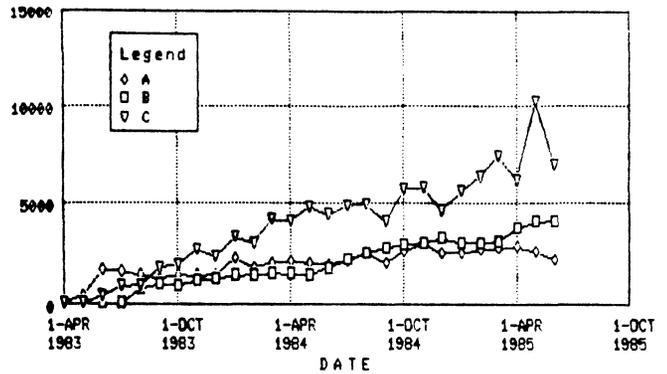
```

An example of the linear regression plot LINREG with intercept, slope, and correlation coefficient is shown in Figure 1. The capability to plot several lines against a date is illustrated in Figure 2 with the plot DATE_MULTI_LINE. When a very large number of points are plotted with X_Y_CON as in Figure 3, the plot is "less busy" than the corresponding plot using X_Y as show in Figure 4.

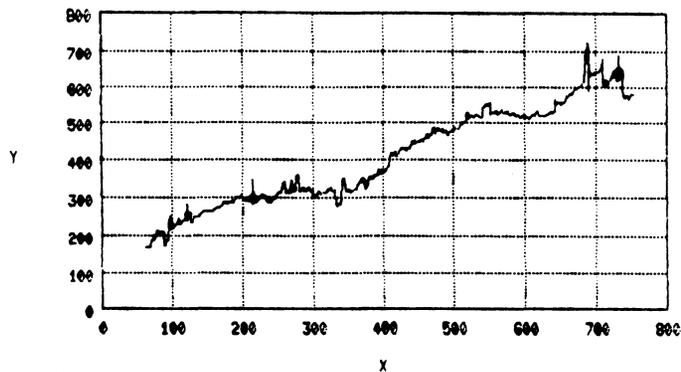
Example of PLOT LINREG
Figure 1.



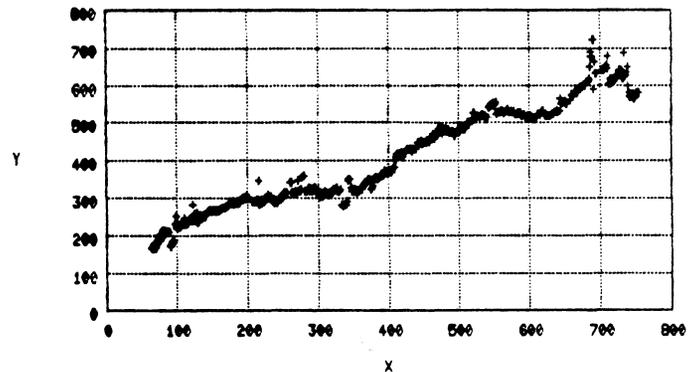
Example of PLOT DATE_MULTI_LINE
Figure 2.



Example of PLOT X_Y_CON
Figure 3.

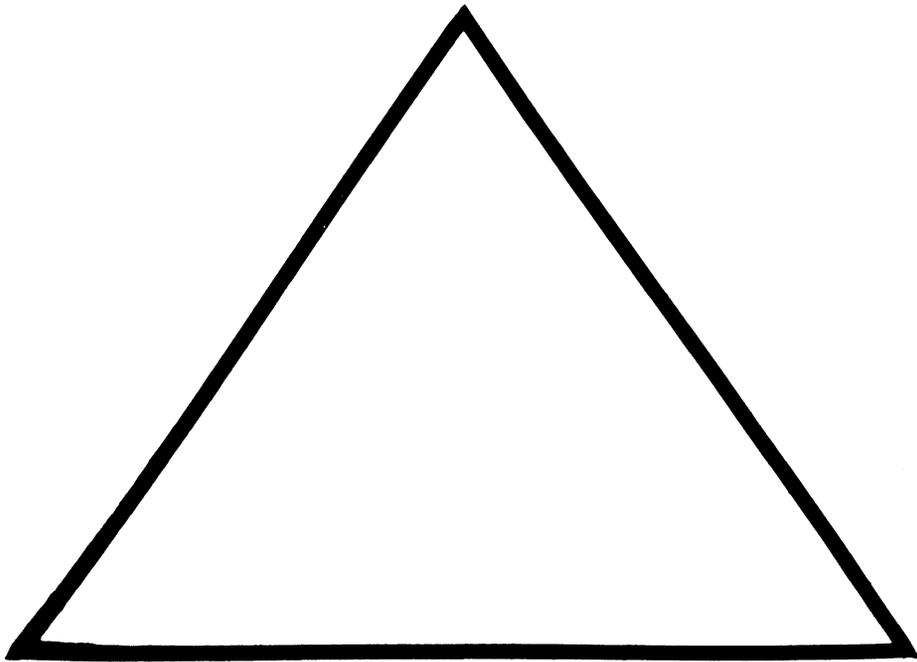


Comparison with PLOT X_Y
Figure 4.





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

Edusig Home Runs

HOME RUNS

The little boy with the big brown eyes and sandy hair looked at his grandfather with love and admiration. He said "Grandfather, I would do just about anything to get a home run in tomorrow's baseball game." The old man looked down through his glasses at his grandson with a kind and gentle smile. "Well, I tell you, Son, I think I have just what the doctor ordered", said the Grandfather. The boy said with a great deal of enthusiasm, "What is it?" Grandfather walked slowly over to the closet and opened the door. He reached up to the top most shelf pulling down what seemed at first glance to be a simple shoebox. "Grandfather, this is just an ordinary box", said the boy. "Yes, it may seem ordinary on the outside, but it is very special on the inside," said the Grandfather. "Let's open it!" said the boy with the big brown eyes. "If we did, what would you expect to find, Son?" said the Grandfather. The boy said with a great big smile, "A home run."

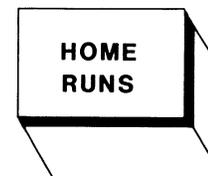
My name is Fred Bell. I have a special empty box that can contain your home runs, your ideas, your happenings concerning computers in education. As editor for EDUSIG this year, I earnestly solicit your input for future issues of the newsletter.

EDU-1

Those of us in the educational scene can get too involved in day-to-day activities and not make available our good works and good perceptions to the rest of our colleagues. Please use this forum to promote the use and understanding of computers in education.

We don't need all home runs--singles and doubles add up! Send your short articles, idea pieces, technical reviews, etc. to:

Fred Bell
EDUSIG Editor
Taft College
29 Emmons Park Dr.
Taft, CA 93268
(805) 763-4282



EDU-2

WHAT (OR WHO) IS A LIGHTHOUSE?

light' house n.: a tower or other building equipped to guide navigators by means of a powerful light that gives a continuous or interrupted signal (Webster's 3rd New International Dictionary, 1971, V. 2, p. 1309)

In 1983 and 1984, Digital Equipment Corporation selected several colleges and universities to participate in their new LIGHTHOUSE program. These LIGHTHOUSE institutions would guide other educational navigators successfully through the sea of computer-based education (CBE). Digital has an extensive computer-based educational product line called Renaissance that encompasses a large number of tools for CBE development and delivery. The participating schools would explore the potential of these tools, using them to produce high quality CBE materials. The LIGHTHOUSE program also offers new developers an opportunity to visit the sites and learn from the experience of these pioneer CBE developers.

The colleges and universities were carefully selected by Digital to provide information about various educational settings. Parameters such as the school size, its geographic location, the particular CBE development model used, and the type of CBE materials being produced vary from site to site and demonstrate many possibilities for CBE use at the college and university level.

The schools range in size from fairly large universities to small community colleges. They are

located literally from east coast to west coast. Each school utilizes its own development model for producing CBE materials. Some sites have a staff of professional programmers and instructional designers who work with faculty content experts to produce the CBE modules. Other sites have their computer staff train and advise faculty developers. Some schools are concentrating on entire CAI courses within one discipline.

Each school demonstrates a unique set of parameters for utilizing CBE in the educational setting. Developers planning to visit a LIGHTHOUSE will be able to find one combination that most closely resembles the conditions at their own institution. In that way, visitors carry away ideas and models that will work for them.

Originally the schools were asked to develop modules using the new CBE authoring language that Digital has produced, the Digital Authoring Language (DAL). DAL is a programming language with customized additions for computer-based education applications, such as paging, answer judging, and scoring. Graphics are easily incorporated into the lessons with such tools as the ReGIS Graphics Editor and the Character Set Editor. Finished modules are delivered to users via the router and record-keeping system, the Courseware Authoring System (C.A.S.).

As Digital adds to the Renaissance product line with more CBE tools, the schools are asked to evaluate these new products. One addition, the Courseware Design System, a set of templates for non-programmers to create CBE modules, has been explored at several LIGHTHOUSE schools. The

template idea may be one way to reduce the large number of hours required for traditional CBE module development and to minimize the need to learn a programming language.

The LIGHTHOUSE program has brought many benefits to Digital, to the LIGHTHOUSE schools, and to the user:

- Digital receives high quality software from the experienced LIGHTHOUSE CBE developers in Digital's CBE language
- Digital receives valuable feedback on updated versions and on new products, and is very responsive to suggestions for improvements
- Digital has a network of sites available to exhibit several successful development models to potential CBE developers
- The LIGHTHOUSES are at the leading edge of Digital's CBE development process, and are able to provide feedback to Digital for improvements and enhancements to the CBE development environment
- The LIGHTHOUSES receive the latest CBE products from Digital for testing and evaluation
- The users have better tested products available to them, since Digital does make changes to products based on feedback from the LIGHTHOUSES
- The users have the opportunity to visit the sites, to examine the CBE process directly, and to discuss with the developers any questions they might have about CBE.

The LIGHTHOUSE sites, sponsored by Digital Equipment Corporation's Education Computer Systems (ECS), Marlboro, Mass., are listed below:

Iowa State University
Ames, Iowa 50011
Clair Maple, Director

Indiana University
Academic Computing Services
Bloomington, Indiana 47405
Doug Grover, Director

Taft Community College
Taft, California 93268
David Cothrun, President

University of Delaware
Office of Computer-Based Instruction
Newark, Delaware 19716
Mary Jac Reed, VAX Project Manager

Your local Digital sales representative is able to provide you with more specific information about the LIGHTHOUSE program.

Articles on the individual LIGHTHOUSES and their specific activities are planned for future issues of THE SIGS NEWSLETTER.

Submitted by Mary Jac Reed
University of Delaware





DECUS

GRAPHICS



GRAPHICS

Chairman

William Kramer
University of Delaware
Newark, DE

Library Committee

James M. Turner
Saber Technology
San Jose, CA

Symposium Coordinator

Bijoy Misra
Smithsonian Institution
Cambridge, MA

DEC Counterpart

Susan Usilton
Digital Equipment Corporation
Nashua, NH

Newsletter Editor

Michael P. Anton
Houston, TX

Information Officer

Mike York
Boeing Computer Services
Seattle, WA

Session Note Editor

Mike McPherson
Michigan State University
East Lansing, MI

Human Interface Working Group Coordinator

Tom Owens
Graphics Research Center
Baltimore, MD

Standards Coordinator

Jim Flatten
Ames Lab
Ames, IA

Engineering Working Group Coordinator

Dana Smith
Wilmington, DE

LETTER FROM THE EDITOR

FROM THE CHAIRMAN

For a long time within DECUS, most of what we had to discuss in the area of computer graphics was how to connect a particular terminal, which more than likely was not manufactured by DEC, to a DEC CPU. Another common topic was how to write the software to drive one of these devices -- not necessarily the most exciting of conversations!

Now, however, things appear to be changing due to two major developments. The first is something the GAPSIG and the graphics industry have been involved in for some time: computer graphics standards. Starting in 1977, the development of the ACM/SIGGRAPH CORE proposal presented the idea of standardizing device independent graphics software. This has now reached maturity with the acceptance of GKS by ANSI and the development of PHIGS. These efforts have made the concept of the graphics workstation, which is far more powerful than a traditional terminal, an important and central issue to the industry.

It now appears that the workstation hardware is also reaching maturity and it is also becoming affordable! As you have certainly heard by now, Digital announced the VAXstation II and the VAXstation 500 this spring. Along with the IBM 5080, this marks a significant change in the graphics environment, since it is possible to get a complete, single-vendor solution to graphics needs from major computer manufacturers. We can look forward to integrated systems, including a powerful CPU, workstations, and graphics software. Considering also the explosion in the number of third party workstations, it is easy to predict that workstations will become cheaper, better and more prevalent.

The GAPSIG has long thought that the combination of graphics standards and good workstations will strongly contribute to making graphics universally available.

In future newsletters and symposia, the GAPSIG will invest a lot of energy and effort exploring the workstation concept and discussing their use. All this should add to an exciting year for the Graphics Applications SIG.

William Kramer
Graphics Applications SIG Chairman

I must start this letter off with an apology. During the past few months I have changed jobs and moved to Houston, Texas. It has been brought to my attention that the post office did not consistently forward my mail thus returning some submissions to the newsletter. If your submission was one that was returned please accept my apology and resubmit it the address given below.

Graphics Applications SIG will be publishing a section in the DECUS SIGs Newsletters. The combined newsletter format will allow the SIG to disseminate information on a more timely nature and allow more DECUS members to be informed about the happenings within the Graphics Application SIG.

The Graphics Applications SIG needs contributions in order to continue as an effective medium for exchange of information regarding Graphics. All contributions should be camera ready copy, e.g. sharp black type on 8 1/2 by 11 inch paper with 1 inch margins.

Michael Anton
Graphics Applications SIG Newsletter Editor
P.O. Box 591293
Houston, TX 77250-1293
(713) 928-4838

Speaking of Standards
Jim Flatten
GAPSIG Standards Coordinator

The fall DECUS Symposium occurs at a particularly appropriate time as far as graphics standards are concerned. At that time it is expected that a proposed graphics standard, PHIGS, will be available for public review. To help educate the DECUS graphics community about this new standard, GAPSIG has scheduled a Pre-Symposium Seminar titled "An Introduction to PHIGS". The seminar will be taught by Dr. Richard Puk, an independent computer graphics consultant from San Diego. Dr. Puk has been actively involved in the development of computer graphics standards since the formation of the ACM SIGGRAPH Graphics Standards Planning Committee. He was formerly the chair of X3H31, the task group working on PHIGS, and he is presently a member of both the ANSI Computer Graphics Technical Committee and the U.S. delegation to the ISO Computer Graphics Working Group. This seminar will be of interest to graphics programmers, analysts and managers as well as those responsible for the planning and development of graphics-based applications. Dr. Puk will also present the GAPSIG keynote session "Computer Graphics after Standardization".

PHIGS stands for the Programmers Hierarchical Interactive Graphics System. It occupies the same place in a conceptual graphics model that GKS and the CORE system do, ie. it is viewing software. However PHIGS is different than either of those systems in fundamental ways and it is aimed at a constituency that cannot be served with the functionality provided by GKS.

PHIGS bears a strong resemblance to GKS. This is no accident. Every effort has been made to be compatible with GKS wherever possible. However because of fundamental differences in the two systems there will be areas of incompatibility. These incompatibilities are deemed necessary to serve the PHIGS constituency.

The PHIGS system operates on a database of graphics primitives that is conceptually centralized. There may however be copies of the database stored in the graphics workstations as well. The elements of the database are called structures and are analogous to segments in the world of GKS and CORE.

Structures are collections of graphics primitives, attribute specifications, modeling transforms and control functions. One important control function is the EXECUTE function which permits the hierarchical structure referred to in the name PHIGS. In order to produce a picture, PHIGS "traverses" the database, executing each of the structure elements as they occur. In this respect, the PHIGS database is very much like the display list used by vector display terminals.

The traversal process gives rise to one of the most important differences between PHIGS and GKS. In GKS properties of graphics primitives are determined at the time they are added to segment storage. With PHIGS the attributes of primitives are determined

at the time the structures are traversed. Editing of structures and display of structures are completely independent processes. There are numerous functions defined in PHIGS to facilitate the editing of structures.

A consequence of the PHIGS architecture is that it may be impossible to determine the attributes of a graphics primitive until it is displayed. This can happen because a particular structure may be EXECUTED by one or more parent structures, each of which could have set attributes (e.g. linewidth, linestyle, color, etc) differently. In PHIGS a child inherits its attributes from its parents. Attributes are not tied to a particular primitive at the time a structure is created.

PHIGS is a three dimensional system. GKS is two dimensional but a three dimensional extension will probably be standardized before PHIGS becomes a standard. Members of the PHIGS task group have been involved in the design of 3D GKS and it is anticipated that the viewing pipelines of the two systems will be very similar.

The foregoing is a short description of some of the PHIGS functionality. I strongly recommend attending the Pre-Symposium Seminar if you think this type of system figures into your graphics future. It is a good chance to find out in detail what is in the proposed standard. See you in Anaheim.



H M S

HMS

Chairman VAX SIG Liaison
Thomas J. Provost
MIT/LNS Bates Linac Facility
Middletown, MA

UNIBUS Hardware
Ron Bogue
LIV Aerospace & Defense Co.
Dallas, TX

Larry Brown
Tektronix Inc.
Beaverton, OR

Storage
Marilyn Fedele
Digital Equipment Corporation
Maynard, MA

Symposium Coordinator
Mike Allen
Lawrence Livermore National Labs
Livermore, CA

Performance
William Wallace

DEC Counterparts
Terminals
Nina Abramson
Digital Equipment Corporation
Maynard, MA

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

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

Networks SIG Liaison
Sandra Traylor
Target Systems
Yorba Linda, CA

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

Printer
Frank Orlando
Digital Equipment Corporation
Maynard, MA

Publications Coordinator (Editor)
Bill K. Walker
Monsanto Research Corp.
Miamisburg, OH

SITE SIG Liaison
Emily Kitchen
A.H. Robbins Co.
Richmond, VA

Diagnostic
George D. Cooke
Digital Equipment Corporation
Maynard, MA

DECUS Europe Liaison
Hans Zoller

Session Notes
DAARC SIG Liaison
Bill Tippie
Kinetic Systems Corp.
Lockport, IL

SITE SIG Liaison
David Hunt
Lawrence Livermore National Lab.
Livermore, CA

RT-11 SIG Liaison
Gary Sallee
Sallee Software Consulting
Yorba Linda, CA

Standards Coordinator
CAMAC Working Group Coordinator
Peter Clout
Los Alamos National Lab
Los Alamos, NM

RSX SIG Liaison
Hans Jung
Associated Press
New York, NY

LUG Coordinator
Gregg Giesler
Los Alamos Science Lab
Los Alamos, NM

Members-At-Large
Mike Rembis
American Dade
Costa Mesa, CA

TOEM (Chips % Boards)
Jack J. Peterson
Horizon Data Systems
Richmond, VA

Pratap Gohel
E.I. Dupont
Ingleside, TX

HHK (Hardware Hints & Kinks)
Wayne Kesling
Monsanto Research Corp.
Miamisburg, OH

George Hamma
Synergistic Technology
Cupertino, CA

Hans Dahlke
Richland, WA

CONTENTS OF THIS ISSUE

- o FROM THE SIG CHAIR -- Tom Provost
- o FROM THE EDITOR -- William K. Walker
- o HOW TO REPORT HARDWARE HINTS AND KINKS (HHK's)
- o HARDWARE HINTS AND KINKS -- Tom Provost
- o OKIDATA PACEMARK 2410 PRINTER -- Gregory N. Brooks
- o DATA TRANSLATION DT 2782 A/D CONVERTER "PGH" OPTION
-- Gregory N. Brooks
- o KDJ11-?? OR "ALL 11/73'S ARE NOT CREATED EQUAL."
-- Wayne E. Kesling

FROM THE SIG CHAIR
Tom Provost

Notes from the chair:

I have re-assumed the job of HMS SIG Chairman, after many years of doing other things. As SIG Chairman I hope to improve the effectiveness of the SIG in serving the DECUS community. Unfortunately the advent of the single newsletter makes it impossible to speak to the membership of the SIG without talking to the world. Those of you who have subscribed to any newsletter will now receive the DECUS newsletter, which will include a compendium of all SIG newsletters. As a result we will no longer have any way to judge whether we indeed have an audience, or whether anyone is interested in the SIG. To alleviate this problem, I would like all those interested in the activities of the HMS SIG to make sure they are HMS SIG members. The SIG membership will not give you anything you wouldn't have gotten anyway, but it will give us some credibility in competing for DECUS resources.

As SIG Chairman I am attempting to improve the effectiveness of our SIG and PRODUCT liasons. If you have strong interest in any of these areas, please let me know. We are increasing the utilization of the HMS Suite at symposia, so if you are going to the Fall DECUS Symposium, plan to drop in the suite. One of our members is forming an HMS LUG. I would like to know if others have found this desirable. I am MITLUG chairman, and MITLUG is a LUG for PDP-11 users regardless of operating system. We often cover hardware topics, but also include software topics. We do not cover VAX, PC or Large System hardware. Let me know if you find hardware topics useful in your LUG.

In addition to the areas mentioned above, I would like to see the SIG more active outside the symposia context. Please send me your suggestions.

Tom Provost

HMS-2

FROM THE EDITOR
William K. Walker

As you have probably figured out by now, what you are holding in your hands is the first issue of the combined DECUS newsletter (dubbed informally "The Big One" or TBO). I will "draw the curtain of charity" over the events that led up to the establishment of TBO and address, instead, the effect that it will have on the content and format of the HMS SIG Newsletter.

The content, so I am told, will remain essentially unchanged. We will continue to publish the same type of material and the procedure for getting articles to me is the same.

The format, however, will be slightly different. There is (supposed to be) a separate section containing the "tear-out forms" for all of the SIG's. Thus, the Hardware Submission Form will be found in this section, rather than on the back page of the HMS SIG section. In addition, it is intended that the instructions for submitting articles be combined into a single section also. Finally, we currently intend to continue to publish on a quarterly basis. Since TBO comes out monthly, you should see HMS SIG articles in about every third issue. Note, however, that the "boilerplate" (cover page, steering committee roster, instructions for submitting articles, and, presumably, the Hardware Submission Form) will appear in every issue.

One final note. There are bound to be a few rough spots in this first issue of the combined newsletter. Let me know what you think. My address and phone are in the "instructions for submission" section.

HOW TO REPORT HARDWARE HINTS AND KINKS (HHK's)

In the "tear-out" section of this newsletter is a Hardware Submission Form. You may use this form to make the HMS SIG aware of any problems which you have encountered, or any improvements which you would like to see in Digital's hardware. Relevant items will be passed on to the appropriate people in Digital, and some will be published in the newsletter and/or presented at the next Hardware Hints and Kinks session.

HMS-3

HARDWARE HINTS AND KINKS
Tom Provost

VAX-11/750 has a filter that catches all the dirt swept off the floor. Cleaning of this filter is apparently not part of the PM schedule. Blocking of the filter causes air-flow sensor to complain. A similar problem exists in the HSC-50.

New MicroPDP-11 boxes have an air baffle to decrease air draw through floppies. This is important in rack-mount configuration in dirty environment.

Three-high RA81 cabs have had heating problems, but these must have been solved, since the 4-high cabs seem to work.

For those of you who like to hack with your MICROVAX II's, you might want to try

MOVCS5 (Rn)+,(Rn)+,(Rn)+,(Rn)+,(Rn)+

Results are interesting. "n" can be anything reasonable.

MicroVAX II systems with 1/4 meg boards have 33Mhz clocks, whereas those with full megabyte boards have 40Mhz. The board products do not have this limitation.

For those with large quantities of VT-240's not on contract, the FC0 to firmware level 2.1 can be obtained for less than advertised price if you negotiate.

Some RA81 disk drives had a glue problem on filter. This causes the HDA to "seem to fail". To avoid symptom keep under 80 degrees F and let it spin up and down normally (don't use AC switch). Problem is known by DIGITAL and is being attended to.

8600's can be crashed by FORTRAN programs. This results from a microcode problem. FC0 is coming. Early recipients of the fix report that it works.

TK25 tape drives appear write-protected when used with 3rd party tape of length less than 600'. (Some third party tape is only 300')

RT-11 V5.2 will fix problem of MT getting lost when interrupted in transfer.

Some questions have been raised as to whether 4-slot backplanes still have a problem. The problem appears when the

boards are used on 8600's, but those of us with complex configurations including 4-slot backplanes on other computers might want to stay tuned.

Moving RD51 drives from one system to daisychain onto another may require changing a jumper located under the drive.

DEC CDC drives (RM05) and CDC CDC drives (9766) have incompatible standards on the unit plugs. CDC's unit 0 is DEC's unit 15 DEC's unit 0 is CDC's unit 14.

Documentation update edition 3 of UDA50 manual indicates that UDA50 cannot be used with a bus repeater... no surprise for most of us.

Floppy disks with reinforced hubs damage drives... thickness problem. Even some of the cheaper disks have reinforced hubs. Reinforced hubs are not necessarily indicated on the outside of the box.

RD52 ... dip switch on drive selects unit #, bears no resemblance to software unit #'s. e.g. set at "3" to select unit DU0.

There is a format utility in diagnostic ZRQBC0. REV A only good for RD51's. REV C works with RD52's. Symptom is INVALID UNIT.

Fabric softener or "Bounce" can be used to reduce static on line printers. Static on printer can cause DMF32 to stall.

Dual-ported RA60's must have both ports set enabled when spinning up drive. Otherwise the other port gives difficulty. Fixed in new microcode coming out in July.

PRO 350 has a ZIF connector to motherboard problem. There is a mandatory (free) FC0. [Involves replacement of entire motherboard! This will also serve as a good test of your local field service technician -- betcha a lot of them will break one of the little handles on the user-insertable FPU chip carrier. -- editor]

DECSA has problems. Server hangs, or terminals hang. Fix is ECO #01297-01 for hardware. Units shipped after 2/25/85 are OK. The ECO is mandatory, and all field units should be upgraded by August 1985. The second fix is terminal software V2.0, also mandatory (free). This was to have been complete by 6/14/84 and available in July.

VT220 paper holder has mounting holes in the wrong place.

BA23-A-AF is the floor stand add-on for the BA23 box.

OKIDATA PACEMARK 2410 PRINTER
Gregory N. Brooks

I have had two Okidata 2410 printers operating for almost six months now and am very pleased to report that they have performed well up to our expectations.

The print quality is very good, they are less noisy than many other printers in their price range, and seem to be very reliable and free of bugs.

One 2410 had a print wire fail just a few hours (of print time) after delivery. However, I had a hassle free warranty repair at the local Xerox Service Center and it has been operating since that time under a heavy load.

The Okidata has only two print modes, data processing quality and correspondence quality. The correspondence quality is not as sharp or as clear as the DEC LA 210, but is more than acceptable for memos, reports, and some letters.

The two biggest gripes I have with the 2410 are the rear feed for paper, and the rather steep cost of the ribbons.

All in all, I have found the 2410 well worth the money, reliable, a good, versatile performer, and a good companion to my PDP 11's. (Soon a good VAX companion.)

Gregory N. Brooks
Washington University

HMS6

DATA TRANSLATION DT 2782 A/D CONVERTER "PGH" OPTION
Gregory N. Brooks

When ordering the PGH (programmable gain) option for the DT 2782, BEWARE!

When the PGH option is added to your 2782, CSR bits 2 and 3 no longer function as "burst mode enable" or "increment mode enable" (respectively), but are now the bits used to program gain.

I found this out through personal experience after having purchased two boards, (six months apart) having used them successfully for several months in interrupt mode, and then trying to implement DMA for the two.

Needless to say I was furious. No one at Data Translation told me I would lose these DMA features, even though I stated DMA capabilities as my main reason for buying.

Data Translation's solution was for me to send the boards back and have the A/D modules changed to models with no PGH at a cost of \$ 475.00 per board. I didn't want to pay, but didn't have a choice. I sent the boards back and they came back without the PGH feature and the DMA bits restored.

When I returned the boards I also sent a letter to Data Translation stating all my problems and that I would be looking at other A/D manufacturers the next time I purchased an A/D unit and a history of my purchases with Data Translation.

There is justice in the world! I received my boards back as stated above, an invoice marked no charge, and a letter apologizing for the mix up. A happy ending to a sad story.

If you are contemplating the purchase of a DT 2782 be sure to ask the sales rep. "Will this option disable any features of this unit?"

It may be a good idea to ask when ordering any of their products or any other manufacturer's products.

Gregory N. Brooks
Washington University

HMS7

KDJ11-?? OR "ALL 11/73'S ARE NOT CREATED EQUAL."
Wayne E. Kesling

If you are considering an upgrade of your PDP-11/23 or PDP-11/23+ to a PDP-11/73, there are a few things that you should know about the seven different KDJ11 boards made by DEC. It must be noted, however, that all seven are not available as board level products to be used for upgrading a system. There are three versions of the dual-width board and four versions of the quad-width board. All are based on the renowned J11 chip set that placed the PDP-11/70 on a board. However, the various boards that the set is placed on are not all the same.

First, the dual-width, KDJ11-A? boards do not have the high speed PMI (Private Memory Interconnect) memory support logic that is present on the quad-width, KDJ11-B? boards. Neither do they have the serial line unit that is on the KDJ11-B? boards. The differences between the various KDJ11-A? boards are in their support of the new FPJ11-AA floating point processor and are explained as follows:

KDJ11-AA Will support the FPJ11 if the board is returned to DEC for an ECO.
KDJ11-AB Has a new etch that supports the FPJ11 but no FPJ11 chip is installed. The chip is available for \$500.00.
KDJ11-AC This is the KDJ11-AB board with the FPJ11 chip installed.

The quad-width, KDJ11-B? boards have a serial line unit that can be used for the system console and arbitration logic for support of PMI (Private Memory Interconnect) memory. As with the KDJ11-A? boards, the major differences in the various KDJ11-B? boards is in their support of the FPJ11-AA floating point processor. There is also a difference in CPU clock speed on one board. The differences are as follows:

KDJ11-BA 18MHZ clock, does not contain an FPJ11 processor chip and, if the etch is below REV E, it is not upgradeable.
KDJ11-BB 18MHZ clock, etch supports the FPJ11, but no chip installed.

KDJ11-BC 15MHZ clock, does not support the FPJ11 and can not be upgraded. At the time of this writing, this is the only quad-width board that is available at the board level.

KDJ11-BF 18MHZ clock, with FPJ11 installed. To not quote an anonymous DEC employee,

"One might expect the KDJ11-BF board to be available at the board level around the third quarter of 1985."

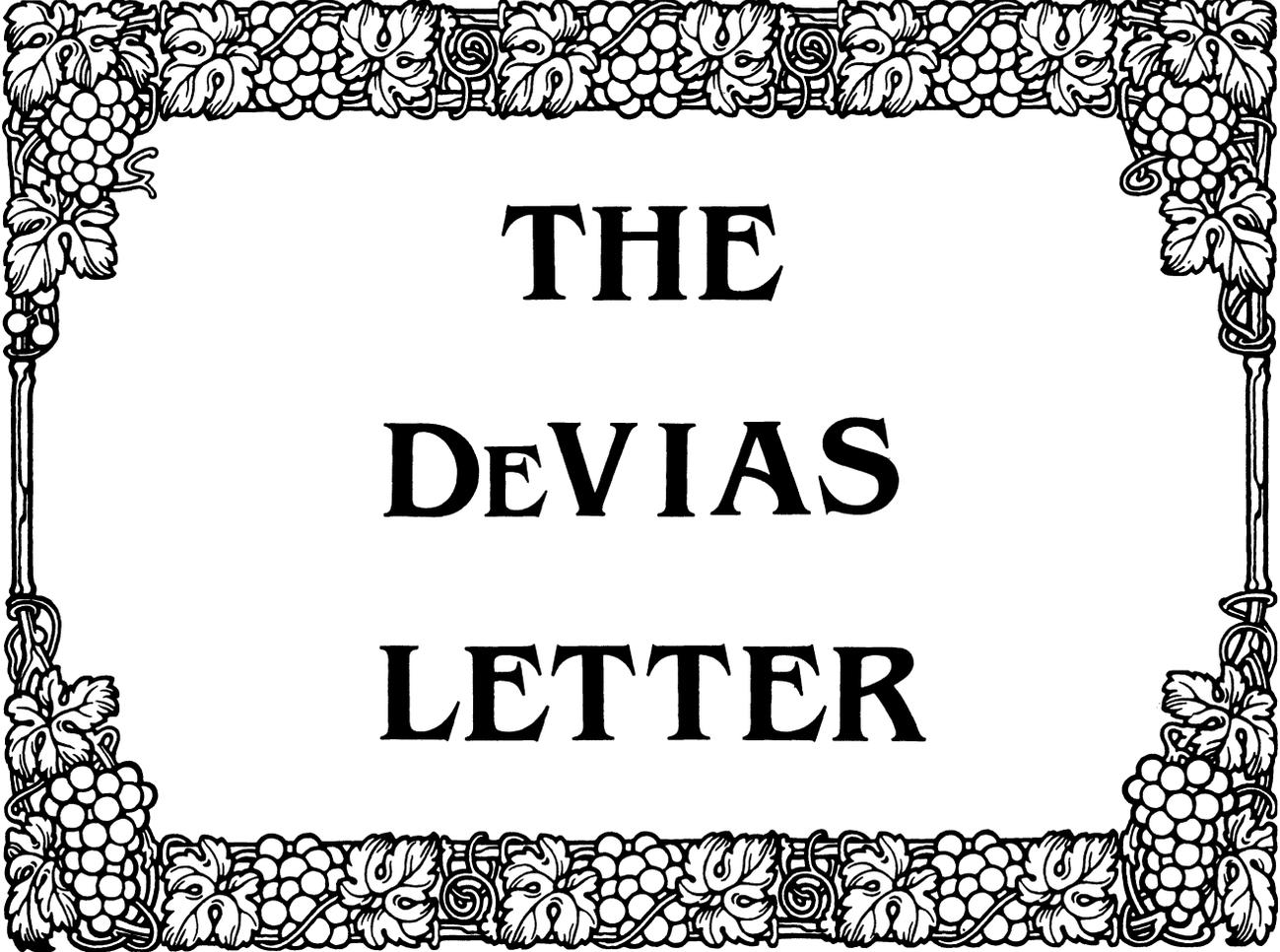
However, I will deny any knowledge of that quotation if confronted with it.

Of possible interest to the readers is that the KDJ11-BC is the board that is installed in a Micro PDP-11/73, which attains 60% of the performance of the PDP-11/70 doing integer operations, and in the PDP-11/84 P, which has PMI memory and attains 90% of the performance of the PDP-11/70 doing integer operations.

This article is only intended to make you aware of the various KDJ11 boards that are in systems and does not contain the necessary information to accomplish an upgrade of your system. That information is available in several micronotes, including old #114 (new uNOTE #003) and old #115 (new uNOTE #004).

Wayne E. Kesling, Sr. Engr.
Monsanto Research Corp.
Miamisburg, Ohio 45342





THE DeVIAS LETTER

I N T H I S I S S U E

The DeVIAS Letter -

Curley's Corner

Letter from the Editor

Letter From Klaus Centemeyer

More V3.2 Notes from Bob Curley

Borger's Browsings - Notes on IAS Version 3.2

Borger's Browsings - Dissassemblers

Using Macros and Assembly Language Hacks
Jim Williams

IAS

Chairman

Bob Curley
University of Pennsylvania
Flourtown, PA

Membership Coordinator

George R. Wells
General Electric Co.
Louisville, KY

Symposium Coordinator

Skip Stanfield
USAF
Washington, DC

WHIMS Commissioner

Kathleen Anderson
Eaton Information Mgmt.
Systems Division
Hampton, VA

Communications Representative

Thomas O'Keefe
United Technologies Mostek
Carrollton, TX

Technical Advisor

Mike Reilly
Digital Equipment Corporation
Maynard, MA

DeVias Newsletter Editor

John Ross Roman
McDonnell-Douglas
Hazelwood, MO

DEC Counterpart

Tim Leisman
Digital Equipment Corporation
Stow, MA

Librarian

Mike Robitaille
Grumman - CTEC, Inc.
McLean, VA

RSX Liaison

Ray French
Boeing Computer Services
Seattle, WA

Library Coordinator

Bob Schultz
INCO Inc.
McLean, VA

Department of Radiation Therapy
University of Pennsylvania
Room 410
133 South 36th Street
Philadelphia, Pennsylvania 19104

25 July 1985

Dear IAS Enthusiast,

This is the newsletter I've been waiting for. The first edition of the combined newsletters of all the SIGs. I feel that it a good thing for DECUS and a good thing for DECUS members, but it is very clear that my opinion is not universally shared. The fact is there is no data to arrive at a convincing decision, there has been no experiment. Now, there will be some data. For two years DECUS has charged money for SIG newsletters that have been published by the SIGs individually (with some small groups). Now, DECUS will do this, publish the "BIG ONE" as it has been called. I would like to hear your opinion.

IAS V3.2 is still running here. It has crashed once - due to a hardware failure of the KY-11 in my 11/70. The software is still solid - as we've come to expect. BUT, there are some problems, especially with the mag tape. It is a serious burden to me an my users. I do not get the feeling of bustling activity at fixing it either.

Please send me (or call - 215-662-3083) with all the little things that bother you about IAS v3.2. Lets get all the stuff out in the open and work at getting them fixed. Good things, too. I've had one report of DECnet troubles and the one in the last issue about Datatrieve. How are the "layered" products running for you? How's your software running?

Does anyone have IAS 3.2 running on an 11/73?

Hot and sticky weather has arrived in Philadelphia. Summer. May the Autumn see all of our problems solved and let there be no discontent this Winter.

Happy Summer,
Bob Curley

Letter from the Editor

As this may be the first time for many of you to experience the DeVIAS Letter, I thought I would take this opportunity to tell you a little about the IAS operating system:

IAS -- The Interactive Application System

IAS is one of the lesser known operating systems in the 16 bit space. However, it has its roots in RSX-11D, once the mainstay operating system on PDP-11s. Besides the 11D real time processing facilities, IAS supports multi-user and time sharing flavors. This versatility makes IAS one of the most useful of PDP-11 operating systems.

IAS supports programming languages such as MACRO-11, FORTRAN, BASIC, COBOL; the Files-11 and RMS file systems, utilities such as PIP, ZAP, DMP, FLX, EDT, EDI, and many others. Although de-emphasized in favor of other products, IAS does support all of the Unibus PDP-11 processors and mass storage devices, including the MSCP devices such as the RC25 and RA series disk drives.

IAS allows concurrent interactive, batch, and real-time processing. Real-time processing is based entirely on priority. A real-time IAS system provides excellent facilities for interrupt processing in an I-O driven application, for example. Interactive processing is managed by a heuristic time-slicing algorithm when real-time activities do not take precedence. Batch processing can operate in "soak" mode, which only uses processing cycles not utilized by real-time or interactive processing, or the system manager can specify the percentage of time available to batch.

The two flavors of interactive processing are multiuser and timesharing, usually identified by their CLIs, MCR for interactive and PDS for timesharing. Although usually not run together, the two flavors are not mutually exclusive, as both can run at the same time on different terminals on a system. The interactive system is more "open" and less protected. Users have more freedom, and are expected to be more knowledgeable. Consequently, they have more privileges.

Timesharing, in contrast, is a more protected and "user friendly" system. User privileges are granted by a system manager, resources are allocated and accounted. Its use is typified by a college timesharing operation, while the multiuser flavor may be used by an OEM product engineering department.

The IAS Special Interest Group

This is the interest group for RSX-11D and IAS. We are involved in real time and timesharing, systems programming and applications, assembly language and high level programming and security and tools. IAS is used in a broad spectrum of applications from communications systems that accent the real time responsiveness to a medical database that accents the terminal responsiveness of this versatile operating system. The SIG sponsors sessions at the DECUS U.S. symposia, publishes this newsletter, and has close ties with the IAS engineering group within Digital. Although IAS has suffered from corporate neglect early in this decade, it has recently been rejuvenated with Release 3.2, with further releases expected.

The DeVIAS Letter needs contributions in order to continue as an effective medium for exchange of information regarding IAS. All contributions should be either Runoff source or camera ready copy, e.g. sharp black type on 8.5 by 11 inch paper with 1 inch margins.

Please send all contributions to:

John Roman
McDonnell-Douglas Corporation - Dept N436
600 McDonnell Blvd.
Hazelwood, Missouri 63042
(314) 234-0984

Klaus Centmayer
 Techn. University Muenchen
 Inst. f. Datenverarbeitung
 Franz-Josef-Str. 38
 D 8000 Muenchen 40

John Roman
 McDonnell Douglas Corp. - Dept.N436
 600 McDonnell Blvd.
 Hazelwood, Missouri 63042
 USA

Munic, 11-Jul-85

Dear John,

In April I have send You a copy of my letter concerning my first impressions about the new IAS V3.2, which I sent to the european IAS-users. In the meantime only a few of the problems have been solved, but further errors have appeared. Enclosed you find some further SPR's.

Concerning the mentioned problems, here the SPR-responses:
 PIP with MT : will be corrected in Update B (is not available !).
 SPR2 : this is a problem of TKB , solution see SW dispatch APR-85 no. 5.3.1.1.

PRE MT-MT copy : no answer up to now, the patch (5.1.17 APR-82) produces checksum error, but works.

My problems with EDT (V3) and the TT-handler , they cannot reproduce them.

Kermit does still not work, because of the missing RMS V2.

I have problems with the DECUS C-compiler (11-SP-18), perhaps somebody uses this software with IAS and can help me.

Sincerely,

K. Centmayer

Attachments

IAS4

digital

SOFTWARE
 PERFORMANCE
 REPORT

FIELD NO.:

CORPORATE SPR NO.:

990883

✓ TO SET UP FOR PROPER ALIGNMENT, START AT MARK BELOW.

PAGE _____ OF _____

OPERATING SYSTEM IAS	VERSION 3.2	SYSTEM PROGRAM OR DOCUMENT TITLE EDT	VERSION OR DOCUMENT PART NO. V03.15	DATE 10-JUL-85
NAME: K. Centmayer FIRM: Inst. f. Datenverarbeitung ADDRESS: Technische Universität München CUST. NO.: Franz-Josef-Str.38, 8000 München 40		DEC. OFFICE AND CONTACT PERSON Munich		DO YOU HAVE SOURCE? YES <input type="checkbox"/> NO <input checked="" type="checkbox"/>
SUBMITTED BY: K. Centmayer		PHONE: 089-349451		CAN THE PROBLEM BE REPRODUCED AT WILL? YES <input checked="" type="checkbox"/> NO <input type="checkbox"/>
ATTACHMENTS MAG TAPE <input type="checkbox"/> FLOPPY DISK <input type="checkbox"/> LISTING <input type="checkbox"/> DECTAPE <input type="checkbox"/>		OTHER: CPU TYPE SERIAL NO. MEMORY SIZE DISTRIBUTION MEDIUM SYSTEM DEVICE DO NOT PUBLISH 11/45 820 124k MT 9 RP 03		
OTHER:		COULD THIS SPR HAVE BEEN PREVENTED BY BETTER OR MORE DOCUMENTATION? PLEASE EXPLAIN IN PROVIDED SPACE BELOW.		

Subject: SPR Number 11-75725
 Additional information

Problem:
 EDT includes sporadic CR LF.

In screen Mode sporadic CR LF characters are included on the screen.
 The edited file is o.k., these characters appear only on the terminal.

Solution:
 The problem is in the generation of the TT-handler, EDT V3 needs Escape-sequence-support (EDI V2 did not).
 The TT-handler must be generated with E\$\$SEQ non-zero.
 With E\$\$SEQ = 4 only VT100 works , VT52 does not work at all, probably for VT52 E\$\$SEQ = 3 is correct.
 Please provide necessary documentation.

ALL SUBMISSIONS BECOME THE PROPERTY OF DIGITAL EQUIPMENT CORPORATION.					
SHORT NAME	MNT. CAT.	MNT. GRP.	XFER GRP.	FL	PRB. TYPE
DATE RECEIVED (MAIL)	DATE TO MAINTAINER		XFER DATE	LOGGED ON	
DATE RECEIVED (ASG)	DATE RECEIVED FROM MAINTAINER		DATE ANSWERED	LOGGED OFF	

EN-01044-07.REVI (35C)

IAS-5

ADMINISTRATIVE SERVICES GROUP SWS

IAS 3.2 EDT V2.0 10-JUN-85

EDT is terminated with memory protect violation.

The following comand sequence:

Øset search exact

find 'any string'

terminates EDT with 'memory protect violation and with pending I/O request'
at PC 61412.

Note: This is EDT V2.0 (from IAS V3.1) new linked in IAS V3.2.

This version of EDT is used because of errors in V3.015 (see SPR).

SPR IAS V3.2 No. 9:

IAS 3.2 DEM V3.201 10-JUN-85

DEM does not start with actual memory display

DEM starts with wrong default values (Base = 64K , size = 320K) and
not with the actual memory size (EXTEND ALL) as the old version did.

Please provide information to change these default values.

Please provide information about the displayed system 'Uptime',
this time starts not with zero but with the time, when the last SAVE
was done.

IAS-6

IAS 3.2 FLX 16.03 30-MAY-85

FLX does not initialize DY:

The new Version of FLX can not initialize a RT-11 format Flexible disk.

The command DY:/RT/ZE produces the error message:

FLX - Invalid device

Reading or writing on a initialized DY: (RT 11 format) is possible.

Temporary solution:

Using FLX (M11) from IAS V3.1 for initializing.

SPR IAS V3.2 No. 7:

IAS 3.2 RMS V2 10-JUN-85

The distributed RMS is not Version 2.

The RMS - kit in IAS V3.2 consists only of old versions.

The files RMSMAC.MLB and RMSLIB.OLB (last insert 18-MAY-79 or 20-JUN-79)
are identical with older versions and really not RMS-11 V2.0 as announced
for IAS V3.2 (Software Dispatch Mar 1984).

There should be new macros, they are necessary for existing programs (Kermit).

IAS-7

MORE V3.2 NOTES
Bob Curley

The following are notes, observations, rumors and SPRs related to V3.2 of IAS.

1. Bob Schuldt says that Mert Crockett found that IE.TMO now has a value of -95. It used to be -80. in V3.1 of IAS. This is a great example of what can happen to you bit twiddlers who hard code constants into your perfect code.

2. IE.TMM is not present in V3.2. DBMS-11 uses it and it comes up missing when we built it under V3.2. On V3.1 it has the value -71. and meant: "too many outstanding messages"!

3. The /WIDTH:n switch of the PDS DIR command does not work for widths other than 80.

4. After mounting a tape 5 to 8 times, the MOUNT command fails on any attempt to mount further tapes. For example:

```
PDS> MOUNT/FOR/DENS:1600 MM0: POKEY
MOUNT -- Device not ready
Volume not mounted
```

The tape is, in fact, loaded and ready. This happens more predictably when you log out with a tape mounted.

5. The /FILE_PROTECTION switch for the PDS MOUNT command fails to work when mounting a tape. "Syntax Error" is printed and the tape is not mounted.

6. SORT-11 does not properly sort variable length records. Records which are short have one character from the longer records appended to them.

7. We have a DEC WS211 Word Processing System connected to three lines on a DH on an IAS system. The WPS system uses "CX mode" to converse with the IAS machine. The users who use the IAS programs via CX mode on a WPS terminal frequently logoff IAS and before IAS sends its LOGOUT message, the users type "r" to exit CX mode on the WPS system. The WPS system, with no place to send the logoff message from IAS, sends IAS an XOFF. IAS does not finish the logout. PDS appears locked that way until a reboot is done or until someone else selects CX and the same HOST line on the WPS system. Immediately they get the waiting logout message.

8. When I format an RK05 it takes 7 minutes and 19 seconds before the verification pass starts. I used to key in a short program that did it in 26 seconds. Using that arithmetic (i.e.

that it takes IAS 15 times longer than optimal to format an RK05) it might be possible to format an RA81 in amazing time.

9. The MOU command fails with the error:

```
MOUNT -- Parity error on device
when mounting a tape using the command
MCR>MOU/DENS=1600/FOR MM0:X
```

10. Under timesharing, the PDS command

```
MOUNT/PROT:(W:RWED) MM0: GUMBY
fails with
SEGMENT FAULT
```

11. Under timesharing, the PDS command TYPE abouts with a SEGMENT FAULT when TYPING a file containing records longer than 512 bytes.

```
12. PDS> SET PROT 200*.DIR (GR:RWED)
gets me
200*.dir - illegal file specification
when
PDS> MCR PIP [0,0]200*.DIR/PR/GR:RWED
works!
```

13. FORTRAN 77 (V4.1) - the SECNDS function does not accurately span midnight.

14. We can not copy files from tape to disk when the tape is mounted with read access.

15. It is now possible to say SET SCI from any PDS terminal and achieve most of the functionality of the console terminal. However, the HELP command does not recognise that the terminal prompt is now SCI.

```
16. Under PDS
MOUNT/FOR MM0:X
gets
MM0: - Illegal Device
and
MOUNT/FOR MM0: X
works.
```

17. The Update B kit includes 10 binders in what appears to be "official orange." The Edit Utilities binder can not contain the material which includes the new EDT manual. In addition, the V3.2 distribution kit includes binder spine inserts for 11 manuals. Thus, the current set requires 12 binders, not the 10 sent out.

18. The PDS COPY command fails to copy files to tape. Or, for that matter, read files from tape.

19. Again, my overall impression is that V3.2 is a good product with some quality control problems and one bad problem - the mag-tape manipulation. We've had one crash, the KY-11 on the 11/70 failed. It stays up and keeps running.

Borger's browsings - Notes on IAS Version 3.2

It's been a fun time, but I finally have version 3.2 up with all the patches we have done at Reese, plus a few additional changes. For your enjoyment, education, or guide I will list the major problems we encountered trying to get the system up.

1. First thing that happened was after trying to do a 2 RK system, (does anyone out there still have them?) we overflowed RK1. Also tape ran off end of reel. Had to boot MT a second time. These are known problems, but they are hard to remember when you do major system gen's 4 years apart.
2. First attempt at SYSGEN phase 1 terminated with a

SN1 -- *FATAL*-Open failure on file SY0:[11,17]IAS.SAV.

Problem was not enough contiguous blocks on disk. Had to change size of save system in phase 1 command file from 96K to 60K to make IAS.SAV file smaller. After this change, phase 1 and 2 went ok.
3. Now had operating system on RK, tried to use BRU from existing 3.1 system to copy all files from distribution mag-tape to new version 3.2 RM03 system disk. BOO HISS, they don't tell you, but version 3.2 has new version of BRU and this is the only one that reads the distribution tape correctly. At least the new BRU can handle all variations of BRU tapes. They could have told us that we had to use new BRU.
4. After this, we tried to gen a new system on a large disk. (In our case 2 winnies configured as 4 RM03's.) Since DR0 was our running system disk, I was trying to do sysgen to DR3:. Tried everything, but could not get sysgen phase 2 to come up. After much head scratching, crash dumping, etc. we finally figured out that we had to specify "SY=DR3" in phase 1. Although it has been years since I did it, I looked back at my records, and version 3.1 of IAS was smart enough to redirect SY to the correct unit number in the boot block when you did a "BOO DRn:WB". (At least our command files with SY redirected to DK0 worked.) SAV is smart enough to look at the boot block of the disk and redirect SY: to the correct disk while he is running. Evidently BOO is the one that forgot.
5. Once we got a single TT handler going on our big disks, we then rebuilt the multiple teletype handler. Surprise, release notes say that autoloader vector PSECT is now generated with read-only characteristics, and may prevent some tasks from building due to "NOT ENOUGH APR'S TO MAP TASK" error. They didn't tell us that TT handlers would bomb, and that their solution, (using the /RW switch on the task image name,) doesn't work well for poor TT. Our final solution was to edit ESQTBL.MAC to change .PSECT ESQSTT from RO to RW, thus reducing the size of the RO

part of TT to back under 4K. It turned out to be a real balancing act to keep the R0 portion of the root segment under 4K and at the same time keep the RW root segment plus segment IDLE totalling out at under 4K.

6. Finally got TT rebuilt, but some people will not like the fact that system will support fewer terminals, (due to added code to support new bells and whistles.) System generation phase 2 came up and TT terminated immediately with the message:

```
TT... **FATAL** ILLEGAL DC11 SPEED TYPE -- interface 1
```

Unfortunately system generation phase 2 only checked to see that TT started up. Phase 2 went gleefully on installing tasks in the new system. Once we tried to boot our initial single TT handler system on that disk we found it was unusable due to all those tasks, (including the system disk handler,) not being properly re-installed. After much crash dumping, task examination, etc. finally found the offending piece of code in INIRTN.MAC. The line:

```
SIINDC: CMPB I.LHUB,#7 ; SPEED RANGE LEGAL?
```

was trying to use I.LHUB as in index into the interface table for the DC11 interface currently being initialized, but the programmer left out the (R5). After changing the line to read:

```
SIINDC: CMPB I.LHUB(R5),#7 ; SPEED RANGE LEGAL?
```

things worked much better. Interestingly enough, that bad code worked fine for many years. What code was really comparing to 7 was upper byte of N.OVPT in low memory pointers, (the Overlay Run Time system work area pointer.)

7. As an aside, we discovered an interesting bug in TKB concerning overlays and using ZAP to examine and/or patch task images. The problem is that task builder prints out the starting block of each segment of an overlaid task:

```
Disk blk limits: ssssss eeeee oooooo dddddd.
```

If you wish to patch an overlaid task you evoke ZAP with the /LE switch to get a listing of the starting block offsets for the various segments and they match the same data from the task map.

The problem occurs when trying to examine or change a Read-Only memory resident overlay and is caused by the fact that all other overlays start on even block boundaries, (cause handlers can only start reading at the start of a block,) but the first R0 memory resident overlays starts at the next 200 byte boundary after the end of the tasks pure root, with subsequent R0 memory resident overlays starting at the next 200 byte boundary. The end result is that the disk blk limits line is totally wrong for a memory resident overlay, (and ZAP is wrong with his

list.) The only way to figure out the correct address is to:

1. Evoke ZAP once using the "/LE" switch to list task segments. You will get the standard segment list which looks something like this:

```
3: 000000-003437      (tasks read-write root)
.                    (other
.                    overlays)
46: 040000-055043     (tasks read-only resident overlay)
30: 060000-075277     (tasks read-only root)
```

2. Take the size of the read-only root, (75277-60000+1 or 15300,) round it up to the nearest 200 byte boundary, (15400,) and convert this to a block and byte offset, (block 15, byte 400.) Add this to the starting block of the task's read-only root, (30,) to produce the actual starting location of the task's read-only resident overlay, (block 45, byte 400.)
3. Re-enter ZAP in absolute mode and set your offset to the start of the read-only resident overlay via the command "45:400;1R" You are now offset properly to the start of the first read-only overlay.
4. If multiple read-only overlays are present, subsequent calculations similar to the one done to calculate the offset due to the task's pure read-only root must be done.
8. Finally had a working system, started to use version 3.1 command startup files, found out that any TER commands to change terminal characteristics tended to bomb TT handlers in mysterious ways. Very confusing, bombs would not always be repeatable, changing order of 2 lines in command file, doing multiple characteristic changes, all kind of things would cause TT handler to lock up. Finally found out that this one was our fault, (or at least partially.) Long ago we had defined another couple of terminal characteristics, UC0 thru UC9, which we used for our own use. Problem was that DEC added some more characteristics and with our additions the size of a single terminals characteristics area grew from 14 to 16 octal bytes. TT routine SAVDF tries to save current characteristics, saving them two to each internal TT node. With 16 bytes +2 for the Terminal Table Address for each saved characteristics area, plus 2 bytes for the first node word which was a link to the next node, size was 42 bytes, two larger than size of TT internal nodes. Solution was to drop off a couple of the UCn characteristics which we never used, which got us down to a 14byte characteristics area. If you add characteristics, check the global T.CSZW. If its 14 you are ok, if it is 16 you have to change the size of TT internal nodes to be a minimum of 2*T.CSZW+2, (in NODES.MAC.)

9. Finally had a system going, went to install one of our favorite system patches that lets non full-blown timesharing systems have a default device, discovered ..FDEV module no longer in SCOMM. Surprise, many SCOMM routines now in SCMSBS routine in exec proper, with only entry points in SCOMM itself. To make room for this and other changes, the Exec now has three resident overlays rather than two, new exec overlay has QIO routines and crash module, routines moved from SCOMM to EXEC include node handling, event logging routines, IO request queueing, page descriptor swapping, along with FSTD, REDT, FDEV, GIAS, PIAS, GIAB, PIAB, and MUFL. Had to rewrite our code that overwrote code in SCOMM to overwrite code in Exec instead, things became harder.

That's about it. System seems to be working ok now. I will close with a listing of a couple of simple Reese basic programs that made my life easier these last few weeks.

The first just produces listings of .STB files, its operation is self evident from looking at the code. The second is quite useful in determining which of the tasks on a disk need of being re-built for version 3.2 because they reference a library or because they are privileged. To use this program you must:

1. Do a PIP TASK.DIR=[*,*]*.TSK;*/LI
2. Edit TASK.DIR with TECO to change it from imbedded carriage control to implied carriage control.
MCR>TEC TASK.DIR
*EX\$\$
3. Execute the enclosed Reese Basic Program which checks the task header for any task in TASK.DIR created before the date of the files on version 3.2 release, and reports any task needing re-building.

Hope things go up easier for you.

```

100 ! PROGRAM TO FIND PRIVILEGED TASKS
110 DIM NA$(20JV),UI$(9JV),IN$(80JV),DT$(9)
120 DIM #4,DD%(1,256)
121 B7%=128
125 LOAD "LB:C1,202JANDOR"
130 OPEN #3, "TASKS.DIR/RO"
140 IF END #3 THEN 500
145 XD=DCEN("10-JAN-85")
150 INPUT LINE #3 IN$
160 IF POS(IN$,"[")< 1 THEN 200
170 PRINT IN$
180 UI$=SEG$(IN$,POS(IN$,"["),POS(IN$,"J"))
200 IF POS(IN$,";")>1 THEN 220
210 GOTO 150
220 NA$=SEG$(IN$,1,POS(IN$,";")-1) : DT$=SBS$(IN$,32,9)
230 IF DCEN(DT$) >= XD THEN 150
299 ON ERROR GOTO 390
300 OPEN #4,UI$+NA$+"/BL/RO"
310 B%=DD%(0,4)
315 CALL "AND"(B%,B7%,C%)
318 IF C%<1 THEN 330
320 PRINT " ",NA$;" FLAG WORD = ";OCT$(B%)
330 L1%=DD%(0,13) : L2%=DD%(0,14)
340 IF L1%=0 THEN 360
350 PRINT " ",NA$;" LIBRARY = ";R5A$(L1%);R5A$(L2%)
360 CLOSE 4
370 GOTO 150
390 PRINT " ";UI$;NA$;" IS GONE"
400 GOTO 150
500 CLOSE
510 EXIT

```

```

10 ! PROGRAM TO LIST CONTENTS OF AN STB FILE
20 DIM A$(122),TY$(14)(8),NA$(15),XX$(6)
25 TY$(1)="MODULE NAME"
30 TY$(2)="C-SECT NAME"
40 TY$(3)="SYMBOL NAME"
50 TY$(4)="TRANSFER ADDR."
60 TY$(5)="GLOBAL SYMBOL"
70 TY$(6)="P-SECT NAME"
80 TY$(7)="VERSION ID"
90 TY$(8)="MAPPED ARRAY"
100 INPUT "NAME (ONLY) OF STB FILE ";NA$
110 OPEN #3, NA$+".STB/RO/LN:122"
120 OPEN #4, NA$+".DOC/WR/LN:132"
130 PRINT #4 : PRINT #4
140 PRINT #4, " ", "LISTING OF ";NA$; ".STB", DAT$( ), TIM$( )
150 PRINT #4
160 PRINT #4, " ", "NAME", " VALUE", " TYPE", " FLAGS"
170 PRINT #4, " ", "-----", "-----", "-----", "-----"
180 PRINT #4
190 NL=0
200 INPUT LINE #3,A$
210 IF LEN(A$)<10 THEN 1000 : ! QUIT IF END OF GSD
220 LC=(LEN(A$)-2)/8
230 FOR I=1 TO LC
240 X1=ASC(SBS$(A$(I*8-5),1)) : IF X1<0 THEN LET X1=X1+256
250 X2=ASC(SBS$(A$(I*8-4),1))
260 X3=ASC(SBS$(A$(I*8-3),1)) : IF X3<0 THEN LET X3=X3+256
270 X4=ASC(SBS$(A$(I*8-2),1))
280 PRINT #4, " ", R5A$(X1+256*X2);R5A$(X3+256*X4),
290 X1=ASC(SBS$(A$(I*8+1),1)) : IF X1<0 THEN LET X1=X1+256
300 X2=ASC(SBS$(A$(I*8+2),1))
305 XX$=OCT$(X1+256*X2) : XL=LEN(XX$)
310 IF XL=6 THEN PRINT #4, " ";XX$,
315 IF XL<6 THEN PRINT #4, " ";STRING$( "0",6-XL);XX$,
320 TY=ASC(SBS$(A$(I*8),1)) : IF TY<0 THEN LET TY=TY+256
330 FL=ASC(SBS$(A$(I*8-1),1)) : IF FL<0 THEN LET FL=FL+256
350 PRINT #4, TY$(TY+1);" ";
355 XX$=OCT$(FL) : XL=LEN(XX$)
360 IF XL=3 THEN PRINT #4, " ";XX$,
365 IF XL<3 THEN PRINT #4, " ";STRING$( "0",3-XL);XX$,
370 NL=NL+1
380 IF NL<55 THEN 470
400 PRINT #4, CHR$(12) : PRINT #4 : PRINT #4
410 PRINT #4, " ", "LISTING OF ";NA$; ".STB", DAT$( ), TIM$( )
420 PRINT #4
430 PRINT #4, " ", "NAME", "VALUE", " TYPE", " FLAGS"
440 PRINT #4, " ", "-----", "-----", "-----", "-----"
450 PRINT #4
460 NL=0
470 NEXT I
480 GOTO 200
1000 EXIT

```

Borger's browsings - Disassemblers

As has happened on each previous release of IAS, I am now busy updating all of the patches we made to IAS to match version 3.2. We have tailored some parts of the system for our use, and every new release we find we have to re-do some of the changes we made to IAS. And when we have to do that, we have to deal with the very real problem of not having sources.

Although we have a source license, and version 3.1 sources on fiche, one has to re-enter the source code from the fiche. For version 3.2 I found out that source will only be distributed on machine readable media, and fiche will not be available. So how does one go about finding out why your system is crashing in the middle of exec routine XXXXXX, or how do you modify MCR for your own special use? You use the magic tool of the IAS code cracker, the disassembler.

Already I hear someone out there saying, "What's a disassembler?". So let me explain. A disassembler is a program that takes DEC object files and converts them back to macro source or listing files. Although I have heard terms such as "weird" applied to disassemblers, (or programmers who use them,) once you get over the basic idea of working backwards, it's just another program. Since DEC does pass out object modules for almost everything, one can get pseudo-source code, (minus all comments,) for any part of the system. Although the loss of comments may seem like a problem it really is not. IAS code, (especially Exec or DCL routines,) are so filled with Global symbols that the code is almost self documenting. A sample run follows:

```

MCR>orc ti:=mp60
; MP60 OBJECT TO MACRO CONVERSION V01.1 17-JUN-85 17:20

; .TITLE MP60
; .IDENT /V03.0 /
;
; .PSECT ZZPRTY,I,LCL,RW,CON
;
; .ENABL LSB
PAR60::
BIS #3,CR.70
TST PER.70
BPL 24$
BIT #PRI7,2(SP)
BNE 244$
24$: BIC #PRI6,PS.EXP
MOV R1,-(SP)
MOV R2,-(SP)
MOV R3,-(SP)
MOV R4,-(SP)

```

```

JSR    PC, .ERPNI
MOV    R4, R1
MOV    (SP)+, R4
BCS    144$
MOV    6(SP), R2
MOV    10(SP), R3
JSR    PC, .MPNOD
MOV    #600, 4(R1)
MOV    #PARRG0, R2
102$:  CLN
      TST    (R2)
      BR    112$
      JMP    (R4)
112$:  BMI    130$
      ADD    #2, R2
      CMP    R2, #172140
      BNE    102$
      BR    140$
130$:  MOV    R2, 36(R1)
      MOV    (R2), 40(R1)
140$:  JSR    PC, .ERLNC
144$:  MOV    PER.70, R1
      BPL    202$
      MOV    (SP)+, R3
      MOV    (SP)+, R2
      MOV    (SP)+, R1
      MOV    #SC.PAR, -(SP)
      CLR    PER.70
      BIC    #3, CR.70
      JMP    SSTCOM
202$:  INC    .CACER+2
      CMP    .CACHE, .CACER+2
      BHI    224$
      BIS    #14, CR.70
224$:  MOV    (SP)+, R3
      MOV    (SP)+, R2
      MOV    (SP)+, R1
      BIC    #3, CR.70
      JMP    EXINTX
244$:  JMP    .MPCRS
      RTS    PC
      .DSABL  LSB
      .END

```

The above sample was picked because it was a very clean example of a disassembler working at its best. Disassemblers do well with code and with data areas set up with .WORD and .BLKW type directives. They do not do as well with things like ASCII messages, Directive parameter blocks, etc. As an example, the following macro source:

```

.ASCIIZ /This is a test./
.EVEN

```

When assembled and then disassembled, comes out as.

```

BIC    -(R1), (R4)+
BIS    (R5), (R1)
BIC    -(R4), -(R0)
CMP    R1, (R3)+
CMP    R1, R1
BIC    (R5)+, (R4)+
BIS    (R1)+, (R3)+
.WORD  56

```

Although the above code will re-assemble into a series of bytes that contain the same ascii message, it does not really help one to understand the operation of the code. The different disassemblers available have various capabilities to deal with or display ascii or rad50 data as opposed to actual code. Once you have gotten used to disassembly code, you will learn to spot ascii and rad50 data, FDB's, etc. rather quickly.

It is often also interesting to see some of the tricks DEC uses and some of the strange and wonderful macro routines. PDX in particular tends to strain disassemblers to their utmost, and they will still blow up on some modules. I still think the best one was a quick DEC method for setting two condition codes with one operation. As a learning programmer I would have done the following:

```

SEC            ;Set carry bit
SEV            ;Set overflow bit

```

but DEC did it in one instruction:

```

SEC!SEV        ;I didn't know you could do that!!!

```

Any deep discussion of disassemblers is beyond the scope of an article, all I want to do is to get you interested in using disassemblers. There are 3 different disassemblers floating around on sig tapes or from DECUS, and they each have their strengths and weaknesses. I use them all and each has cases where it does the best job. A discussion of the relative merits of each follows.

DISOBJ

This was the first disassembler, written by Larry Simpson when he was with our institution in the early 70's. As an early version from which others perhaps learned, it had a couple of serious limitations but some good points.

1. DISOBJ produced an output which looked like a macro listing file. The octal code or data was presented as one to three words followed by the dis-assembled code. Ascii and rad50 conversions were listed in the comment field. This output format I still find best in cases where you are trying to figure out just what is happening when the program tries to decode a directive parame-

ter block or ascii text.

2. DISOBJ is a one pass disassembler. It does not have the ability to automatically create local or global labels. You have to go thru the source and edit them in.
3. DISOBJ could extract a module from an object file and disassemble it on the fly, without having to extract the module first using LBR.
4. DISOBJ output must be edited to take out the first four columns, (address and 1 to 3 words of instruction,) to be used as source for Macro.
5. DISOBJ tends to terminate abruptly in cases where it has trouble figuring out what is wrong.
6. DISOBJ is very primitive in its error reporting, only issuing an error number.

ORCAM

ORCAM, (macro spelled backwards,) appeared from D J Dunstan in the late 70's. ORCAM had a couple of strong points.

1. ORCAM generated direct macro source code.
2. ORCAM was a three pass program. As a result it handled complex relocation code, etc much better, and was able to put labels in the generated source code, making the programmer's life much easier.
3. ORCAM knew the difference between DATA and CODE psect declarations. If one had code coded well, ORCAM would start disassembling object into .BYTE and .WORD rather than instructions.
4. If one examined a first pass and decided that a certain area was really data instead of code, (or vice versa,) you could over-ride ORCAM's disassembly mode.
5. ORCAM can not extract modules from object libraries.
6. ORCAM was a real improvement, its main limitation was that at times one would have liked to see the actual octal numbers to try to decode ascii or rad-50 data.
7. ORCAM also tends to crash trying to disassemble things like modules from PDX.

DOB

This disassembler is the latest one I know of. It was written by Tom

Getzinger of Hughes Aircraft, with some minor changes by Harry Herman. It has the following features:

1. It seems to correctly handle any of the object module formats that the previous two disassemblers blew up on.
2. It has a good system for identifying local symbols across .PSECT boundaries.
3. It has a /LB switch to automatically extract the object from a library file.
4. It has three switches, (/FP, /FI and /EI) to turn off the disassembly of FPP, FIS and EIS instruction, and creating .WORD nnnnnn output.
5. It has two switches (/RA and /AS) which cause output to contain Rad50 and/or Ascii dumps of the code in the comment field.
6. DOB does not (nor can it) tell the difference between DATA and CODE PSECTS.

There may be more than these floating around, I don't know. They all have their good and bad points, I have at times found cases where only one of the three did what I wanted, (gave me the information about what was happening and did not blow up half way thru the disassembly.) If you really want to start playing with disassemblers, (you have to like solving puzzles and probably loved Adventure if you do,) here is where to get the latest (as far as I know,) versions of the programs

Michael Reese DISOBJ	[300,23] Spring 81 reese version [371,23] Spring 81 enhanced version
ORCAM	[312,315] Fall 83 (also DECUS)
Hughes DOB	[351,30] Fall 83

WILLIAMS'S HACK BOOK

USING MACROS
AND
ASSEMBLY LANGUAGE
HACKS

May 1983 1st edition
Sep 1983 2nd edition
May 1985 3rd edition

(Williams's Hack Book)

Including things that work
and pitfalls to avoid

by

Jim Williams
Principal Software Engineer
RT-11 Development Group
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. Jim Williams, DECUS, and Digital Equipment Corporation assume no responsibility for any errors that may appear in this document.

Some care was taken in the compilation of this document and any suggestions, corrections, or additions would be appreciated. Please send them to Jim Williams, MLO 5-5/E76, Digital Equipment Corporation, Maynard, MA 01754.

DEC, DECUS, PDP, PDT, UNIBUS, RT-11 are trademarks of Digital Equipment Corporation.

Even though it is not mentioned elsewhere in this document, UNIX is a trademark of Bell Laboratories. [The previous, it seems, must appear in each and every computer related document, serious or not.]

This document was produced using RT-11, KED, DECUS RUNOFF, SPX, SPOOL, LS, and PIP.

CHAPTER 1

Introduction

This document is organized into the following parts:

1. How to define and use macros (hack techniques).
2. Hacks that work.
3. Hacks and other things that don't work (pitfalls).
4. Processor difference list.
5. DefInd macro definitions.

The information contained in this document has been gathered from many sources. Among the sources are the components of RT-11 V5, KED, a previous internal list of hacks from Joe Worrall (another Software Engineer in our group), and the fertile (fevered?) imagination of the author. No references are made to the sources for each item for several reasons, 1) to protect the innocent, 2) to protect the guilty, and 3) to be fair since the origin of many of these is hidden in the dim dark past.

CHAPTER 2

Using and defining macros

Macros are a sort of shorthand. They are used to make assembly language programming simpler and to improve readability and maintainability. They can perform these functions in several ways:

1. By providing a single "command" that will generate multiple commands. (e.g. AND)
2. By providing alternative mnemonics for operations which are more descriptive of the programmer's intent than the "raw" assembly language mnemonics. (e.g. BON)
3. By providing additional checking, over and above the checking done by the assembler normally, for a program. (e.g. .ASSUME)
4. By providing an "interface-point" which allows the interface between separate modules to change. The "unchanged" module is then changed by simply redefining the macro and reassembling, rather than searching for all the interface references and modifying them manually. (e.g. .DrDef)
5. By providing for the "one-time" coding and debugging of a complex or obscure function, which then may be invoked several time. (e.g. DfSubr)
6. Retrofitting a module to a processor that lacks an assumed instruction. (e.g. SOB)

WILLIAMS'S HACK BOOK
Using and defining macros

The following macro defines an "AND" instruction since the bit operations on the PDP11 are COM which is "NOT", BIS which is "OR", and BIC which is not "AND" (it is [NOT <src>] AND <dst>). Most PDP11 processors also implement XOR which is "EXCLUSIVE OR".

```
.Macro And      Src,Dst
  Mov          Src,-(SP)
  Com         @SP
  Bic         (SP)+,Dst
.EndM
```

If you were working with lots of logic problems you might want to define the following macros, simply for better mnemonics:

```
.Macro Or      Src,Dst
  Bis         Src,Dst
.EndM
```

```
.Macro Not     Dst
  Com        Dst
.EndM
```

AndB, OrB and NotB could also be defined just by appending "B"s in the proper places:

```
.Macro AndB    Src,Dst
  MovB       Src,-(SP)
  Com       @SP
  BicB      (SP)+,Dst      ; note word operation
.EndM
```

```
.Macro OrB     Src,Dst
  BisB      Src,Dst
.EndM
```

```
.Macro NotB    Dst
  ComB     Dst
.EndM
```

WILLIAMS'S HACK BOOK
Using and defining macros

Macros can be used to define "better" mnemonics for some of the branch instructions. The branch instructions that are used to alter control after testing a flag with BIT(B), or TST(B) since BNE means "on" and BEQ means "off". The following macros define BON and BOFF to make this clearer:

```
.Macro Bon     Dst
  Bne         Dst
.EndM
```

```
.Macro Boff    Dst
  Beq         Dst
.EndM
```

These are used in the following manner:

```
Tst      Flag
Bon      FlagOn

Bit      #Image,Flags
Boff     NoImag
```

This is the .ASSUME macro which is used to "test" assumptions at assembly time. The macro is defined as follows:

```
.Macro .Assume A,Rel,C,Message
.If Rel <A>-<C>
.IfF
.If B <Message>
.Error;"A Rel C" is not true;
.IfF
.Error Message
.EndC
.EndC
.EndM
```


WILLIAMS'S HACK BOOK
Using and defining macros

The following macro scans a string for "*" and counts the number of characters in front of the "*", generating a byte containing that count, followed by an asciz string of all the non-"*" characters in the string. If there is no "*" in the string, a "*" is assumed following the last character in the string.

"string" may be specified as: to generate:

```

AB*C          .Byte  2
               .Asciz  "ABC"

DEF           .Byte  3
               .Asciz  "DEF"

*GHI         .Byte  0
               .Asciz  "GHI"

```

```

.Macro Abbrev String
.NChr  ....2,<String>
.....1=.
.=.+1
.....3=0
.IrpC  x,<String>
.IIf   EQ  'x-*'  ....2=.....3
.IIf   NE  'x-*'  .Byte  'x'
.....3=.....3+1
.EndR
.Byte  000
.....3=.
.=.....1
.Byte  ....2
.=.....3
.EndM

```

The following macro definition generates two calls to different entry points of a subroutine depending on the addressing mode of the "parm" argument.

"parm" may be specified as: to generate:

```

<blank>      Call   SubrR0

R0           Call   SubrR0

#FOO        Call   SubrIn
            .Word   Foo

FOO         Mov    Foo,R0
            Call   SubrR0

```

WILLIAMS'S HACK BOOK
Using and defining macros

```

.Macro Subr  Parm
.....1=0
.IIf  NB  <Parm> .NType  ....1,Parm
.IIf  NE  ....1
.IIf  NE  ....1-^o27
      Mov  Parm,R0
      Call SubrR0

.IIf
.=.+2
      Tst  Parm

.=.-6
      Call SubrIn

.=.+2
.EndC
.IIf
      Call SubrR0

.EndC
.EndM

```

The two entry points could look like this:

```

SubrIn:      Mov    @(SP),R0      ; inline parm
              Add    #2,@SP      ; load into R0
SubrR0:      Add    #2,@SP      ; skip word on return
              ...
              Return

```

Macro definitions can be used to define new macros. For instance the following macro defines new macros with various names, each like the "Subr" macro above in function:

```

.Macro DfSubr Subr
.NChr  ....1,Subr
.IIf  GT  ....1-4.      .Error  ....1;name too long;
.Macro Subr  Parm
.....1=0
.IIf  NB  <Parm> .NType  ....1,Parm
.IIf  NE  ....1
.IIf  NE  ....1-^o27
      Mov  Parm,R0
      Call Subr'R0

.IIf
.=.+2
      Tst  Parm

.=.-6
      Call Subr'In

.=.+2
.EndC
.IIf
      Call Subr'R0

.EndC
.EndM

```

WILLIAMS'S HACK BOOK
Using and defining macros

.EndM

Invoking DfSubr with the following:

```
DfSubr Read
DfSubr Writ
DfSubr Sort
```

would define macros called:

```
Read
Writ
Sort
```

which would accept 1 Parm and act the same way as the Subr macro defined above.

Another use for macros is to provide system independence. For example if you had a program (KED) which ran on RT-11 and you wished to move it to another operating system (say RSX-11M) you might find the following "meta" macros useful in converting RT-11 specific EMT requests to subroutine calls.

```
.Macro DefInd SysCall,Parms
.MCall .'SysCall
.Macro $'SysCall      Parms
        .'SysCall      Parms
.=-2
Call    $'SysCall
.EndM
.EndM
```

For instance you can define a call to \$PRINT which uses the same parameter passing scheme as .PRINT using the following:

```
DefInd Print Addr
```

which will generate the following:

```
.MCall .Print
.Macro $Print Addr
        .Print Addr
.=-2
Call    $Print
.EndM
```

WILLIAMS'S HACK BOOK
Using and defining macros

The \$PRINT macro can then be used just like the .PRINT macro.

```
$PRINT #FOO
Mov    #BAR,R0
$PRINT
```

The way that this works is that the .PRINT macro generates:

```
Mov    #addr,R0
Emt    ^o351
```

The \$PRINT macro generates the following:

```
Mov    #addr,R0
Emt    ^o351
.=-2
Call    $Print
```

Which is equivalent to:

```
Mov    #addr,R0
Call    $Print
```

See Appendix B for a list of DefInd calls for most of the standard RT-11 system requests.

The following macros are used in the monitor to provide different ways to access user program data depending on the monitor type (SJ, FB, XM). This is an example of using macros to provide an interface point.

PUT moves a word from the monitor to the user. Generates a 'Mov' in SJ and FB. Generates a 'Push' and 'MTPI' in XM.

GET Moves a word from the user to the monitor. Generates a 'Mov' in SJ and FB. Generates a 'MFPI' and 'Pop' in XM.

WILLIAMS'S HACK BOOK
Using and defining macros

```
.If EQ MMG$T

.Macro Put Src,Dst
  Mov Src,Dst
.EndM Put

.Macro Get Src,Dst
  Mov Src,Dst
.EndM Get

.IfF

.Macro Put Src,Dst
.If DIF <Src>,<(SP)+>
.If IDN <Src>,<#0>
  Clr -(SP)
.IfF
  Mov Src,-(SP)
.EndC
.EndC

.Macro Put MTPI Dst
  Put
.EndM Put

.Macro Get Src,Dst
  MFPI Src
.If DIF <Dst>,<-(SP)>
  Mov (SP)+,Dst
.EndC
.EndM Get
.EndC
```

CHAPTER 3

H a c k s

STACK OPERATIONS

SUMMARY: Using the stack is often a good way to reduce the size of a program, both the size of the instructions that are in the code and the size of the data areas since the storage on the stack can be reused easily.

The following are examples of code that use the stack and may save space:

Save a work register on the stack, point to a table.

The obvious way is:

```

Push   R1           ; save a work register
Mov    #List,R1     ; point R1 to list
...
Pop    R1           ; restore work register

```

This takes 4 words of instructions.

The shorter way is:

```

List:  Jsr   R1,10$   ; save R1
       .Word #List,R1 ; point R1 to list
       ...           ; skip list
10$:   ...
       Pop   R1       ; restore R1

```

This takes 3 words of instructions.

The two methods have several tradeoffs. The first method is easy to understand. The first method can be used several places in a program to reference the same "List", the second method works only once per "List" since the list is embedded in the code. The second method is "PIC". The second method is shorter.

Replace "Top" with @Top

If the top of the stack contains a pointer to a value which is to become the new top of the stack (replacing rather than pushing the old top), the following is the straightforward code:

```

Mov    @(SP),@SP    ; @Top replaces Top

```

which is a 2 word instruction. (The @(SP) is indirect indexed with an implicit index value of 0).

This is the shorter version

```

Mov    @(SP)+,-(SP) ; @Top replaces Top

```

This is a 1 word instruction.

Manipulate stack and Carry bit

```

Tst   (SP)+         ; pop stack, clear carry
       ; set zero and negative
       ; conditions from value

Com   (SP)+         ; pop stack, set carry

Inc   (SP)+         ; pop stack,
       ; leave carry as is

Mov   (SP)+,(SP)+   ; pop 2 words from stack,
       ; leave carry as is

Cmp   (SP)+,(SP)+   ; pop 2 words from stack,
       ; "randomize" carry

```

"FORTRAN" CALLING CONVENTIONS

Fortran style calls are of the form:

```

Mov   #List,R5      ; point to parameter list
Call  Routin        ; call the subroutine

```

```

List: .Byte Count,Junk ; count is number of parms
       .Word Parm1     ; junk is "undefined"
       ...
       .Word ParmN    ; Nth parameter

```

Inline parameter lists

This requires 5 words for "overhead" + 1 word per parameter

WILLIAMS'S HACK BOOK
H a c k s

This may also be done inline:

```
      Mov    PC,R5          ; point to parameter list
      Br     10$           ; skip list
                          ; first byte of BR is count
      .Word  Parm1
      ...
      .Word  ParmN
10$:  Call   Routin        ; call the subroutine
```

This requires 4 words for "overhead" + 1 word per parameter

Note that the "inline" technique will not work with subroutines that have assumed the "junk" byte (high byte of the word containing the "count") is zero.

Calling SYSLIB routines from Macro Programs

In RT-11 V5.2 a new routine has been added to SYSLIB which aids in calling FORTRAN calling convention routines (also in SYSLIB). The convention for FORTRAN routines include the LACK of a requirement for the called routine to save any registers. The normal convention for MACRO routines is the requirement for the called routine to save registers. The CALL\$F routine in SYSLIB will save registers R1 through R4, call a routine, then restore R1 through R4 and return. The address of the routine to call is placed in R0 and the address of the parameter list for the called routine is placed in R5. As an example, to call the INDEX routine the following could be done:

```
      .Globl Call$F Index
      Mov    Index,R0
      Mov    IParm,R5
      Call   Call$F
      ...
      .Even
IParm: .Byte  2,0
      .Word  Buffer
      .Word  Patter
```

This can be facilitated with a macro as defined below:

WILLIAMS'S HACK BOOK
H a c k s

```
.Macro CallF Subr PList
      .Globl Subr Call$F
      Mov     Subr,R0
.If     NB    <PList>
      Mov     PList,R5
.EndC
      Call   Call$F
.EndM
```

which can be used in the following way:

```
CallF Index IParm
CallF RAD50 RParm
```

SETTING, CLEARING AND TESTING FLAGS

To reduce confusion use the following macros to define mnemonic branches:

```
.Macro Bon Dest
      Bne Dest
.EndM

.Macro Boff Dest
      Beq Dest
.EndM
```

Flags may be words, bytes, or bits.

For word flags the following may be used:

```
SetW:  Mov    PC,Flag      ; PC <> 0 in a program
ClrW:  Clr    Flag         ; clear the flag
TstW:  Tst    Flag         ; 0 -- off, non-zero -- on
      Beq    Off
      Bne    On
TstW:  Tst    Flag
      Boff   Off
      Bon    On
```

For flag bytes the following may be used:

```
SetB:  MovB  #-1,Flag      ; The low byte of the PC CAN
      ; be zero ( this is a common
      ; pitfall)

ClrB:  ClrB  Flag         ; clear the flag

TstB:  TstB  Flag         ; 0 -- off, non-zero -- on
      Bon   On
      Boff  Off
```

For flag bytes that are on odd boundaries the following may be used:

```
SetB:  MovB  @PC,OddByt   .ASSUME OddByt!1 EQ 1
      ; the low byte of the offset
      ; must be odd, hence
      ; non-zero.

      .Odd
OddByt: .BlkB  1
```

The following hacks may be used on byte flags (with care):

```
Set2B: Mov  #-1,Flag1     .Assume Flag1&1 EQ 0
      ; set 2 byte flags   .Assume Flag1+1 EQ Flag2

Clr2B: Clr   Flag1        .Assume Flag1&1 EQ 0
      ; clear 2 byte flags .Assume Flag1+1 EQ Flag2

Tst2B: Tst   Flag1        .Assume Flag1&1 EQ 0
      Boff  xland2        .Assume Flag1+1 EQ Flag2
      Bon   xlor2         ; test 2 flags
      ; if Flag1 AND Flag2 off
      ; if Flag1 OR Flag2 on

      .Even
Flag1: .BlkB  1
Flag2: .BlkB  1
```

Note that bit flags are generally not worth the bother, because to set, clear, or test them individually requires a longer instruction (BIS, BIC, or BIT) which invariably requires a mask word.

If you do use bit flags you can improve things somewhat with the following tricks:

Put the most popular (to test) bit flags at 100000 and 200. These can be tested using TST(B):

```
Popull =: 100000
Popul2 =: 000200
Rare3  =: 000001

      .Assume Popull EQ 100000

Tst   BitFlg
Bmi   Pop10n
Bpl   Pop10ff

      .Assume Popul2 EQ 000200

TstB  BitFlg
Bmi   Pop20n
Bpl   Pop20ff
```

The BIT instruction can be used to test several bits at once. At times this capability can actually be useful.

```
Bit   #Popull!Rare3,BitFlg
Bmi   Pop10n      .Assume Popull EQ 100000
      ; Popull is on, Rare3 don't
      ; care
Bon   Rare30n    ; Popull is off, Rare3 on
...   ; Both are off

Bit   #Popull!Rare3,BitFlg
Bpl   Pop10ff    .Assume Popull EQ 100000
      ; Popull is off, Rare3 don't
      ; care
Boff  Rare30ff  ; Popull is off, Rare3 off
...   ; at least 1 is on
```

N TIME SWITCHES

Sometimes it is desirable to perform a piece of code once only (initialization code). The following code examples provide for once only, nth only and alternate times switches.

One time switch

```
ASL  #100000      ; first time thru?
Bcs  First        ; yes
                        ; Note this is
                        ; self-modifying code.
```

```
ASL  FFlag       ; first time thru?
Bcs  First        ; yes
                        ; Note this is "purer"
                        ; code.
```

```
FFlag: .Word 100000--. ; "constant" but modified
```

Two time switch

The same idea can be applied to first 2 times.

```
ASL  #140000
Bcs  Fst2nd
```

...

Sixteen time switch

And up to 16 times.

```
ASL  #177777
Bcs  Fst16
```

Alternate time switch

This code alternates between Flip and Flop.

```
COM  #0
Bne  Flip
Beq  Flop
```

CARRY

Success and Failure Exits

Subroutines often return a success/error status by clearing the carry bit for success, setting it for failure. The following code provides success and error exits in a succinct manner:

```
Ok:   Tst      (PC)+
Error: Sec
      [Mov     -(SP),Rx]
      Return
```

Note that the SEC instruction may be following by common exit cleanup instructions (such as stack pops) as long as the instructions do not modify carry.

Saving Carry

The following code can be used to save the status of the carry bit:

```
Ror   -(SP)
...
Tst   @SP
Bmi   Carry
...
Rol   (SP)+
```

if a register is available (with an even value in it)

```
Ror   R0
...
Asl   R0
```

Another way with an even valued register contents:

```
ADC   R0      ; make odd if carry set
```

Set Carry If Non-Zero Flag

```
Neg   Flag      ; destructive
Cmp   #0,Flag    ; non-destructive
```

Clearing a Variable Without Changing Carry

```
Mov    #0,Loen
Bic    R0,R0
```

BOUNDS CHECKING SUBROUTINE

Often you can leave carry in the proper state for success/failure indication on exit by arranging the operands of the CMPs in a suitable order:

```
Letter:
  Cmp    R0,#'A      ; uppercase A or greater?
  Blo    10$         ; no, exit with carry set
  Cmp    #'Z,R0      ; uppercase Z or lower?
10$:
  Return                ; carry set -not uppercase
                      ; letter, clear, is letter

Binary:
  Cmp    R0,#'1      ; high limit or greater?
  Br     20$         ; join common code

Octal:
  Cmp    R0,#'7      ; high limit or greater?
  Br     20$         ; join common code

Hex:
  Cmp    R0,#'F      ; high limit or greater?
  Blo    30$         ; yes, too high
  Cmp    #'A,R0      ; low limit?
  Bhis   30$         ; no, in range

Decimal:
  Cmp    R0,#'9      ; high limit or greater?
20$:
  Blo    30$         ; too high
  Cmp    #'0,R0      ; too low?
30$:
  Return
```

USING AUTO INCREMENT/DECREMENT
TO ADD/SUBTRACT 1,2,4 FROM REGISTER(S)

```
Cmp    (R0)+,(R0)+   ; R0 + 4 (must be even)
Tst    (R0)+         ; R0 + 2 (must be even)
CmpB   (R0)+,(R0)+   ; R0 + 2 (may be odd)
TstB   (R0)+         ; R0 + 1 (may be odd)
Inc    R0            ; R0 + 1 (the easy way)

Cmp    -(R0),-(R0)   ; R0 - 4 (must be even)
Tst    -(R0)         ; R0 - 2 (must be even)
CmpB   -(R0),-(R0)   ; R0 - 2 (may be odd)
TstB   -(R0)         ; R0 - 1 (may be odd)
Dec    R0            ; R0 - 1 (the easy way)

Cmp    (R0)+,(R1)+   ; R0 + 2 (must be even)
                      ; R1 + 2 (must be even)
Cmp    -(R0),-(R1)   ; R0 - 2 (must be even)
                      ; R1 - 2 (must be even)

Cmp    (SP)+,(R1)+   ; pop stack
                      ; R1 + 2 (must be even)

Cmp    (SP)+,-(R1)   ; pop stack
                      ; R1 - 2 (must be even)

Cmp    -(SP),(R1)+   ; push stack
                      ; R1 + 2 (must be even)

Cmp    -(SP),-(R1)   ; push stack
                      ; R1 - 2 (must be even)

CmpB   (SP)+,(R1)+   ; pop stack
                      ; R1 + 1 (may be odd)

CmpB   (SP)+,-(R1)   ; pop stack
                      ; R1 - 1 (may be odd)

CmpB   -(SP),(R1)+   ; push stack
                      ; R1 + 1 (may be odd)

CmpB   -(SP),-(R1)   ; push stack
                      ; R1 - 1 (may be odd)

Cmp    (PC)+,(R1)+   ; skip next instruction
                      ; (must be 1 word long)
                      ; R1 + 2 (must be even)
```

WILLIAMS'S HACK BOOK
H a c k s

```

Cmp      (PC)+,-(R1)    ; skip next instruction
                    ; (must be 1 word long)
                    ; R1 - 2 (must be even)

CmpB     (PC)+,(R1)+    ; skip next instruction
                    ; (must be 1 word long)
                    ; R1 + 1 (may be odd)

CmpB     (PC)+,-(R1)    ; skip next instruction
                    ; (must be 1 word long)
                    ; R1 - 1 (may be odd)

```

DOING A ROUTINE TWICE

```

Twice:   Call    @PC
Once:    ...
        Return

```

INCREMENTING DEBUG COUNTS

```

Count:   Inc      (PC)+
        .Word    .-.
        Inc      #0

```

NOOPERATIONS

```

Nop      (240)
Nop2     =:      260

```

WILLIAMS'S HACK BOOK
H a c k s

```

10$:     Br      10$

10$:     B..     10$

        Bic     #0,R0      ; affects Cond Codes
        Bis     #0,R0
        Bis     R0,R0

        Mov     R0,R0      ; affects Cond Codes
        ...
        Mov     SP,SP

```

INTENTIONAL TRAPS (BUGCHECKS)

```

Jsrr     R0,R0
Jsrr     R0,R1
...
Jsrr     PC,PC

Jmp      R0
...
Jmp      PC

```

Note: Mov R0,@#1 will not always trap. HALT will not always stop.

```

Br      .      ; Hang
Jmp     .      ; Hang

```

also

```

Tst     MustB0 ; error if non-zero
Bne     .      ; is a possible hang on bug

```

TRAPPING INTERRUPTS

```
=:      Vector
.Word   .+2
Halt    ; to stop on interrupt

=:      Vector
.Word   .+2
BPT     ; to go to debug on interrupt
```

NO-OPING VECTORS

```
=:      Vector
.Word   .+2
RTI     ; ignore any interrupt
```

TESTING FOR 1 BIT SET

In some circumstances you need to determine if exactly one bit is set in a word (byte). This can be done using the following code fragment:

```
...           ; R0 contains word to be tested
Mov   R0,-(SP) ; copy it for destruction
Beq   Bit0     ; no bits set (stack needs popped)
Mov   @SP,-(SP) ; and a second copy
Neg   @SP      ; take 2's complement (invert _inc)
Bic   (SP)+,(SP)+ ; try clearing it out
Beq   Bit1     ; result is 0, exactly 1 bit in R0
...           ; else more than one bit in R0
```

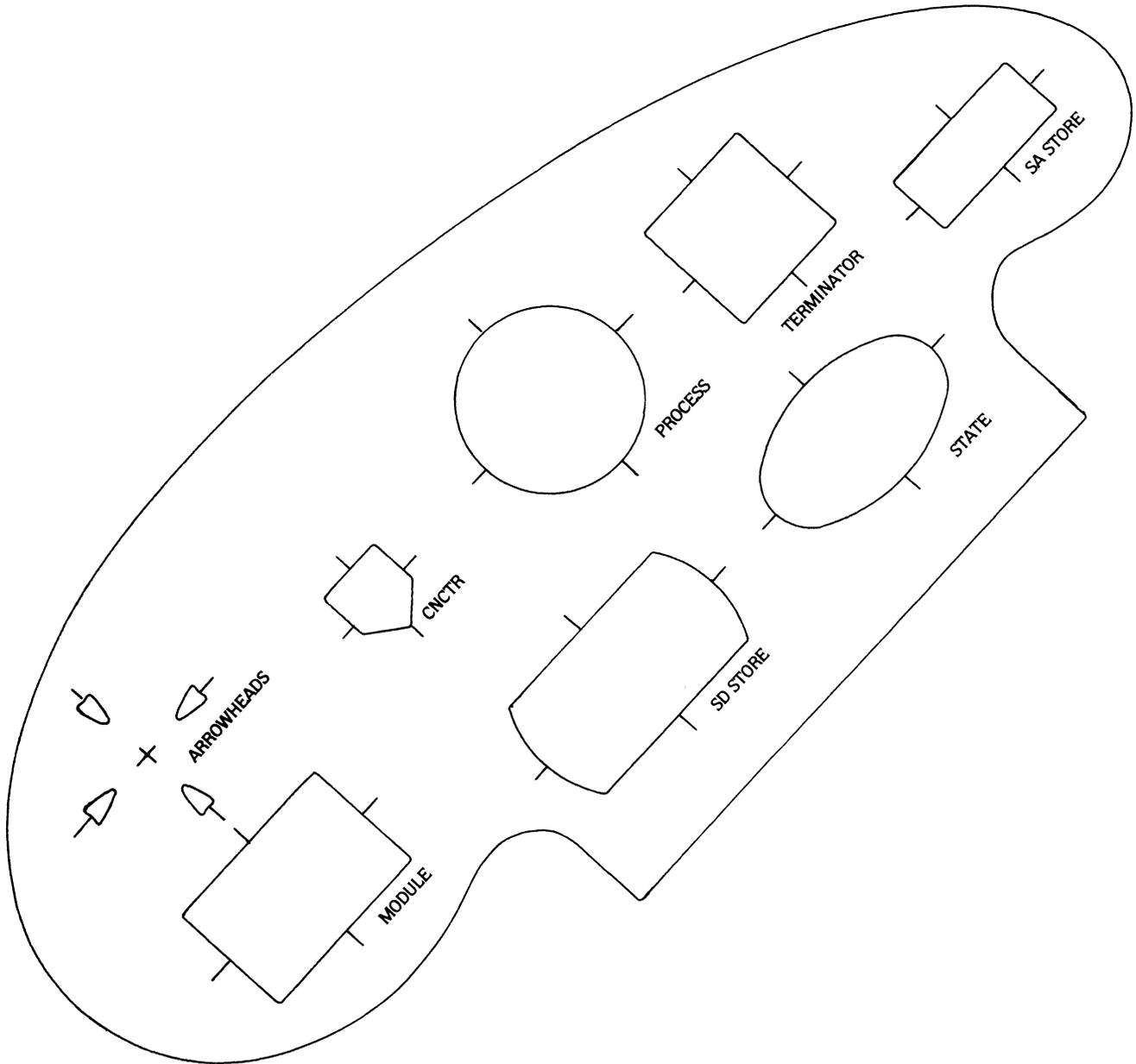
CALLING and RETURNING with "DIRTY" STACK

It is possible (and sometimes necessary) to call a subroutine and have the routine return with the stack alignment altered from when the call occurred. With the normal (CALL xxx or JSR PC,xxx) form, the stack must be correctly aligned (or the return address must be "artificially" placed on the top of the stack). Using other forms of the JSR (JSR Rx,xxx) the return address is in a register, rather than on the stack. So routines that are called this way can successfully return even though the stack was misaligned (either on purpose, or more likely, accidentally). The moral is not to dismiss a subroutine as not being a stack corrupter, because it returned successfully.

To Be Continued in October Issue.



“LEVERAGE”



LANGUAGES AND TOOLS SIG

LANGUAGES & TOOLS

Chairperson

Katherine Hornback
Lear Siegler, Inc.
Grand Rapids, MI

Vice Chairperson

Barbara Chase
Hughes Aircraft Company
Los Angeles, CA

Symposium Coordinator

J.R. Westmoreland
Utah Power and Light
Midvale, UT

Asst. Symp. Coordinator-Campground/Suite

Melody Westmoreland
Midvale, UT

Session Chair Coordinator

Earl S. Cory
Eaton Information Management
Westlake Village, CA

Newsletter Editor

Alan L. Folsom, Jr.
Fischer & Porter Co.
Warminster, PA

Session Note Editor

Mark Katz
GTE Government Systems
Needham, MA

Tape Librarian

Dave Martin
Hughes Aircraft Co.
Los Angeles, CA

GAPSIG Standards Coordinator

Jim Flatten
Ames Lab
Ames, IA

Standards Coordinator

FORTAN Coordinator
Jay W. Wiley
Bechtel Power Corp.
Norwalk, CA

DEC Counterpart

PDP-11
Joe Mulvey
Digital Equipment Corp.
Nashua, NH

VAX

William Segal
Digital Equipment Corp.
Nashua, NH

UNIX Coordinator

Rod Creason, Jr.
Sacramento, CA

MODULA II Coordinator

Jack R Davis
Philips Home Interactive Systems
Knoxville, TN

Ada Coordinator

Bob Gable
Lear Siegler
Grand Rapids, MI

Wishlist Coordinator

Alan Rizzuto
EMC Controls, Inc.
Cockeysville, MD

PASCAL Coordinator

Derek R. Smith
Wilmington, DE

RSTS Coordinator

George Stuart
Ogden - Weber/AVC
Ogden, UT

Methods Coordinator

PSS Committee Rep.
Pat VanMunn
Measurex, Inc.
Cupertino, CA

TeX Coordinator

Large System Liaison
Store Coordinator
Sam Whidden
American Mathematical Society
Providence, R.I.

CMS/MMS Coordinator

Edgar Whipple
Lawrence Berkeley Lab.
Berkeley, CA

RT-11 Coordinator

Michelle C. Wong
Sacramento, CA

DMS Liaison

Keith Hare
(Home)
Granville, OH

AI SIG Liaison

Don Rosenthal
Space Telescope Science Inst.
Baltimore, MD

CL SIG Liaison

Jim Wilson
Pfizer Inc.
Terre Haute, IN

VAX/VMS Liaison

Louise A. Wholey
Measurex Corp.
Cupertino, CA

DEC Personnel Coordinator

Howard Holcombe
Medford, NJ

EX-Chairman

James W. Livingston, Jr.
Measurex Corp.
Cupertino, CA

Member-At-Large

Shava Nerad
Telematic Systems
Cambridge, MA

RSX Liaison Committee

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

TABLE OF CONTENTS

EDITOR'S NOTES	2
LETTER FROM THE CHAIR	3
PRELIMINARY PLANS FOR FALL SYMPOSIUM	4
LETTERS	5
FUN WITH DCL	7
DOCUMENTATION SURVEY FORM	9
LTSIG FORTRAN 8X STANDARDS IMPROVEMENT REQUEST BALLOT	11
STATUS OF WORK TOWARDS REVISION OF PROGRAMMING LANGUAGE FORTRAN (FIB-1)	24

EDITOR'S NOTES

Congratulations, you are now holding in your hand the first issue of the combined DECUS newsletter(s). At the time I write this, it is the intention of the DECUS staff to distribute this issue to all DECUS members, by way of introduction to the new format. I hope they do, because I have included once again, by special request, the Fortran Information Bulletin, which describes the proposed revisions to Fortran for the forthcoming standard. In addition, we have been requested by X3J3 to distribute a questionnaire, soliciting opinions regarding the proposed standard. Please, please take the time to fill this out, and return it to Jay Wiley. This is an opportunity to directly affect an issue that may well concern you in the near future. Let's not waste it. An overview, and the questionnaire, directly precedes the FIB.

In a similar vein, the DEC Software Documentation folks have submitted a survey form, to determine your feelings regarding their various documentation products. If you don't fill this out and return it, you will never be allowed to complain about documentation again.

As you may have noticed, a new name, Leverage, has been chosen for the newsletter. This is certainly classier than "The Heap", and I hope we can live up to all it implies. Please help to do so, but submitting articles, letters, or suggestions. Earl Cory has volunteered to start a new column, "Fun with DCL", which should prove useful. He can use your suggestions or submissions, and we both welcome your comments on that column, or any other you'd like to see.

By the way, I received a nice letter from Brian Cooper of Eyring Research Institute, which raised a number of issues which I hope to address in the near future. At the moment, though, I'd just like to say that his suggestion got my vote for a new newsletter name. He suggested "Software Tool Pigeon", which would, of course, be abbreviated "Stool Pigeon". Clever, not too high falutin', and full of possibilities for illustrations. I may adopt it for the name of a column.

On one final note, I've been receiving more letters lately, for which I am very grateful. I've reprinted a couple of them in this issue, and will continue to do so unless specifically requested otherwise by the writer. Letters, comments, and suggestions can be addressed to me at:

Alan L. Folsom, Jr.
Dept 431, Fischer & Porter Co.
E. County Line Road,
Warminster, Pa. 18974

Letter from the Chair

- K. Hornbach

Welcome to the newly formatted, newly named Languages and Tools newsletter! While L & T will probably won't have articles in every single monthly issue, you can be sure that we will reach you most months. And we have started out with an outstanding issue with some vitally important articles for Fortran users.

The new name for the newsletter, *Leverage*, came from Sam Whidden of the American Mathematical Society. The languages and tools under the L&T SIG have productivity improvements as their central focus - they are the *leverage* that the software engineer uses to get ever more quantity and quality from his work.

If you are at all interested in Fortran, be sure to read over the section on the proposed Fortran 8x standard. We are asking all DECUS members who use Fortran to fill in the "Standards Improvement Request", the results of which will be presented to the Fortran 8x committee. Full details on how this SIR came about are found later in this issue. But I wanted to stress here the importance of **reading the SIR ballot and sending it in!**

Preliminary Plans for Fall Symposium

The U.S. Fall DECUS Symposium is planned for Decemeber 9 - 13 at the Disneyland Hotel in Anaheim, Ca. The Languages and Tools SIG is busily planning to make it one of the most valuable ever for SIG members.

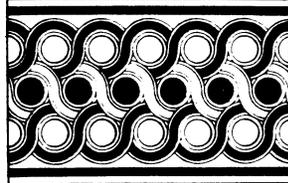
Software Project Management is the theme for the Languages and Tools SIG for this Symposium. We will be exploring the issue from both the software management and support tools points of view.

We also have a full schedule of pre-symposium seminars planned for the Sunday before Symposium:

- **Developing Medium Scale Applications using VAX Ada** - to be taught by a DEC Ada developer
- **Using LaTeX for Text Formatting** - LaTeX is an easy-to-use macro package that can be used with TeX*, a public-domain, typeset-quality text formatter. This seminar will be taught by Leslie Lamport, the author of LaTeX.
- **Software Development Tools** - This seminar will cover what software tools are available to support the entire software life cycle, as well as give practical advice on how to obtain, support, and convince people to use these tools. This seminar will be taught by Kathy Hornbach, Chair of the Languages and Tools SIG.
- **Structured Analysis for Real Time Systems** - this seminar describes a version of the Yourdon/DeMarco Structured Analysis method, with extensions for real time systems. This method was developed by Lear Siegler, and is in use by several major aerospace firms. It will be taught by Derek Hatley, its developer.
- **Managing Software Projects with CMS and MMS** - DEC's Code Management System and Module Management System are powerful tools for managing source code during software development. This seminar will cover how to effectively integrate the use of these tools into the context of your project development.

* TeX is a trademark of the American Mathematical Society

Systems Alternatives



43 State Street
Montpelier, VT 05602
802-229-0823

July 19, 1985

Alan Folsom, Editor
"The Heap," DECUS LTSIG Newsletter
Dept 431, Fischer & Porter Co.
E. County Line Road
Warminster, PA 18974

To The Editor:

As a former consultant to Digital Equipment Educational Services with the IVIS applications development group within the Computer Aided Instruction section, I would like to comment on DEC's response to the LTSIG wishlist item nine: "DEC should provide on-line CAI classes for all new software tools."

DEC responded that Ed. Svcs. feel that CAI courses "are not appropriate for technical people...They [CAI in general] are expensive to develop, and must justify themselves as a product."

Although these comments may seem unrelated, their relationship is well defined in the development of CAI. To develop effective CAI for a technical market, the educational designer must understand the nature of his subject matter thoroughly, and have thorough empathy with (i.e. be able to anticipate the thinking patterns of) his target audience.

For a more limited process or tool (for instance, maintaining a printer), a limited flow of control may be defined, which includes "outs" like "If these steps still do not solve the problem, call for help." Defining the process of USING a software tool is a development problem on the same scale as writing the tool itself, since the CAI developer must in effect "automate" the limits of the functional specifications to exercise the learner's performance.

What I think that DEC Ed. Svcs. could produce is FAMILIARIZATION instruction—"roadmap" CAI for each tool. These could even be used on the DEC Electronic Store system to "justify themselves as a technical product."

We technical people do not need CAI to teach us how to apply a tool. We can use the age-old documentation/experiment/observation/documentation reference cycle. But to easily evaluate and learn the increasingly higher level tools with which we are presented, we need to have a tag into the purpose and principles of the tool. With this edge, documentation and reference materials will become more quickly accessible to the engineer involved.

I hope that DEC will reconsider their response to this wishlist item with this issue in mind.

Shava Nerad
Systems Analyst
Systems Alternatives, Inc.
13 State St., P. O. Box 1439
Montpelier, VT 05602

L&T-5 802/229-0823

Northern Telecom, Inc
Network Support Systems Division
P. O. Box 649
Concord, NH 03301
July 10, 1985

Alan J. Folsom, Jr.
Dept 431, Fischer & Porter Co.
County Line Road
Warminster, PA 18974

Dear Mr. Folsom,

I am writing regarding the plan to incorporate the SIG newsletters into one large newsletter. While I am aware that the change has been officially announced, the debate may continue for some time. If there is any possibility of a reversal of the policy change, you might find an additional argument useful.

I work for a company which employs quite a few programmers, who make use of a 3-Vax cluster and a DECnet of other Vaxes and RSX-11 systems. A number of different applications and development projects are ongoing simultaneously.

Our library receives all DECUS newsletters, some of which are in constant demand. Even if the exact same information could be distributed in one large newsletter as is currently published in the small ones, a loss of usefulness would take place; whereas users on different projects can currently read the newsletters which interest them, under the new configuration only one user will be able to read the magazine at a time.

It is my feeling, and the consensus of my coworkers, that the combination of the newsletters will lessen the usefulness of the information they hold.

Sincerely yours,

Beverly T. Kasper
System Support Specialist

L&T-6

FUN WITH DCL

This column is a new feature of the Language & Tools section of the combined newsletter. Each month I will present some DCL commands or command procedures that have been found to be useful by me (or you) in software development. Useful hints and tricks that may be done with DCL will be included.

I am not restricting this to VAX/VMS. RT-11, RSTS/E, RSX-11M, all have DCL to some level. I encourage each of you to send us any DCL procedures, symbols, one-liners, etc that you find to be useful. Address your suggestions to me or Al Folsom, the Leverage newsletter editor.

Menus to create a friendly interface for users are easily written in DCL. The user's own login command file is a DCL command procedure that might be a menu or call one.

Part of the menu procedure might be of this form:

```
$      V=F$VERIFY(0)
$      Say = "Write Sys$output"
$      On control_y then goto ABORT
$ ENTRY:
$      Time_stamp = "      "+F$TIME()
$      Clear
$      Say Time_stamp
$      Type Sys$input
```

System Utility Menu

```
[00] Exit.
[01] LOG OFF
[02] Copy files
[03] Backup a disk
[04] Mount a disk
[05] Initialize a tape
```

```
$ REENTER:
$      Inquire CHOICE " Choice? <0>"
$      Say ""
$      If CHOICE .eqs. "" then goto EXIT
$      If CHOICE .gt. 0 .and. CHOICE .le. 5 then goto 'CHOICE'
$      Say "Improper Choice, Please Re-Enter"
$      Say ""
$      Goto REENTER
$ 1:
$      Logoff
$ 2:
$      @COPY.COM
$      Goto ENTRY
$ 3:
```

```
$      @BACKUP.COM
$      Goto ENTRY
$
$
$
```

A system-wide menu could be called by defining a symbol in SYLOGIN.

For example:

```
$ MENU := @device:[directory]MENU.COM
```

Then when ever the user types MENU, the menu command file is activated.

Earl S. Cory

366 North Nueve Court
Camarillo, California 93010

DIGITAL LANGUAGES AND TOOLS DOCUMENTATION SURVEY

Help us provide you with the best documentation possible. Use the following questionnaire to give us your opinion of the documentation set that accompanies the DIGITAL language or tool that you use most often. (You may photocopy this questionnaire to comment on additional documentation sets.) Return the completed survey to:

Reader Survey
 Digital Equipment Corporation
 110 Spit Brook Road ZK02-3/004
 Nashua, New Hampshire 03062-2698

Your job _____ How long _____
 Prior software experience _____ years
 Name and version of product used _____
 Length of time using this product _____ years

INSTRUCTIONS: To the right of each statement are three columns of numbers. Write the different manual titles on the line at the top of each column. Then respond to each statement with respect to each manual by circling the number that best represents your opinion:

	1	2	3	4	5
	Strongly Disagree	Disagree	No Opinion	Agree	Strongly Agree
PUT MANUAL TITLES HERE	_____	_____	_____	_____	_____
I can find information quickly.	1 2 3 4 5			1 2 3 4 5	1 2 3 4 5
The manuals are well organized.	1 2 3 4 5			1 2 3 4 5	1 2 3 4 5
The indexes are good.	1 2 3 4 5			1 2 3 4 5	1 2 3 4 5
The table of contents is good.	1 2 3 4 5			1 2 3 4 5	1 2 3 4 5
The material is easy to read.	1 2 3 4 5			1 2 3 4 5	1 2 3 4 5
The material is technically accurate.	1 2 3 4 5			1 2 3 4 5	1 2 3 4 5
The page layout is easy on my eyes.	1 2 3 4 5			1 2 3 4 5	1 2 3 4 5
The tutorial information is adequate.	1 2 3 4 5			1 2 3 4 5	1 2 3 4 5
The conceptual information has adequate explanations.	1 2 3 4 5			1 2 3 4 5	1 2 3 4 5
The illustrations are useful.	1 2 3 4 5			1 2 3 4 5	1 2 3 4 5
The examples clearly demonstrate product use.	1 2 3 4 5			1 2 3 4 5	1 2 3 4 5

(over)

Use the following scale for answering the next set of questions:

	1	2	3	4	5						
	Never	Seldom	No Opinion	Occasionally	Always						
I use the index to find information.			1 2 3 4 5		1 2 3 4 5			1 2 3 4 5			
I use running heads to locate information.			1 2 3 4 5		1 2 3 4 5			1 2 3 4 5			
I flip through the book to find information.			1 2 3 4 5		1 2 3 4 5			1 2 3 4 5			
I use the table of contents to find information.			1 2 3 4 5		1 2 3 4 5			1 2 3 4 5			
I use this manual more than others in the set.			1 2 3 4 5		1 2 3 4 5			1 2 3 4 5			

Answer YES or NO to the following statements:

- I can identify and order the correct manual for my needs. (Y) (N)
- Online HELP answers most of my questions. (Y) (N)
- I find critical information in the Release Notes. (Y) (N)
- I would like to keep Quick References for several products in a single binder. (Y) (N)
- The size of Quick References should be: 3" X 5", 5" X 7", or 7" X 9"
- Feel free to write comments and suggestions in the space below.
Thank you for taking the time to fill out this questionnaire.

Introduction:

During the Spring and Fall '84 Symposia, a number of DECUS members expressed their concern about the proposed content of the next version of the ANSI 3.9 standard (i.e. FORTRAN 8X; it is anticipated, that the next standard will be finally accepted, sometime in the 1980s). During a BOF on FORTRAN 8X at the Fall '84 Symposium, it was agreed, that DECUS members need to know a lot more about this proposed standard. As a result of this BOF, the L&T SIG published a summary of the FORTRAN 8X language in the May '84 copy of the L&T newsletter, The HEAP. This summary by Jerrold Wagener was called the Fortran Information Bulletin (FIB-1). Also, they invited the ANSI FORTRAN 8X committee (X3J3) to attend the Spring '85 Symposium and to participate in a Q&A session on the proposed standard.

Three members of X3J3 came to New Orleans (Jeanne Adams, chairman; Dr. Walt Brainerd, technical lead; and Dr Jerrold Wagener) and gave a one hour formal presentation on FORTRAN 8X; this was followed by a one hour Q&A session. Over 500 DECUS members attended these sessions. After the formal Q&A, an additional 2.5 hours of Q&A were held in the L&T suite. It was concluded, that a Standards Improvement Request Ballot was the best way, to obtain a consensus opinion of what the DECUS membership wants in the next FORTRAN standard. The results of the SIR ballot will be sent to the X3J3 committee, and they will again attend another Q&A session at the Fall '85 Symposium. The U.S. chapter of DECUS is also working to sponsor a full voting member in X3J3.

The results of this SIR will be used to define a DECUS position on the proposed FORTRAN 8X standard. The results can also directly affect the final contents of that standard. Kevin Harris of Digital Equipment Corporation has provided DECUS, the input on the implementation difficulty and the performance overhead for this version of the standard.

The remainder of this article consists of:

- 1) An explanation of the ballot;
- 2) The ballot; and
- 3) A copy of FIB-1.

REMEMBER TO VOTE! Your votes will be used by DECUS, DEC, and X3J3 to determine the consensus opinion of DEC users.

Explanation of the Ballot:

The following explanation discusses the differences between the current FORTRAN 8X language (as embodied in Standing Document 8 of the Technical Subcommittee for the revision of FORTRAN, or X3J3/S8 for short), and the one described in FIB-1. X3J3/S8 is currently at revision 95, dated June 1985 (95th meeting; actually, S8 is less than 2 years old).

The two major changes since FIB-1 was written are that:

- 1) All aspects of the Event Handling mechanism have been removed and replaced by a Condition/Enable mechanism. Unlike Event Handling, the Condition/Enable mechanism is procedure based and geared toward handling common occurrences in ordinary FORTRAN programs, like overflow and underflow. It is remarkably similar to the existing VAX condition handling mechanism, currently available to VAX FORTRAN programmers through the use of the LIB\$ESTABLISH, LIB\$REVERT, LIB\$STOP, and LIB\$SIGNAL routines, user written condition handlers, and useful system default handlers. On the ballot, this item is listed under section VIII, Execution Control.

- 2) Entity oriented declarations have been removed. These are shown in section 2.1 of FIB-1 as the "declare-statement". In the current S8, declarations are attribute oriented, like FORTRAN 77, except that additional attributes can be added to a keyword. Such as:

```
REAL, ARRAY(100,200) :: A,B
```

instead of:

```
REAL A(100,200), B(100,200)
```

Also, there have been a large number of minor changes in content and terminology. However, the change in organization of S8 is significantly different from that of the FORTRAN 77 standard and is not described in FIB-1. This ballot is organized according to the chapters in the S8 document, rather than haphazardly, as has been the case with previous surveys on this subject. The major subject headings with Roman numerals on the ballot correspond exactly to Version 95 of X3J3/S8. The items in the ballot generally reflect the content of those sections. Sections I., II., XIV., and XV. are not shown, because they do not describe new language elements.

The individual ballot items generally cover the space of new language features included in FORTRAN 8X. The INCLUDE statement is shown under the appropriate heading, even though it is not currently allowed in FORTRAN 8X. The reason is that INCLUDE has proven to be very popular with the user community; thus it is important to assess its desirability side-by-side with the extensive new MODULE/USE facility in FORTRAN 8X designed to fill its role.

The items have been grouped in such a way as to give a modest size ballot, but containing sufficient detail, that the results of the ballot will be useful in guiding the DEC and DECUS representatives in their deliberations with the X3J3 committee.

The deprecated features section of the ballot consists of a single question. It is explained below, together with the current list of deprecated features. Also, given below is a short explanation for each feature listed on the ballot, organized according to the sections in the S8 document:

Deprecated Features in FORTRAN 8X:

Several of the extensions in FORTRAN 8X are designed to replace, rather than augment, existing FORTRAN 77 language features. The rationale for these changes is described elsewhere. These existing features are said to be deprecated. A list of these deprecated features is given below. Ordinarily, these older features would simply be removed from the new standard. However, since this action would make it difficult for the users to make the transition to the use of the new standard, they are deprecated instead. Deprecation means that, rather than simply disappearing in the FORTRAN 8X standard, these existing features are fully allowed in FORTRAN 8X programs. However, since many of them do not coexist comfortably with one or more new features, they are potential targets for removal in FORTRAN 9X (the presumed follow on standard to FORTRAN 8X).

This part of the ballot assesses user feelings on the subject of deprecated features. It simply questions whether you are willing to allow the X3J3 committee to mark some existing FORTRAN 77 features as deprecated in the FORTRAN 8X standard, as is currently the intent.

A YES vote means that you agree, that the functions performed by the deprecated features are adequately covered by new features in FORTRAN 8X, and that the difficulties encountered in the transition period are adequately compensated for by the power of the new features for future use.

A NO vote means the opposite. It means that you disagree with the concept of having deprecated features in FORTRAN 8X. It means that you do not consider that new FORTRAN 8X features cover the uses of the existing features, or that the cost of converting existing code to the new features will outweigh the benefits of those new features. It means that you wish the committee to design only those new features, that are strictly upwards compatible with existing features and practice, and that are efficient when used with the old features. It means that you do not want the FORTRAN 9X committee to have the power to drop FORTRAN 77 features and practice from the FORTRAN 9X language.

List of deprecated features (From Chapter 15 of X3J3/S8, Version 95, June 1985):

- 1) FORTRAN 77 source form (Columns 1-5 for labels, column 6 for continuation, column 73-80 for sequence numbers, insignificant blanks in statements).
- 2) Alternate return.
- 3) Assumed size dummy arrays.
- 4) Passing an array element or substring to a dummy array.
- 5) Specific names for intrinsic functions.
- 6) Statement functions.

- 7) BLOCK DATA program unit.
- 8) Arithmetic IF statement.
- 9) ASSIGN statement.
- 10) Assigned GOTO statement.
- 11) COMMON statement.
- 12) Computed GOTO statement.
- 13) DATA statement.
- 14) DIMENSION statement.
- 15) DOUBLE PRECISION statement.
- 16) ENTRY statement.
- 17) EQUIVALENCE statement.
- 18) FORTRAN 77 form of DO statement.
- 19) PAUSE statement.
- 20) X edit descriptor.

III. Lexical Elements:

FORTRAN 8X has two source forms. One is the existing FORTRAN 77 source form, with the familiar rules concerning columns 1-5 for labels, column 6 for continuation, and 72 columns of significant characters. The new form, called "free form source", is largely incompatible with both the FORTRAN 77 form and the "tab format", which is familiar to users of the various DEC FORTRAN compilers. The FORTRAN 8X standard prohibits any mixing of the two forms in a single compilation unit. The items on the ballot are:

Significant blanks. The use of the character " " (blank) has always been insignificant in FORTRAN. It is significant in FORTRAN 8X. Programs are in error, if they run lexical elements (constants, variable names, keywords, operators) together without separating them with blanks. They are also in error if extra blanks are used.

Insignificant separator. Many users expressed the desire to separate digit groups on long numeric constants using blanks. This capability is provided by allowing the "_" (underscore) character to be treated as insignificant when used in numeric constants.

Lines up to 132 characters. FORTRAN 8X free form source makes no provision for sequence numbers in columns 73-80, which exist in many older FORTRAN source programs. By comparison, VAX FORTRAN allows long source lines, but under user control with a compiler qualifier.

Special usage for columns 1 to 6 is eliminated. Code can start in column 1, if there is no label on the line. Continuations are indicated by an "&" (ampersand) character at the end of the previous line.

The "!" (exclamation point) character is used to indicate end-of-line comments. This usage is familiar to most users of DEC FORTRAN compilers. Multiple statements are allowed on a line in FORTRAN 8X. This is allowed in some DEC FORTRAN implementations, but not VAX FORTRAN. Problems using this feature with line oriented debuggers are anticipated.

IV. Data Types:

Numeric type declaration capabilities are described in section 2.3 of FIB-1.

Character constant syntax has been extended, to allow the use of the quote character to delimit strings; this is in addition to the apostrophe character.

The derived data type facility is described in section 2.4 of FIB-1. Users familiar with the data structuring capability supported in VAX FORTRAN V4 should note, that the FORTRAN 8X derived data type facility is significantly different. The syntax elements of the two are somewhat different. For example, the structure qualification character in VAX FORTRAN V4 is "."; it is "%" in FORTRAN 8X. The VAX FORTRAN syntax was deliberately chosen to be sufficiently different, so that both facilities could eventually be used in the same program, if this was desirable.

A major difference between VAX FORTRAN and FORTRAN 8X is that the FORTRAN 8X derived data type facility is strongly checked, similar to the Pascal and Ada languages; whereas the VAX FORTRAN record structure facility is not strongly checked and can be used in very flexible ways, much like the equivalent facilities in the C language. Neither language supports a pointer data type, because such a type will frequently defeat compiler optimization without the programmer being aware of the degradation.

Both FORTRAN 8X and VAX FORTRAN V4 support a variant component capability. In FORTRAN 8X, a tagged method is used; in VAX FORTRAN an untagged method is used.

In FORTRAN 8X, some component attributes can be specified by run-time parameters for subroutines and functions. This capability is called parameterized derived types, and it is allowed for the attributes of array dimensions, character length, precision specifications, and range specifications.

In FORTRAN 8X, constants of a derived type can be specified by giving the type name, followed by a parenthesized list of the atomic component values. Neither VAX FORTRAN nor FORTRAN 8X has a syntax for allowing shortcut references to components of a structure.

On an entirely different subject, FORTRAN 8X allows the BIT data type. An earlier proposal for bit strings was removed from the standard. The current proposal allows bits to be treated as normal data items rather than strings. Presumably vendors will allocate bit arrays densely packed into memory. To allow this allocation, storage association between bit entities and numeric and character entities is prohibited.

V. Entity declaration and specification statements:

While entity oriented declarations have been removed, the declaration form has been extended beyond the FORTRAN 77 form; see the example described above in the explanation under point 2 for the "ARRAY" attribute. Several such different attributes are allowed in FORTRAN 8X. The first ballot item under this heading is for indicating, how you like the whole idea of such attributes. Other items are specific for the individual attributes.

The VALUE attribute is designed to replace the DATA statement.

Visibility attributes PUBLIC and PRIVATE are for controlling the scope of the names being declared, when using the MODULE/USE facility and Internal procedure declarations.

The ARRAY attribute is self evident.

The ALLOCATABLE attribute is part of the dynamic heap memory allocation capability, associated with the ALLOCATE statement; see section 2.2 of FIB-1 for more information on allocatable arrays.

The OPTIONAL attribute is for support of optional arguments for functions and subroutines.

The IMPLICIT NONE statement, familiar to VAX FORTRAN users, has been added to FORTRAN 8X.

The REAL CHAR statement is supported to enable users to express floating point constants with a guaranteed precision and range. This is described in more detail in section 2.3 of FIB-1.

VI. Accessing Data Objects:

To a large extent, this section is devoted to accessing arrays, characters, and components of structures. The array features are summarized in FIB-1, section 2.2. The items in the ballot are separated to allow you to express preferences as to the level of array capability you need for your applications. The last item in this section of the ballot, titled "array section parent for substring", refers to the ability, allowed in FORTRAN 8X, to take a substring of a subsection of an array of character strings. While this is well defined, there is some fear of very low performance implementations, given the fact that the result is a set of discontinuous sections of discontinuous strings. Thus, this item is broken out in the ballot, so that you can express whether this feature is really needed.

Also in this section is the component selection feature of FORTRAN 8X. Other than the use of the "%" character rather than the ".", this is the same kind of thing available in other languages with structured data. The reason that "." cannot be used is that FORTRAN 8X allows user defined infix operators (monadic and dyadic) using the syntax ".name.". See FIB-1, section 2.5 for more information on this topic.

VII. Expressions and Assignment:

All the items in this section are described either in FIB-1, or the discussion above. They have been broken out to allow you to express your level of interest in these various capabilities.

VIII. Execution Control:

The items in this section are described in section 2.8 of FIB-1, except for Enable, Signaling, and Conditions, which are described above in the explanation under point 1.

IX. I/O statements:

The new VALUES keyword in control information lists in I/O statements standardizes the long needed feature, of determining how many values were transferred, when a variable length record is being read into a (larger) fixed size array. The NULLS keyword gives you similar information about a list directed read of an array, when only some of the elements are specified.

The POSITION, ACTION, and DELIM keywords in OPEN and INQUIRE statements are described in section 2.9 of FIB-1.

The PAD keyword in OPEN and INQUIRE statements allows the user to control, whether the run time system pads or issues an error, when a length mismatch occurs between the I/O list and the external record.

The IOLENGTH specifier is included to allow the user to optimize the I/O performance of a program, in a portable way, by INQUIREing about the maximum length record that can be written to a given file in a single operation.

X. FORMATS (Called Input/Output Editing in S8):

This area has few extensions beyond FORTRAN 77. Section 2.9 of FIB-1 sums up the situation succinctly. Since FIB-1 was written, the B edit descriptor has been added for supporting the bit data type.

The main item of interest is the inclusion, at long last, of a standard form of NAMELIST I/O. While NAMELIST I/O has been available in most vendor's implementations for some time, there are many variations in program and data syntax. This is in spite of its inclusion in a well known auxiliary FORTRAN standard. The FORTRAN 8X version is yet a different variation, which does not use a NAMELIST statement at all. Rather, it uses the specification `PMT=**` in the control information list, and allows NAMELIST input or output from the I/O list quantities.

XI. Program Units:

This topic is covered in sections 2.5 and 2.6 of FIB-1. As discussed above, the reason that INCLUDE has been put into this section of the ballot is the complexity and difficulty of the MODULE/USE feature.

The closest analogy to the MODULE/USE feature defined in FORTRAN 8X is the Package concept in the Ada language. It allows for data declarations, internal procedure declarations and other name declarations to be packaged up in one convenient place for easy reference. It allows for name mapping and selective visibility of names exported and imported to the MODULE; it does this by using the ONLY, ALL, EXCEPT, PRIVATE, and PUBLIC keywords.

Unlike Ada, however, FORTRAN 8X has no elaborate compilation library mechanism to insure both integrity and flexibility of the declarations. This leads to the problem, that in order to support MODULE/USE, the vendor needs to use one of 3 mechanisms:

- 1) An elaborate compilation library scheme, like Ada's. While vendors of large machines could certainly implement this, it would seem to decrease portability, because the low end vendors would have no incentive to use such an elaborate scheme.
- 2) Require that modules all be compiled at once. This requirement, like that of the PASCAL language, actually makes it harder to maintain large applications, not easier.
- 3) Invent a simple means of mapping MODULE names to file names, to perform something like an INCLUDE function. While this is probably a reasonable compromise between 1) and 2), and might work fine in a single user system environment, it certainly seems inadequate when working on large multi-module programs on computers with rich file systems like VAX, DECsystems, and PDP-11s running most DEC supported operating systems.

In contrast, the INCLUDE statement has the combined advantages of power, flexibility and widespread implementation and usage already. For this reason, the DEC representatives to X3J3 have consistently supported INCLUDE instead of MODULE/USE.

These are both included as ballot items, so that you may express your opinion on this issue.

Also under this heading is the topic of interface specifications. These allow strong type checking across procedure calls, even when separate compilation is used.

XII. Functions and Subroutines:

Most of the items in this chapter are described in section 2.6 of FIB-1, and they are available in some form in many common programming languages.

Recursive procedures are not hard to implement overall, given a stack based architecture, like that of the PDP-11 and VAX machines. However, there are potential problems for user programs, since many Subroutines and Functions take advantage of the "automatic" SAVE, that happens because local variables are statically allocated. Thus, new user programs using the RECURSIVE keyword would generally not have problems, while existing programs converted to this feature should expect them.

Keyword arguments are only allowed with interface specifications or simultaneous compilation of the calling and called routines, because they require, that the compiler map tie the keyword mechanism to a fixed argument order.

Alias (identified) array actual arguments require a descriptor passing mechanism (also called a dope vector). Since the called routine has no way of knowing whether its caller will pass an alias array or a base array, it must accommodate the more complex mechanism. Thus, there is the potential for upward compatibility problems with this feature, when mixing object code for FORTRAN 77 subroutines and functions with FORTRAN 8X calling programs. There is also a potential for performance degradation of existing user code with this feature, because FORTRAN 77 arrays, even when using passed length and assumed size features, are passed by the more efficient reference mechanism.

The user written elemental function feature was not described in FIB-1. This capability, proposed only recently, would allow user functions, written with certain constraints, to behave like certain array valued intrinsic functions when called with array arguments.

XIII. Intrinsic:

There are many more intrinsic functions in FORTRAN 8X than FORTRAN 77. Many of the new ones are described in FIB-1 and have to do with arrays and new numeric type specifiers. The environment intrinsics are described in section 2.3 of FIB-1. Many of the array intrinsics are described in section 2.2.

The terminology "elemental" or "transformational" is used to characterize array intrinsic functions. An elemental function is one that takes an array argument, applies a uniform operation to each argument, and returns an array of the same shape as its argument. A transformational function, by contrast, generally returns a result whose shape is different from the input argument, and whose values depend on interactions among the values of the input array.

FORTRAN 8X SIR BALLOT

DECUS membership number: _____

FORTRAN experience level: Wizard _____ Expert _____ Knowledgeable _____
 Normal _____ Novice _____

Our site uses the following operating systems: VAX/VMS _____ VAX/ELAN _____
 ULTRIX _____ UNIX(tm) _____
 RSX-11 _____ RSTS _____ RT-11 _____ IAS _____
 TOPS-10 _____ TOPS-20 _____
 OTHER _____

Type of Organization:	Employed as:
Scientific _____	Professional Programmer _____
Engineering _____	Problem Solver (occasional) _____
Commerical _____	Compiler Writer/Implementator _____
Government _____	Home Computer User _____
Vendor (computer) _____	Manager of Software _____
University _____	Other _____
Other _____	

Do you support the concept of deprecated features in FORTRAN 8X? YES _____ NO _____
 (Note: This does not count toward your total point count.)

(SIR balloting process is as follows: Users allocate points to items with 100 points total. Points can be awarded positively to encourage an item, or negatively, to discourage it; at most 10 points can be awarded on any one item. Absolute value of points are counted toward the 100 point total; thus, a +10 and a -10, use 20 points, not 0 points.)

Category codes:

BENEFIT CATEGORY:

- 3 - New run time capability, not available in existing languages*.
- 2 - New run time capability, found in one or more widely used existing languages.
- 1 - New compile time capability, not available in existing languages.
- 0 - New compile time capability, found in one or more existing languages.

* Note: The interpretive language APL is being excluded here from the list of "widely used existing languages". It has had most of the array features in FORTRAN 8X for many years. However, it has not proved sufficiently popular to attract large numbers of existing FORTRAN 77 users. Thus, a more FORTRAN like approach, such as the FORTRAN 8X array processing extensions, appears to be warranted.

IMPLEMENTATION DIFFICULTY (VAX specific, rough estimate):

- 3 - Already in VAX FORTRAN V4.
- 2 - Easy - Underlying mechanism exists; syntax or other easy work only.
- 1 - Moderate - Underlying mechanism needed, but is straightforward.
- 0 - Hard - Underlying mechanism new, difficult, or some special consideration.

PERFORMANCE OVERHEAD* (VAX specific, rough estimate):

- 3 - Uses existing mechanisms; performance level of VAX FORTRAN V4.
- 2 - Underlying mechanism new; but comparable performance to V4.
- 1 - New mechanism; may be slower than V4, but no degradation of existing code.
- 0 - New; slower than V4, and may degrade existing code performance.

* Note: For benefit categories 2 and 3, this refers to run time overhead (slowdown). For benefit categories 0 and 1, this refers to compile overhead (slowdown).

X3J3/S8 FEATURE (Benefit-Implementation Difficulty-Overhead) POINT
See category codes below for meaning

III. Lexical Elements:

	BENEFIT CATEGORY:	IMPLEMENTATION DIFF.	PERFORMANCE OVERHEAD	POINT
New source form items:				
Significant blanks	(0-0-0)			_____
Insignificant separator " _"	(0-0-0)			_____
Lines to 132 characters	(0-3-3)			_____
Column 1-6 usage removed	(0-2-3)			_____
"!" initiating comments	(0-3-3)			_____
Multiple stmts per line, ";" separator	(0-0-0)			_____
New continuation convention	(0-1-0)			_____

IV. Data Types:

Numeric type attributes (precision, range)	(2-2-3)			_____
CHARACTER constant syntax (allow " as well as ')	(0-2-3)			_____
Derived data types:				
Simple data structuring mechanism (like V4)	(2-3-3)			_____
Variant components	(2-2-2)			_____
Derived type parameterization (array dimensions, char length, precision attributes)	(0-0-1)			_____
Strong typing	(0-0-0)			_____
Derived type constants	(0-1-1)			_____
Bit data type	(2-0-1)			_____

V. Entity declaration and Specification Statements:

Declaration form:				
Attributes on type declarations	(0-2-1)			_____
Value attribute (replaces DATA stmt)	(0-2-2)			_____
Visibility attribute	(0-0-0)			_____
Array attribute	(0-2-3)			_____
Allocatable attribute	(2-0-0)			_____
Optional attribute	(2-0-0)			_____
IMPLICIT NONE statement	(0-3-3)			_____
REAL CHAR statement	(1-1-1)			_____

Category codes:

BENEFIT CATEGORY:

- 3 - Nev run time capability, not available in existing languages*.
- 2 - Nev run time capability, found in one or more widely used existing languages.
- 1 - Nev compile time capability, not available in existing languages.
- 0 - Nev compile time capability, found in one or more existing languages.

* Note: The interpretive language APL is being excluded here from the list of "widely used existing languages". It has had most of the array features in FORTRAN SX for many years. However, it has not proved sufficiently popular to attract large numbers of existing FORTRAN 77 users. Thus, a more FORTRAN like approach, such as the FORTRAN SX array processing extensions, appears to be warranted.

IMPLEMENTATION DIFFICULTY (VAX specific, rough estimate):

- 3 - Already in VAX FORTRAN V4.
- 2 - Easy - Underlying mechanism exists; syntax or other easy work only.
- 1 - Moderate - Underlying mechanism needed, but is straightforward.
- 0 - Hard - Underlying mechanism nev, difficult, or some special consideration.

PERFORMANCE OVERHEAD* (VAX specific, rough estimate):

- 3 - Uses existing mechanisms; performance level of VAX FORTRAN V4.
- 2 - Underlying mechanism nev; but comparable performance to V4.
- 1 - Nev mechanism; may be slower than V4, but no degradation of existing code.
- 0 - Nev; slower than V4, and may degrade existing code performance.

* Note: For benefit categories 2 and 3, this refers to run time overhead (slowdown). For benefit categories 0 and 1, this refers to compile overhead (slowdown).

VI. Accessing Data Objects:

New array features:

Whole array references, operations	(3-0-1)	_____
Array section references, operations	(3-0-1)	_____
Identified (alias) arrays	(3-0-0)	_____
Array valued constants	(1-1-1)	_____
Array valued expressions	(3-0-1)	_____

Vector valued sections

(3-0-0)	_____
---------	-------

Allocatable arrays:

Allocate and free statements	(2-1-0)	_____
Passing as arguments	(2-2-3)	_____
Returning a newly allocated array	(2-0-0)	_____
Array section parent for substring	(3-0-0)	_____
Component selection for derived types	(2-3-3)	_____

VII. Expressions and Assignment:

New bit expressions	(2-2-2)	_____
Overloaded operators	(0-0-0)	_____
Derived monadic and dyadic expressions	(0-0-0)	_____
Whole array assignment	(3-1-3)	_____
Masked array assignment (WHERE)	(3-1-1)	_____
Selected array element assignment (FORALL)	(3-1-1)	_____
Derived type operations (assignment, comparison, I/O, passing as arguments)	(2-2-3)	_____

VIII. Execution Control:

New DO construct	(0-2-3)	_____
CASE	(0-2-3)	_____
Enable, Signaling, conditions	(3-2-3)	_____

IX. I/O statements:

Control information list keywords:	(2-2-3)	_____
VALUES	(2-2-3)	_____
NULLS	(2-2-3)	_____
OPEN/INQUIRE keywords:	(2-2-3)	_____
POSITION	(2-2-3)	_____
ACTION	(3-2-3)	_____
DELIM	(2-2-3)	_____
PAD	(2-2-3)	_____
INQUIRE keyword:	(3-2-3)	_____
IOLength	(3-2-3)	_____

Category codes:

BENEFIT CATEGORY:

- 3 - New run time capability, not available in existing languages*.
- 2 - New run time capability, found in one or more widely used existing languages.
- 1 - New compile time capability, not available in existing languages.
- 0 - New compile time capability, found in one or more existing languages.

* Note: The interpretive language APL is being excluded here from the list of "widely used existing languages". It has had most of the array features in FORTRAN 8X for many years. However, it has not proved sufficiently popular to attract large numbers of existing FORTRAN 77 users. Thus, a more FORTRAN like approach, such as the FORTRAN 8X array processing extensions, appears to be warranted.

IMPLEMENTATION DIFFICULTY (VAX specific, rough estimate):

- 3 - Already in VAX FORTRAN V4.
- 2 - Easy - Underlying mechanism exists; syntax or other easy work only.
- 1 - Moderate - Underlying mechanism needed, but is straightforward.
- 0 - Hard - Underlying mechanism new, difficult, or some special consideration.

PERFORMANCE OVERHEAD* (VAX specific, rough estimate):

- 3 - Uses existing mechanisms; performance level of VAX FORTRAN V4.
- 2 - Underlying mechanism new; but comparable performance to V4.
- 1 - New mechanism; may be slower than V4, but no degradation of existing code.
- 0 - New; slower than V4, and may degrade existing code performance.

* Note: For benefit categories 2 and 3, this refers to run time overhead (slowdown). For benefit categories 0 and 1, this refers to compile overhead (slowdown).

X. FORMATS (Input/Output Editing):

Name directed	(2-2-3)	_____
(similar to but not the same as NAMELIST)		
B (bit) editing	(2-2-3)	_____

BENEFIT CATEGORY:
IMPLEMENTATION DIFF:
PERFORMANCE OVERHEAD:

XI. Program Units:

Interface specifications	(0-0-0)	_____
MODULE and USE mechanism:		
Allows declarations, nested scope	(0-0-0)	_____
Name scope control		
(ONLY, ALL, EXCEPT, PRIVATE, PUBLIC)	(0-0-0)	_____
Name mapping	(0-0-0)	_____
INCLUDE stmt (not in FORTRAN 8X now)	(0-3-3)	_____

XII. Functions and Subroutines:

Internal procedures	(0-1-1)	_____
Recursive procedures	(2-0-1)	_____
Argument intent	(3-2-3)	_____
Optional arguments	(2-2-2)	_____
Keyword arguments	(0-0-0)	_____
Alias array arguments	(3-0-0)	_____
User written array valued functions	(3-0-0)	_____
User written elemental functions	(3-0-0)	_____

XIII. Intrinsics:

Environmental intrinsics	(3-2-3)	_____
Array intrinsics:		
Elemental array functions	(3-0-2)	_____
Transformational array functions	(3-0-0)	_____

ABSOLUTE POINT TOTAL (maximum of 100 points) _____

Category codes:

BENEFIT CATEGORY:

- 3 - New run time capability, not available in existing languages*.
- 2 - New run time capability, found in one or more widely used existing languages.
- 1 - New compile time capability, not available in existing languages.
- 0 - New compile time capability, found in one or more existing languages.

* Note: The interpretive language APL is being excluded here from the list of "widely used existing languages". It has had most of the array features in FORTRAN 8X for many years. However, it has not proved sufficiently popular to attract large numbers of existing FORTRAN 77 users. Thus, a more FORTRAN like approach, such as the FORTRAN 8X array processing extensions, appears to be warranted.

IMPLEMENTATION DIFFICULTY (VAX specific, rough estimate):

- 3 - Already in VAX FORTRAN V4.
- 2 - Easy - Underlying mechanism exists; syntax or other easy work only.
- 1 - Moderate - Underlying mechanism needed, but is straightforward.
- 0 - Hard - Underlying mechanism new, difficult, or some special consideration.

PERFORMANCE OVERHEAD* (VAX specific, rough estimate):

- 3 - Uses existing mechanisms; performance level of VAX FORTRAN V4.
- 2 - Underlying mechanism new; but comparable performance to V4.
- 1 - New mechanism; may be slower than V4, but no degradation of existing code.
- 0 - New; slower than V4, and may degrade existing code performance.

* Note: For benefit categories 2 and 3, this refers to run time overhead (slowdown). For benefit categories 0 and 1, this refers to compile overhead (slowdown).

Mail to: **MAIL DEADLINE IS 23 OCTOBER 1985**
 Jay W. Wiley, 32-270
 Bechtel Power Corp.
 POB 60860, Terminal Annex
 Los Angeles, CA 90060

Status of Work Toward Revision of Programming Language Fortran

by
Jerrold L. Wagener
Amoco Production Company
Tulsa, Oklahoma

May 1984

Please direct comments or questions concerning the technical content of this report to: Jerrold L. Wagener; Amoco Research Center; PO Box 591; Tulsa OK 74012; (918) 660-3978

The Chairman of Fortran Standards Committee X3J3, Jeanne Adams, would like to call your attention to a series of Fortran Forum meetings to be held in the summer and fall 1983. The first two of these one day informational meetings have been scheduled for Wednesday, August 8, at E G and G, Idaho Falls; and for Monday, August 13, at Colorado State University, Fort Collins. More detailed announcements of these Forums appear on pages 43 and 45. Additional information can also be obtained by calling Jeanne Adams at (303) 491-7596. Jeanne would also like to hear from other organizations that would be interested in acting as host for a Fortran Forum.

Observers are welcome to attend regular meetings of X3J3. Future meetings of the committee are scheduled for 6 to 11 Aug 1984 in Jackson WY; 12 to 16 Nov 1984 in Fort Lauderdale FL; February 1985 in northern California; and May 1985 in Champaign-Urbana IL. Because of limitations on meeting space, anyone who would like to attend an X3J3 meeting should request further information in advance from the X3J3 Vice Chairman, Martin Greenfield, at (617) 671-2912. International meetings are also planned for 1 to 5 July 1985 in Germany; and 8 to 12 July 1985 in England.

PLEASE COMPLETE THE SURVEY on pages 2, 3, and 4.

FORTRAN 8x SURVEY
FORTRAN STANDARDS COMMITTEE X3J3
Jeanne Adams, Chair, X3J3

This questionnaire has been developed to survey the opinions of the user community on the new features and the architecture proposed by X3J3 for inclusion in the next draft FORTRAN standard. The results of this and other surveys and questionnaires will be used by the X3J3 committee in assessing the strength of each new facility for inclusion in a draft standard. Information about the new features is contained in "FORTRAN Information Bulletin", Number 1. Return your questionnaire to:

Andrew Johnson
 MS 10C17-3
 Prime Computer Inc.
 500 Old Connecticut Path
 Framingham, Ma 01701

RESPONDENT PROFILE

Check those that apply.

Type of Organization:	Are you employed as:
Scientific _____	Professional Programmer _____
Engineering _____	Problem Solver (Occasional) _____
Commercial _____	Compiler Writer/Implementor _____
Government _____	Home Computer User _____
Vendor (Computer) _____	Manager of Software _____
University _____	Other _____
Other (state) _____	

Was FORTRAN your first computer language? Yes ___ No ___

What is your "language of choice?" _____

Name the vendor and operating system currently used.

Do you now use: FORTRAN 66 _____ FORTRAN 77 _____ 77 Subset _____

Do you subscribe to: FORTEC _____ SIGPLAN _____ SIGNUM _____

Have you seen the FORTRAN Information Bulletin summarizing the proposals currently under consideration for FORTRAN 8x? Yes ___ No ___

List two features for FORTRAN 8X that--

You want in FORTRAN:	You do not want in FORTRAN:
1. _____	1. _____
2. _____	2. _____

=====

Name _____ Address _____

Tel. No. _____

CHECKLIST ON FORTRAN 8x FEATURES

	Strongly Approve	Approve	Do not Approve	Strongly Disapprove	Do Not
Array Processing Features					
WHERE Construct					
Data Structures					
New Source Form(free)					
Significant Blanks					
Recursion					
Precision Specification					
Environmental Intrinsic					
Loop Constructs					
Enhanced CALL					
CASE Construct					
Internal Procedures					
IMPLICIT NONE					
Entity Oriented Decl.					
Derived Data Type					
MODULES					
Event Handling					
Varying CHARACTER Length (Note it has been removed)					
Bit Data Type (Note that it has been removed)					
Macros (Note that it has been removed)					
Architecture with deprecated Features (next page)					
=====					
General Impression of FORTRAN 8x					

DEPRECATED FEATURES

Which of these features would you make a candidate for deletion in the 199X FORTRAN standard, note FORTRAN 9x. All of these features will be retained in 198x. Do you want these removed from 199x, retained in 199x or are you undecided? Check one.

	Removed from 199x	Retained in 199x	Unde- cided
FORTRAN 77 Fixed Position Source From	_____	_____	_____
Alternate RETURN	_____	_____	_____
Assumed Size Dummy Arrays	_____	_____	_____
Passing Scalar to Dummy Array	_____	_____	_____
Specific Names (not Generic) for Intrinsic	_____	_____	_____
Statement Functions	_____	_____	_____
BLOCK DATA Program Unit	_____	_____	_____
Arithmetic IF Statement	_____	_____	_____
ASSIGN and Assigned GO TO Statements	_____	_____	_____
COMMON Statement	_____	_____	_____
Computed GO TO Statement	_____	_____	_____
DATA Statement	_____	_____	_____
DIMENSION Statement	_____	_____	_____
DOUBLE PRECISION Data Type	_____	_____	_____
ENTRY Statement	_____	_____	_____
EQUIVALENCE Statement	_____	_____	_____
FORTRAN 77 DO Statement	_____	_____	_____
PAUSE Statement	_____	_____	_____

Please include any comments you wish to make concerning FORTRAN 8x features and features marked as candidates for deletion from FORTRAN 9x on the reverse side of this page.

Status of Work Toward Revision of Programming Language Fortran

by
Jerrold L. Wagener
Amoco Production Company
Tulsa, Oklahoma

May 1984

TABLE OF CONTENTS

1. Introduction	6
1.1 Array Operations	
1.2 Numerical Computation	
1.3 Derived Data Types	
1.4 Modular Definitions	
1.5 Deprecated Features	
2. Language Summary	9
2.1 Full Language Overview	
2.2 Array Processing	
2.3 Numeric Data Type	
2.4 Derived Data Types	
2.5 Modules	
2.6 Procedures	
2.7 Program Source Form	
2.8 Control Constructs	
2.9 Input/Output	
2.10 Event Handling	
2.11 Other Features	
3. Language Architecture	33
3.1 The Core	
3.2 Modules	
3.3 Deprecated Features	
3.3.1 Summary of Deprecated Features	
3.3.2 Storage Association	
3.3.3 Redundant Functionality	
3.3.4 Other Deprecated Features	

This report describes the current status of the technical work of X3J3 since the adoption of X3.9-1978 (Fortran 77). This work, informally referred to as "Fortran 8x", is incomplete and tentative, and is subject (and likely) to change prior to issuing a draft proposed standard (expected to occur no earlier than 1985). The purpose of this report is to summarize the status of current work by X3J3 relative to a future revision of the current Fortran standard. A list of criteria for this revision is summarized in section 3.1 of this document. Comments on any and all aspects of the features described in this report are welcomed.

1. INTRODUCTION

Among the additions to Fortran 77 contemplated for the next Fortran standard, five stand out as the major ones:

array operations

improved facilities for numerical computation

programmer defined data types

facilities for modular data and procedure definitions

the concept of "deprecated" features

A number of other additions are also included in Fortran 8x, such as improved source form facilities, more control constructs, recursion, dynamically allocatable arrays of any size, and event handling.

No Fortran 77 features will be removed; it remains X3J3's intent that any standard-conforming Fortran 77 program will be a standard-conforming Fortran 8x program, and that, with any exceptions clearly listed in the document, new Fortran 8x features can be compatibly incorporated into such programs.

1.1 Array Operations

Computation involving large arrays is an extremely important part of engineering and scientific uses of computing. Arrays may be used as atomic entities in Fortran 8x, and operations for processing whole arrays and sub-arrays (array sections) are included in the language for two principal reasons: (1) these features provide a more concise and higher level language that will allow programmers to more quickly and reliably develop and maintain scientific/engineering applications; (2) these features can significantly facilitate optimization of array operations on all computer architectures.

The Fortran 77 arithmetic, logical, and character operations and intrinsic functions are extended to operate on array-valued operands. These include

whole, partial, and masked array assignment, array-valued constants and expressions, facilities to define user-supplied array-valued functions, and new intrinsic functions to manipulate arrays, extract general sections, and to support extended computational capabilities involving arrays (e.g., array reduction).

1.2 Numerical Computation

Scientific computation is one of the principal application domains of Fortran, and the guiding objective for all of the technical work is to strengthen Fortran as a vehicle for implementing scientific software. Though nonnumeric computations are increasing dramatically in scientific applications (and a number of the tentative additions to Fortran reflect that trend), numeric computation remains the workhorse. Accordingly, proposed additions include portable control over numeric precision specification, inquiry as to the characteristics of numeric information representation, and improved control of the performance of numerical programs.

1.3 Derived Data Types

"Derived data type" is the term given to that set of features in Fortran 8x that allows the programmer and package writer to define arbitrary data structures and operations on them. Data structures are user-defined aggregations of intrinsic and derived data type fields. Intrinsic operations on structured objects include comparison, assignment, input/output, and use as procedure arguments. The derived data type facilities may be used, without further operation definition, as a simple data structuring mechanism. With additional operation definitions, derived data types provide an effective implementation mechanism for data abstractions.

Procedure definitions in Fortran 8x may appear internally in a program unit, and as such may be used to define operations on derived data types. Internal procedures take essentially the same form as external procedures, with additional provisions for their use as infix operators. New operator symbols may be defined, and the intrinsic operator symbols may be overloaded for use with new data types. Internal procedures may be used simply as a mechanism for defining procedure packages (whether or not new data types are involved), and may be used for new operations on objects of intrinsic data types.

1.4 Modular Definitions

There is no way in Fortran 77 to define a global data area in one place and have all the program units in an application use that definition. In addition, the ENTRY statement is awkward and restrictive for implementing a related set of procedures, possibly involving common data objects. And finally there is no means in Fortran by which procedure definitions, especially interface information, may be made known locally to a program unit. All of these deficiencies, and more, are remedied by a new type of program unit that may contain any combination of data element declarations, derived data type definitions, procedure definitions, and procedure interface information. This program unit, called a MODULE, may be considered to be a generalization of and replacement for the BLOCK DATA program unit. A module may be referenced by any program unit, thereby making the module contents available to that program unit. This provides vastly improved facilities for defining global data areas and procedure packages. It also provides a convenient mechanism for encapsulating derived data type definitions (including operations defined on them), i.e., for encapsulating data abstractions.

1.5 Deprecated Features

With the advent of superior facilities, the use of certain older features of Fortran should be discouraged, and some of these features should possibly eventually be phased out of the language. For example, the numeric facilities alluded to above provide the functionality of DOUBLE PRECISION; with the new array facilities non-conformable argument association (such as passing an array element to a dummy array) is unnecessary (and in fact is not useful as an array operation); BLOCK DATA units are obviously redundant and inferior to modules. It is the current intent to identify such "superseded" facilities as deprecated (according to Webster: "mild or regretful disapproval; lower estimated value") features. Deprecated features will remain part of Fortran 8x; it is the intent that complete upward compatibility be maintained between Fortran 77 and Fortran 8x. Deprecated features may, however, be candidates for removal from the version of the Fortran standard following Fortran 8x. It is the intent that in this way official notice is given many years prior to removing a feature from the standard language. In Section 3.3 below, the proposed deprecated features are identified, together with possible functional replacements.

2. LANGUAGE SUMMARY

In this section a summary of proposed Fortran 8x features is given, with emphasis upon new features. The description uses a form of BNF slightly modified from that used in the Fortran 77 standard (X3.9-1978). Metasymbols are italicized, and comprise

is introduces a syntactic class definition
or introduces a syntactic class alternative
[] encloses an optional item
[]... encloses an optionally repeated item

The principal difference between this form of BNF and that in X3.9-1978 is that each definition starts with the name of the syntactic class that is being defined; the actual definition follows the metasymbol *is* or *or*. In X3.9-1978, only the right-hand side appeared, with informal text introducing the class being defined. Lower-case words, including hyphenated words, are syntactic class names. Upper-case is used for words that actually appear in the Fortran code (terminal symbols), even though lower-case is also allowed in the new source form. Terminal symbols that may have imbedded blanks (e.g., ENDIF and END IF) are shown in only one form. There is no attempt to be completely comprehensive in this summary, especially with Fortran 77 features. The syntax should be considered illustrative and incomplete; in the interest of brevity, some syntactic items (e.g., integer-expr) are not defined, and hopefully are adequately clear from context. Both syntax and semantics are subject to change prior to issuing a draft standard.

Each of the following sub-sections has the three part form of (1) a general discussion of the particular topic of the sub-section, (2) the BNF descriptions of the features under discussion, and (3) a set of notes and examples that provide specific points of information. Where new intrinsic functions are listed in the syntax only the function names are given (not the argument lists); following the names are brief descriptions of the functionalities.

2.1 Full Language Overview

The following highly condensed summary of the full language is provided so that one can see at a glance the major components of proposed Fortran 8x. So that it is clear which are the proposed new features, these are italicized (along with the metasymbols) and placed first (or occasionally last) in a list of syntactic class alternatives. Unitalicized items are essentially unchanged from Fortran 77, and are not further described in this information bulletin. The new features are all described in greater detail in the succeeding sections. The deprecated Fortran 77 features are last in any list of syntactic class alternatives. So that it is clear in this section which features are the deprecated ones, *ob* is used in place of *or* in these cases. (The italicizing

of new features, and the use of *ob*, occurs only in this section on full language overview.)

program-unit *is* unit-heading
 [use-statement]...
 [local-specification]...
 [executable-construct]...
 [internal-procedure]...
 END [unit-type [unit-name]]

unit-heading *is* *MODULE* module-name
 or *HANDLER* handler-name
 or [PROGRAM program-name]
 or function-heading
 or subroutine-heading
 ob BLOCK DATA identifier

local-specification *is* type-definition
 or procedure-interface
 or declare-statement
 or REFER refer-name (attribute [,attribute]...)
 or type-declaration
 or IMPLICIT implicit-list
 or PARAMETER (constant-definition-list)
 or SAVE [save-list]
 or INTRINSIC procedure-name-list
 or EXTERNAL procedure-name-list
 or FORMAT (format-specification)
 ob COMMON [/common-block-name/] common-list
 ob DATA initial-value-list
 ob DIMENSION dimension-list
 ob EQUIVALENCE equivalence-list
 ob statement-function

declare-statement *is* identifier [,identifier]... : attribute [,attribute]...

attribute *is* non-char-type
 or intent-attribute
 or optional-attribute
 or REFER (refer-name)
 or CHARACTER [(length)]
 or DIMENSION (dimensions)
 or INITIAL (constant-expr)
 or CONSTANT (constant-expr)
 or SAVE

or INTRINSIC
 or EXTERNAL
 type-declaration *is* CHARACTER[*len] var-decl[*len] [,var-decl[*len]]...
 or non-char-type var-decl [,var-decl]...
 non-char-type *is* floating-point-data-type
 or TYPE (type-name)
 or EVENTMARK
 or INTEGER
 or LOGICAL
 ob DOUBLE PRECISION
 var-decl *is* identifier [(dimensions)]
 executable-construct *is* block-case
 or block-do
 or block-if
 or basic-statement
 ob fortran77-do-loop
 basic-statement *is* array-assignment
 or identify-statement
 or event-handling-statement
 or EXIT [construct-name]
 or CYCLE [construct-name]
 or assignable-object = expr
 or io-stmt
 or GO TO label
 or CALL subroutine-name[(actual-argument-list)]
 or CONTINUE
 or RETURN [return-code]
 or STOP [stop-code]
 or IF (scalar-logical-expr) basic-statement
 ob IF (arithmetic-expr) branch-list
 ob ASSIGN label TO integer-variable
 ob GO TO integer-variable [(label-list)]
 ob GO TO (label-list), integer-variable
 ob PAUSE [pause-code]
 assignable-object *is* [derived-data-type-object %]... object
 object *is* array-object
 or scalar-object
 expr *is* [unary-operator] basic-expr

basic-expr *is* value
 or constant-name
 or assignable-object
 or function-name [(actual-arg-list)]
 or basic-expr binary-operator basic-expr
 or (expr)

Notes -

- Statement labels are not indicated in this summary; any statement may be labelled.
- The declare-statement allows "entity-oriented" declarations, where the entity name is given first, followed by a list of its attributes. This style of declaration complements the "attribute-oriented" declaration style of Fortran 77.
- The ENTRY statement is deprecated and not listed here. It is used to partition the local-specifications and executable-constructs in a program unit; internal-procedures provide equivalent functionality.
- The MODULE program-unit may not contain a block of executable-constructs (except as internal-procedure bodies) (see Section 2.5 below).
- BLOCK DATA program-units may contain only local-specifications.
- Note the END statement extensions, even though they are not italicized.
- Procedure-headings have minor extensions, even though they are not italicized (see Section 2.6 below).
- Note that, because an assignable-object may be an array or array section, expressions and assignments are considerably extended, even though they are not italicized (see Section 2.2 following).
- Restrictions on ordering of local-specifications are not shown here.
- The EXIT and CYCLE statements may only appear within block-do constructs; the RETURN statement may only appear within (internal and external) function and subroutine subprograms.

- The REFER statement associates a refer-name with a collection of attributes. The REFER (refer-name) attribute may be used in subsequent attribute lists to represent the collection.

2.2 Array Processing

The proposed array facilities view whole arrays and array sections as atomic computational objects from which expressions can be composed and to which assignment can be made. That is, a basic aim of array processing is to operate where possible directly on the arrays themselves rather than on their elements individually. All scalar operations are extended to conformable arrays, operating on them in an element-by-element fashion. Both user-written and intrinsic functions may return array values. Operations on whole arrays are thus made available in functional form.

The ability to manage and control storage of arrays has been significantly enhanced by addition of the following features: (1) automatic arrays (local arrays with variable dimensions) are created on entry to a procedure and destroyed on return; (2) allocatable arrays are created by execution of an ALLOCATE statement, and are destroyed by execution of a FREE statement or by return from the procedure in which they are created; (3) assumed shape arrays (dummy arrays) have implicit dimension information passed during a call, somewhat analogously to the passing of length information to Fortran 77 CHARACTER*(*) dummy arguments.

```
array-object      is array-name
                  or array-section [(substring-range)]

array-section     is array-name (subscript-range[,subscript-range]...)

subscript-range  is integer-expr
                  or [integer-expr]:[integer-expr][: integer-expr]
                  or one-dim-integer-array-expr

substring-range  is [integer-expr]:[integer-expr]

array-constructor is [ constructor-value [,constructor-value]... ]
                    (note: outer brackets part of constructor)

constructor-value is scalar-expr
                  or integer-expr : integer-expr [: integer-expr]
                  or repeat-factor array-constructor

repeat-factor    is integer-expr
```

```
array-assignment is array-object = array-expr
                  or WHERE (array-logical-expr) array-object = array-expr
                  or WHERE (array-logical-expr)
                      [array-object = array-expr]...
                      [OTHERWISE
                      [array-object = array-expr]... ]
                  END WHERE

                  or FOR ALL (index-range [,index-range]... [,logical-expr])
                      assignment-statement

index-range      is integer-variable = integer-expr:integer-expr[:integer-expr]

identify-statement is IDENTIFY <range> virtual-array = parent-array

range            is [lower-bound:] upper bound [, [lower-bound:]upper-bound]...

virtual-array    is array-name (integer-variable [,integer-variable]...)

parent-array     is array-name ( linear-mapping [,linear-mapping]... )

array-intrinsic-function is arithmetic-logical-reduction-function
                          or inquiry-function
                          or construction-function
                          or manipulation-function
                          or geometric-location-function
                          or MATHMUL      matrix multiplication

arithmetic-logical-reduction-function is SUM          sum all elements of an array
                                          or PRODUCT      product of all elements
                                          or MAXVAL        maximum value in an array
                                          or MINVAL        minimum value in an array
                                          or COUNT        number of true values in an array
                                          or ANY          true if any value in array is true
                                          or ALL          true if all values are true
                                          or DOTPRODUCT    dot product of two arrays

inquiry-function is RANK          rank of argument array
                  or EXTENT       size of array in each dimension
                  or SIZE         total number of elements in array
                  or LBOUND       lower bound in each dimension
                  or UBOUND       upper bound in each dimension

construction-function is SPREAD     replicates array by increasing rank
                  or REPLICATE    replicates by increasing extent
                  or MERGE        merges two arrays, using a mask
```

	or DIAGONAL	creates a diagonal array
	or PACK	packs array into a vector, with mask
	or UNPACK	inverse of PACK
	or SHAPE	changes the shape of a given array
manipulation- function	/s CSHIFT	circular shift of argument array
	or EOSHIFT	end-off shift of argument array
	or TRANSPOSE	matrix transpose of argument array
geometric- location- function	/s FIRSTLOC	locate first true element
	or LASTLOC	locate last true element
	or PROJECT	select masked values from array

Notes and Examples -

- Arrays of zero size are permitted.
- Two arrays are conformable if they are the same shape (same rank and same extent in each dimension).
- In executing array assignment statements the entire right hand side is evaluated before any assignment is made to the target array. This is significant when the same array (or sections of the same array) appears on both sides of the assignment operator.
- Element-by-element computation means that the operation takes place many times (all logically in parallel), once for each pair of corresponding elements of the operands, and the result is an array conformable with the operands. For example,

```
real A(100), B(100)
A = A + B*3.0 + sin(B)
```

performs $A(I) = A(I) + B(I)*3.0 + \sin(B(I))$ for all I between 1 and 100. Arithmetic, logical, and character operators, and scalar intrinsic functions operate in this manner.

- Whole array operations may be masked by FORALL and WHERE statements (and WHERE blocks), avoiding computation on certain elements and leaving portions of the target array unchanged:

```
real A(100,100), B(100,100), THRESHOLD
where (B.ne.0) A = A/B ! avoid zero-divide
where (A.gt.THRESHOLD) A = THRESHOLD ! flatten peaks
forall (I=1:100,J=1:100,I.ne.J) A(I,J) = 0.0 ! keep diag
```

- Array-valued functions are analogous to scalar-valued functions:

```
real function CONCAT(A,B)
real A(:), B(:), CONCAT(size(A)+size(B))
CONCAT(1:size(A)) = A
CONCAT(size(A)+1:) = B
end function CONCAT
```

- An array-constructor forms a one-dimensional array from scalar values (note that array construction employs square brackets as delimiter symbols). This may be "shaped" into multi-dimensional arrays with the SHAPE intrinsic function. One use of constructors is to create array constants:

```
integer A(10)
A = [1,2,3,4,5,0,0,0,0,0] ! or A = [1:5,5{0}]
```

- Generally wherever a whole array may be used an array section may be used. Examples of array sections of REAL A(8,8), B(10) are:

```
B(5:1:-2) ! elements B(5), B(3), B(1)
A(1:7:2,2:8:2) ! one color
A(2:8:2,1:7:2) ! on a checkerboard
A(1:4,:) ! upper half of A
```

- Sub-arrays and general sections may be selected and given new "virtual array" names by using the IDENTIFY statement:

```
real C(1:100,1:100), D(1000)
identify <1:100> DIAG(I) = C(I,I)
```

Since IDENTIFY is an executable statement, the virtual array definition can be changed dynamically during execution:

```
identify <1:1000> DIAG(I) = D(1001-I)
```

- Example of the use of assumed-shape arrays:

```
function XYZ (A,B)
real A(:,,:), B(-1:5:)
```

In Fortran 77 this would have to be accomplished with something like:

```
FUNCTION XYZ (A,B,I,J,K,L)
REAL A(I,J), B(-1:K,5:L)
```

- The concepts of reduction, construction, manipulation, and

geometric-location correctly suggest that the array intrinsic functions provide general operations on multidimensional arrays. Only the functions MATMUL, TRANSPOSE, and DOTPRODUCT are specialized to matrix/vector operations. Following are some examples using array intrinsic functions, assuming the declarations:

```
real X(N), Y(M), T(M,N), E(M,N)
complex C(N,N)
```

Note that some operations may be masked (mask=), dim=1 specifies operations on columns, and dim=2 specifies operations on rows.

$$\sum_{i=1}^N x_i = \text{sum}(X, \text{mask}=X.\text{gt}.0.1)$$

$$\sum_{i=1}^N (x_i - \bar{x})^2 = \text{sum}((X - \text{sum}(X)/N)**2)$$

$$\max_{1 \leq i \leq N} \left\{ \frac{\sum_j |c_{ij}| x_j}{x_i} \right\} = \text{maxval}(\text{matmul}(\text{abs}(C), X)/X)$$

radii $x_i = \sum_j |c_{ij}|$ of Gershgorin's circles, $i=1, N$

$X = \text{sum}(\text{abs}(C), \text{mask}=\text{not}.\text{diagonal}(. \text{true}., N), \text{dim}=2)$

chi-squared statistic $\chi^2 = \sum_{i,j} \frac{(t_{ij} - e_{ij})^2}{e_{ij}}$ where $e_{ij} = (\sum_k t_{ik}) * (\sum_k t_{kj}) / (\sum_k t_{kk})$

```
X = sum(T, dim=1)           ! column sums
Y = sum(T, dim=2)         ! row sums
E = spread(Y, 2, N) * spread(X, 1, M) / sum(T) ! outer product
CHI_SQ = sum((T-E)**2/E)
```

- Examples of queries without loops or conditional code on:

```
real T(M,N) ! test scores, M students with N tests
```

```
top score for each student:  maxval(T, dim=2)
no. scores above average:  count(T.gt.sum(T)/size(T))
lowest score above average: minval(T, mask=T.gt.sum(T)/size(T))
any student all above average? any(all(T.gt.sum(T)/size(T), dim=2))
```

2.3 Numeric Data Type

These facilities give the programmer portable control over the specification of real and complex data objects; they extend the REAL and COMPLEX type statements. The environmental-intrinsic-functions provide information that models the processor-supplied numeric system. They provide a portable means for adapting numeric algorithms to various arithmetic environments.

floating-point-data-type	is	REAL [(float-point-attr[, float-point-attr])]
	or	COMPLEX [(float-point-attr[, float-point-attr])]
float-point-attr	is	[PRECISION10=] attribute-value
	or	[EXP_RANGE=] attribute-value
attribute-value	is	integer-constant-expr
	or	*
real-constant-char	is	REAL_CHAR [(float-point-attr[, float-point-attr])] letter
floating-point-inquiry-function	is	ACTUAL_PREC (floating-point-argument)
	or	ACTUAL_EXP_RANGE (floating-point-argument)
environmental-intrinsic-function	is	RADIX model base for numbers
	or	PRECISION number of significant digits
	or	MINEXP smallest exponent value
	or	MAXEXP largest exponent value
	or	HUGE largest number of argument type
	or	TINY smallest positive number
	or	EPSILON positive number small relative to 1.0
	or	EXPONENT exponent value of argument
	or	SCALE scale argument by power of base
	or	NEAREST different value nearest to argument
	or	FRACTION fractional part of argument
	or	SETEXPONENT specify exponent part of argument
	or	ABSSPACE absolute spacing of numbers near argument
	or	RECSPACE reciprocal of relative spacing

Notes and Examples -

- The floating point attributes provide specifications for minimum numeric properties of the processor-supplied floating point system used to implement the relevant data objects.
- The integer constant expressions in the floating point attributes may reference the functions ACTUAL_PREC and ACTUAL_EXP_RANGE, which return the actual precision and actual exponent range for the given datum.

- In any given floating-point-data-type definition each float-point-attr alternative may appear at most once; if the float-point-attr keywords are not used, these attributes must appear in the order listed in the above definition of float-point-attr.
- A type specification with an attribute of, for example, REAL(10), for an entity X is a requirement that X be represented by a processor data type that has at least 10 decimal digits of precision. The uses of X in expressions, say, determines the precision of the arithmetic operations. Such a specification provides a portable method of indicating that the algorithm requires at least 10 decimal digits of precision.

- The function value ABSSPACE(X) for example can be used to terminate an iteration in a portable way by requiring that the absolute relative difference between two iterates, X and Y, is less than ABSSPACE(X) - that is, terminates the iteration when `abs(X-Y).le.absspace(X)`:

```
do; if (abs(X-Y).le.absspace(X)) exit
...
repeat
```

- A common difficulty with transporting numerical software from machine to machine is the difficulty with changes of precision, say from single precision on one machine to double precision on another. This problem particularly arises for the conversion of constants, which are typically spread throughout a program and cannot be easily converted when the precision of the data types is changed. For instance, consider the constant 1.1 which cannot be represented exactly on a binary (or hexadecimal) machine. When changing from single precision to double precision the constant 1.1 must be rewritten in the program as 1.1D0 in order to obtain the double precision value of this datum. Using the REAL_CHAR specification to specify an exponent character, the precision of all constants that use this exponent character can be readily changed.
- Another common problem in writing portable software is the safe and accurate range conversion of floating point variables to specified forms. For example, to determine the square root of a number X, the usual Newton's iteration converges too slowly to be a viable algorithm if the initial iterate is too far from the square root. Using the intrinsic functions FRACTION and EXPONENT, the interval over which the iterates can range can be drastically reduced, from which the square root can be efficiently computed.

From this result and EXPONENT(X) the square root of X can be formed using the SETEXPONENT intrinsic function. Thus, the program would look like:

```
! Perform the range reduction on X, assuming X is positive.
IEXP = exponent(X)
F = fraction(X)
if (mod(IEXP,2).eq.1) then; IEXP=IEXP+1; F=F/radix(X); endif
! Now X = F*radix(X)**IEXP, IEXP is even, and the square root
! of F is in the interval [1/radix(X)**2,1]. Now find SF, the
! square root of F, and reconstruct the answer from SF and IEXP:
ANSWER = setexponent(SF,IEXP/2)
```

2.4 Derived Data Types

The structure of a derived data type is a programmer-defined aggregation of fields, each field being a data element of primitive type (or another derived data type). The aggregation pattern may be fixed or variant (that is, partially dependent on one of the prior fields). A derived data type is defined with a TYPE construct; variables of this type may be declared in the normal manner. Intrinsic operations on derived data objects are assignment, equality comparison, input/output, and use as procedure arguments. These intrinsic operations may be augmented with additional programmer-defined operations. The structure qualification symbol (for referencing a component of a structured object) is the percent sign (%).

```
type-definition      /s TYPE type-name
                    {field-declaration}...
                    {variant-case-block}
                    END TYPE [type-name]

variant-case-block   /s SELECT CASE (tag-field-name)
                    { CASE case-selector
                      [ field-declaration]... }...
                    [ variant-case-block ]
                    END SELECT

field-declaration    /s type-declaration
                    or declare-statement

component-selection  /s structured-object-name % field-name
```

type-constructor *is* type-name (expr[,expr]...)
type-constant *is* type-name (constant-expr[,constant-expr]...)

Notes and Examples -

- Functions may return derived data type values.
- The following definition of PLOT_OBJECT defines a data structure made up of two fields of type real (the location in two-space), one field indicating the nature of the object, and finally the object data itself. In this case the object may be either a point symbol or a line segment.

```

type PLOT_OBJECT
  X0,Y0: real           ! object location
  CODE: integer        ! type of object
  select case (CODE)
    case (1); SYMBOL: character ! point symbol
    case (2); X1,Y1: real       ! line segment
  end select
end type PLOT_OBJECT

```

Let P,Q: type (PLOT_OBJECT) declare two variables P and Q.
Then the following expressions are examples of valid operations:

```

( P%X0 + Q%X0 ) / 2      ! mid-point of P and Q along X
if (P%CODE.eq.Q%CODE) then ! compare P and Q CODE fields
P = PLOT_OBJECT(0.,0.,1,"+") ! give P an initial value
read *, Q                ! input value of Q

```

2.5 Modules

Modules are proposed program units for the packaging of data type definitions, data object declarations, procedure definitions, and procedure interfaces. Modules are not themselves executable. A module that contains only data type definitions and data object declarations serves as a global data module. One that contains only procedure definitions and interfaces serves as a procedure library. One that contains data structure definitions and operations (internal procedures) on them serves as a data abstraction mechanism. A module may serve as a language extension mechanism by containing new operation definitions on intrinsic data types.

The use-statement "attaches" the specified module to the using program unit, thereby making the public definitions and declarations in the module available to the program unit. The use-statement without the module-name is for internal procedures specifying access to its host's definitions (note that in the absence of a use-statement specifying otherwise, all host definitions are automatically available to an internal procedure).

Through options on the use-statement, a using program may limit its access to definitions in the module and may rename the entities it uses. On the other hand, the writer of a module has complete control over which module entities are public (visible to using programs) and which are private; in the absence of explicit specification to the contrary, the default is public. (The syntax for specifying public and private is not shown, nor is the statement for specifying that the default visibility is private instead of public.)

```

use-statement       is USE[/module-name/] ONLY (access[,access]...)
                    or USE[/module-name/] [ALL[(rename[,rename]...)]
                                          [EXCEPT(identifier[,identifier]...)]]

access              is identifier
                    or rename

rename              is identifier = identifier

```

Notes and Examples -

- An example of the use of modules for defining global data pools may be found in Section 3.3.2.4 below.
- A scheme for using modules to encapsulate procedure libraries may be found in Section 3.3.2.5 below.
- The following is an example of the use of modules as a mechanism for encapsulating data abstractions; in this case the abstract data type is PLOT_OBJECT, defined above, with two defined operations CONNECT and BISECT:

```

module PLOT_MODULE

! insert definition of type PLOT_OBJECT from previous section

! define an operation to connect two plot_objects with a
! line segment; CONNECT returns a PLOT_OBJECT value that is
! a line segment (CODE=2) connecting two given plot_objects

internal type(PLOT_OBJECT) function CONNECT(P,Q) operator(//)

```

```

CONNECT%CODE = 2
CONNECT%X0 = P%X0
CONNECT%Y0 = P%Y0
CONNECT%X1 = Q%X0
CONNECT%Y1 = Q%Y0

```

P,Q: type(PLOT_OBJECT)

```

! finished if P is a point symbol,
! otherwise use tail (X1,Y1) of P
! as the head (X0,Y0) of CONNECT
if (P%CODE .eq.2) then
CONNECT%X0 = P%X1
CONNECT%Y0 = P%Y1
end if
end internal function CONNECT

! note that two plot_objects, A and B, can be connected
! using the infix operator notation A // B

! define an operation to bisect a line segment, with an "x";
! BISECT returns a point symbol PLOT_OBJECT value

internal type(PLOT_OBJECT) function BISECT(P)
P: type(PLOT_OBJECT)
BISECT = P ! return P itself if P is not a line segment
if (P%CODE.eq.2) then
BISECT%CODE = 1
BISECT%SYMBOL = "x"
BISECT%X0 = (P%X0+P%X1)/2
BISECT%Y0 = (P%Y0+P%Y1)/2
end if
end internal function BISECT

end module PLOT_MODULE

Any program unit may employ these definitions if it contains
the statement:
use /PLOT_MODULE/

```

2.6 Procedures

Procedures are allowed to be called with keyword actual arguments, called with optional arguments, defined with argument intent (e.g., input only), and called recursively. A keyword actual-argument is of the form KEYWORD=actual-argument, where KEYWORD is a dummy-argument-name. This has three advan-

tages: (1) if dummy-argument name is wisely chosen, then keywording effectively increases readability of the actual-argument-list, (2) keyworded actual-arguments may be placed in any order, which eliminates order errors (such as transposing two adjacent arguments) in actual-argument-lists, and (3) arguments not needed on a particular call may be omitted.

Procedure interface information may be provided for any external or dummy procedure, including procedures defined by non-Fortran means. This feature gives dummy-argument information to calling programs (and for called functions, function type information), which is needed for using keyword and optional arguments. It also makes possible validation of actual argument types and number. Procedure interfaces, via the INTERFACE block, may appear directly in the specifications of calling programs. However, more often interface information will be packaged in modules, which are then USED by the calling program.

In Fortran 8x, it is proposed that procedure definitions may be made within any program unit. Such a procedure is called an "internal procedure" and is known only locally within the program unit in which it is defined. The form of internal procedures is identical to that of external procedures except for the INTERNAL on the heading and END statements. Also, internal procedures automatically inherit all host data and procedure definitions, unless such inheritance is explicitly suppressed with the use of the USE statement (e.g., USE ONLY () suppresses all inheritance from the host). Internal functions provide the same functionality as statement functions, but are more flexible and not so restricted. A common use of internal subroutines will be as "remote code blocks". Internal procedures are called in exactly the same way as external procedures.

An important use of internal functions is to provide operations on derived data types. Since infix operators are so common in scientific notation, an option is provided with internal functions to allow them to be used as infix operators. Similarly an option with internal subroutines allows the assignment operator (=) to be used with programmer-defined data conversions. Thus, the various features of internal procedures, together with derived data type definition and module encapsulation facilities, gives Fortran 8x a powerful and flexible data abstraction mechanism.

```

function-heading is [RECURSIVE] [data-type] FUNCTION function-name
                  [(dummy-argument-list)] [RESULT(identifier)]

subroutine-heading is [RECURSIVE] SUBROUTINE subroutine-name
                    [(dummy-argument-list)]

procedure-reference is function-name [(actual-argument-list)]
                    or CALL subroutine-name [(actual-argument-list)]

```

actual-argument-list *is* positional-argument-list [, keyword-argument list]
or keyword-argument-list
positional-argument-list *is* expr [, expr]...
keyword-argument-list *is* dummy-argument-name= expr [, dummy-argument-name= expr]...
optional-attribute *is* OPTIONAL
intent-attribute *is* IN
or OUT
or IN OUT
present-intrinsic-function *is* PRESENT (dummy-argument-name)
procedure-interface *is* INTERFACE
 [interface-description]...
 END INTERFACE
interface-description *is* function-heading
 [local-specification]...
or subroutine-heading
 [local-specification]...
internal-procedure *is* internal-procedure-heading
 [use-statement]...
 [local-specification]...
 [executable-construct]...
 [internal-procedure]...
 END INTERNAL [unit-type[unit-name]]
internal-procedure-heading *is* INTERNAL function-heading [OPERATOR (operator-symbol)]
or INTERNAL subroutine-heading [ASSIGNMENT]
operator-symbol *is* . identifier .
or intrinsic-operator-symbol

Notes and Examples -

- Arguments are optional, as specified by the OPTIONAL attribute; the PRESENT function may be used to determine if a given optional argument was actually supplied in the call.

- No positional arguments may appear after the first keyword argument.
- OPERATOR functions have one or two arguments (the operands of the operation); such functions may be called using either the functional form or the infix operation form. The intrinsic operator symbols (+, -, *, /, **, //, .AND., .OR., etc.) may be overloaded. Alternatively any .name. form may be defined as an operator symbol.
- ASSIGNMENT subroutines have two arguments, the first one being the left-hand-side of the assignment (the entity being defined), and the second being the value to be assigned. The usual assignment syntax (with the = sign) may be used in assignment procedure calls.
- In the case of both operator and assignment definitions, the procedure body defines in detail the intended operations.
- Note that internal procedures may be nested.
- The RESULT identifier may be used with recursive functions to remove ambiguity between recursive calls and function value assignment (RESULT may also be used with non-recursive functions, including OPERATOR definitions).
- The attributes of procedure headings (INTERNAL, RECURSIVE, etc.) may appear in any order.
- See the preceding section for examples of internal procedures.
- The following two examples illustrate recursion and the use of the PRESENT function:

```

recursive integer function ACKERMANN(M,N) result (ACK)
                                  M,N: integer
          if (M.eq.0) then; ACK = N+1
          elseif (N.eq.0) then; ACK = ACKERMANN(M-1,1)
          else                  ; ACK = ACKERMANN(M-1,ACKERMANN(M,N-1))
          end if
end function ACKERMANN

real function READNUM (UNIT,FMT)
                                  UNIT: integer, in, optional
                                  FMT: character(*), in, optional
LUN,IOS: integer
LUN = 5; if (present(UNIT)) LUN = UNIT ! establish unit no.

```

```

if (present(FMT)) then
  read (LUN,FMT,iostat=IOS) READNUM
  if (IOS.ne.0) READNUM = 0
else
  read (LUN,*) READNUM
end if
end function READNUM
! X = READNUM ( )      reads from unit 5, list-directed format
! X = READNUM (10)    reads from unit 10, list-directed format
! X = READNUM (FMT= '(F10.2)') reads unit 5, format F10.2

```

- The above two examples are external procedures and, as such, the routines calling them do not normally have any interface information about them. In Fortran 8x, the use of an interface block, which provides calling routines with interface information, allows the compiler to check for correctly formed procedure calls. The following example is an interface block that contains interface information pertaining to the above procedures READNUM and ACKERMANN. This interface block may be placed in either the calling routine itself, or a module being used by the calling routine.

```

interface
  recursive integer function ACKERMANN(M,N)
                                M,N: integer
  real function READNUM(UNIT,FMT)
                                UNIT: integer, in, optional
                                FMT: character(*), in, optional
end interface

```

2.7 Program Source Form

The Fortran 8x source form is completely column-independent, and source lines (records) may be any length (a limit of 1320 characters per source statement is, however, still the rule). Statements may be separated on a line with the ";", "'" (not in a character context) initiates a comment (either on a line by itself or following a statement), and "&" at the end of a line signifies continuation. Blanks are significant characters, and identifiers may not contain embedded blanks; conversely, blanks must separate adjacent identifiers and keywords. Identifier names may be up to 31 characters long and may contain the "_" (underscore) character. Upper and lower case may be used interchangeably (except in CHARACTER constants). Either apostrophes or quotation marks may be used as character constant delimiters.

The following is a more formal description of much of the new source form.

```

source      is [statement ;]... [statement-start] [comment] eol
            or [statement-continuation] [comment] eol

statement-start is statement [;]
               or statement-fragment &

statement-continuation is [&] statement-fragment [; [source]]
                       or [&] statement-fragment &

comment      is ! [character]...

character-constant is ' [character]... '
                   or " [character]... "

```

Notes -

- A statement-fragment is a contiguous portion of a statement (and may be null). A statement-fragment may end in the midst of a (continued) character constant or H, apostrophe or quotation mark edit descriptor only if there is no trailing comment.
- If a statement-fragment (in a statement-continuation) is preceded by a "&", any blanks preceding that "&" are insignificant.
- The eol stands for the end of a source line of text.
- If the underscore is used in an identifier it must not be the first character of the identifier; the underscore is a significant character.
- If the character used to delimit a character-constant (' or ") is to appear within the constant itself, it must appear exactly twice in succession (i.e., "" "" is the same as ' ' ' and ' ' ' is the same as " " ").
- Many of the examples in this document illustrate aspects of the Fortran 8x program source form.

2.8 Control Constructs

Two control constructs are added, one for loop control and one for case selection:

block-do *is* DO [(loop-control)]
 [executable-construct]...
 REPEAT

loop-control *is* integer-variable = integer-expr, integer-expr [,integer-expr]
 or integer-expr TIMES

block-case *is* SELECT CASE (enumerable-expr)
 [CASE case-selector
 [executable-construct]...]...
 END SELECT

case-selector *is* (value-range [,value-range]...)
 or DEFAULT

value-range *is* constant-expr
 or [constant-expr]:[constant-expr]

Notes -

- Branches into block constructs are not allowed.
- The block-do specifies repetitive execution of its block of loop-statements until the loop-control (if any) is satisfied, or until an EXIT statement is executed.
- An EXIT statement is allowed only within a block-do (it may be in a block-if or block-case that is one of the loop-statements); its execution terminates execution of the innermost block-do containing the EXIT statement.
- Execution of the CYCLE statement causes the next iteration of the loop to commence.
- The indexed form of loop-control is semantically identical to the Fortran 77 DO statement with an index of type integer.
- Enumerable-expr is an expression of type integer, character, or logical; constant-expr is a constant expression of the same type as enumerable-expr.
- The case-selectors must be disjoint in any given block-case.

- CASE DEFAULT is optional and may not appear more than once in any given block-case; if present, it may appear in any position among the sequence of CASE clauses.
- The block-case causes execution of the block corresponding to the case-selector that contains the value of the enumerable-expr; if there is no such case-selector then the CASE DEFAULT block is executed; if there is no matching case-selector and there is no DEFAULT block, an error exists.
- Block constructs (if, do, case) may be named by placing an alphanumeric name to the right of each of the control statements (e.g., do, repeat, else if, etc.). EXIT and CYCLE statements may contain the name of the block-do construct to be exited or cycled.

2.9 Input/Output

Input/Output additions include several new OPEN statement specifiers, name-directed I/O, and new edit descriptors.

position-specifier *is* POSITION= position-expr

action-specifier *is* ACTION= action-expr

delimiter-specifier *is* DELIM= delim-expr

name-directed-format-specifier *is* [FMT=] **

name-directed-input *is* variable-name = data-value [, data-value]...

slash-edit *is* [repeat-count] /

engineering-edit *is* [repeat-count] EN width . digits [E exponent]

Notes -

- The position-specifier specifies the position of the file upon open; position-expr must evaluate to one of the character values 'REWIND', 'APPEND', or 'ASIS'.
- The action-specifier specifies read-only, write-only, etc; action-expr must evaluate to one of the character values 'READ', 'WRITE', or 'BOTH'.

- The delimiter-specifier is used in conjunction with name-directed I/O; delim-expr must evaluate to one of the character values 'NONE', 'QUOTE', or 'APOSTROPHE'.
- The engineering-edit descriptor causes 1-3 digits to be displayed to the left of the decimal point such that the exponent is a multiple of three.

2.10 Event Handling

The event handling mechanism provides a means to transfer control on the occurrence of an event to a specified program unit called an event handler. Events must be declared, and their monitoring can be selectively switched on and off to accommodate the desired level of optimization.

An eventmark is a data object that registers the occurrence of an "event". The declaration of an eventmark contains the condition for which the event will occur. The value of the eventmark is either .ON. or .OFF. (the constants of the data-type EVENTMARK). The scoping and definition rules for eventmarks are the same as for any other data type.

Event handlers are similar to subroutines, except that they do not have arguments and cannot be called. When an event occurs, the connected handler is automatically invoked. Upon completion of a handler, execution may either RESUME with the statement following the one in which the event occurred, or RETURNUP (return from the procedure in which the handler was invoked).

```

event-handling-  is ACTIVATE (event-name-list)
statement        or DEACTIVATE (event-name-list)
                 or CONNECT (event-name, handler-name)
                 or DISCONNECT (event-name)
                 or DISCONNECT (handler-name)

```

Notes -

- The intrinsic operations on eventmarks are assignment and test for equality.
- An optional part of the eventmark declaration (not shown) allows the specification of the condition(s) under which the eventmark is automatically set on.

2.11 Other Features

Other added features include IMPLICIT NONE for turning off implicit typing and several new CHARACTER intrinsic functions. In addition, restrictions have been removed on: assignment from overlapping character positions, zero-length character strings, concatenation of CHARACTER dummy arguments, and specifying character constants.

```

character-      is IACHAR  same as ICHAR, but based on ASCII
intrinsic-function or ACHAR  same as CHAR, but based on ASCII
                 or TRIM   remove trailing blanks from string
                 or VERIFY check for unwanted characters
                 or ADJUSTL circular shift left thru leading blanks
                 or ADJUSTR circular shift right thru trailing blanks
                 or REPEAT  replicate a string of characters
                 or ISCAN  first position of any character from set

implicit-list   is implicit-type [, implicit-type]...
                 or NONE

implicit-type   is type ( letter-range [, letter-range]... )

```

3. LANGUAGE ARCHITECTURE

The proposed Fortran 8x language contains all of the features of Fortran 77, some of which are identified as "deprecated," and additional new features as described in Section 2 above. The "core" is that set of language features that are not identified as deprecated. "Modules" are nonexecutable program units which contain prepackaged definitions to be used by executable program units.

3.1 The Core

The core is a complete and consistent language comprising a set of language features sufficiently rich for the implementation of most applications. The core will have at least the same functional capabilities as Fortran 77. In addition, the core should

- (a) be especially suitable for scientific applications,
- (b) be portable,
- (c) be safe to use, and effective for the development of reliable software,
- (d) be widely efficiently implementable,
- (e) be concise,
- (f) comprise generally accepted contemporary language technology,
- (g) minimize non-automatable conversion from Fortran 77.

A core-conforming program contains only core features (i.e., does not contain any deprecated features).

3.2 Modules

Modules provide a mechanism for defining program facilities for subsequent use in application programs. Such facilities include global data pools, procedure libraries, procedure interface definitions, data abstractions, and language extensions in the form of new operators. Modules take the form of MODULE program units, and contain data type definitions, data object declarations, procedure definitions, and procedure interface definitions. The USE statement provides the mechanism for using the public contents of a module in an application program.

Modules are useful, for example, where the same definitions are needed in a number of different program units, because they allow such definitions to be made only once in a "central" place. This should contribute substantially to software reliability and reusability. The use of modules can also contribute to execution efficiency through in-line expansion of procedures. Procedure libraries may be configured as internal procedures in a module and thereby made available for in-line expansion in the using program units.

Modules may be individually standardized, so that standard facility definitions are available to implementors for efficient implementation (e.g., to take advantage of specialized hardware). An example might be a module for bit data type, which defines the structure of and operations on bit data; standardizing such definitions allows implementors to efficiently implement a standard bit data facility if they so choose.

Since a module contains "ready to use" definitions, a standard module provides standardized facilities without an added cost burden to a standard-conforming Fortran implementation. Therefore, such facilities are part of any Fortran implementation, regardless of whether or not the implementor chooses to optimize implementation of them. For this reason, module standardization is an extremely powerful and cost-effective mechanism for extending the functionality of standard Fortran.

A standard module must be core-conforming. Operators defined in the module must not have the potential to alter the meaning of any core-intrinsic operation. The module must contain, in the form of appropriate commentary, functional descriptions of all module operations; such functional descriptions take precedence over any accompanying procedural implementations of functionality. A functional description may be a (set of) mathematical statement(s), or take any other form appropriate for comprehensive specification of the functionality. In addition, the module must contain a succinct and lucid description of the use of the module facilities.

Initial possibilities for standard modules include

- bit data type
- maximum accuracy arithmetic
- varying-length character (string data type)

Standard procedure libraries, such as the ISA process control library, the Industrial Real Time Fortran (IRTF) library, and the Graphical Kernel System (GKS) library are candidates for standard modules.

Standard modules are separate standards, and are not part of the Fortran standard itself. There is a close relationship between standard Fortran and standard modules, of course, which gives rise to the concept of a "Fortran

family of standards". Any group, including X3J3, may propose standard modules. In addition X3J3 may choose to include certain standard modules in a part of the standard Fortran document.

3.3 Deprecated Features

As described in Section 1.5, certain features of the proposed Fortran 8x language are identified as deprecated and as such are identified as candidates for removal from subsequent versions of the Fortran standard. In this section, the deprecated features are summarized, together with a discussion of possible replacements for needed functionality.

3.3.1 Summary of Deprecated Features

1. Fortran 77 source form (e.g., restricted use of columns 1-6)
2. alternate RETURN
3. assumed size dummy arrays
4. passing a scalar (e.g., array element or substring) to a dummy array
5. specific names for intrinsic functions
6. statement functions
7. BLOCK DATA program unit
8. arithmetic-if statement
9. ASSIGN statement
10. assigned-goto statement
11. COMMON statement
12. computed-goto statement
13. DATA statement
14. DIMENSION statement
15. DOUBLE PRECISION data type
16. ENTRY statement
17. EQUIVALENCE statement
18. Fortran 77 DO statement
19. PAUSE statement

3.3.2 Storage Association

Storage association is the association of data objects through physical storage sequences, rather than by object identification. Storage association allows the user to configure regions of physical storage, and to conserve the use of storage by dynamically redefining the nature of these storage regions. Though the considerable dangers of programmer access to physical storage sequences have been well known for a long time, not until Fortran 8x has Fortran had adequate replacement facilities for certain important functionality provided by storage association. Six items in the above list (items 3, 4, 7, 11, 16, and 17) are due to the identification of storage association as deprecated.

3.3.2.1 Assumed Size Dummy Arrays

These are dummy arrays with asterisk as the size of the last dimension. In Fortran 8x dummy arrays may also be assumed shape, in which case the extent in each dimension is that of the actual argument. Assumed-shape arrays provide all of the functionality of assumed-size ones, and more. Assumed-size arrays assume that a contiguous block of storage is being passed, whereas with assumed-shape arrays an array section not having contiguous physical storage (such as a row of a matrix) may be passed also.

3.3.2.2 Passing a scalar (e.g., Array Element or Substring) to a Dummy Array

This functionality is now achieved, and much more safely, by passing the desired array section. For example if a one-dimensional array XX is to be passed starting with the sixth element, then instead of passing XX(6) to the dummy array one would pass the array section XX(6:). If the eleventh through forty-fifth elements are to be passed, the actual argument should be the array section XX(11:45).

3.3.2.3 BLOCK DATA Program Unit

The principal use of BLOCK DATA subprograms is to initialize COMMON blocks. The global data functionality of COMMON is alternatively provided by MODULE program units; global data in modules may be initialized at the point of declaration, and thus modules provide a complete replacement option for BLOCK DATA subprograms.

3.3.2.4 COMMON Statement

The important functionality of the COMMON statement has been its use in providing global data pools. In Fortran 8x, global data pools may also be provided, and more safely and conveniently, with MODULE program units and USE statements. Suppose that it is desired to have a global data pool consisting of the following:

```
INTEGER X(1000)
REAL Y(100,100)
COMMON /POOL1/ X,Y
```

Each program unit using this global data would need to contain these specifications. Alternatively, one can define the global data pool in a MODULE program unit:

```
MODULE POOL1
  INTEGER X(1000)
  REAL Y(100,100)
END MODULE
```

Then each program unit using this global data would contain the statement

```
USE /POOL1/
```

This is safer than using COMMON, because the structure of the global data pool appears in only one place. In addition the USE statement is very short and easy to use. Facilities are provided in the USE statement (but not shown here) to rename module objects if different names are desired in the program unit using the module objects.

Modules have another advantage: they do not involve storage association, and therefore a module can contain any desired mix of character, non-character, and structured objects. Since COMMON involves storage association, a given COMMON block cannot contain both character and non-character data objects.

3.3.2.5 ENTRY Statement

The ENTRY statement is typically used in situations where there are several operations involving the same set of data objects:

```
procedure-heading
  local-specifications
entry1
  ...
  RETURN
entry2
  ...
  RETURN
...
entryn
  ...
  RETURN
END
```

The MODULE program unit provides the same functionality:

```
MODULE module-name
  local-specifications
INTERNAL procedural
  ...
  END INTERNAL
INTERNAL procedure2
  ...
  END INTERNAL
...
INTERNAL proceduren
  ...
```

L&T-60

```
END INTERNAL
END MODULE
```

A program unit using this module may call each internal procedure in it, exactly as if they were entry points. An advantage is that some of the internal procedures in a module may be functions and some may be subroutines, whereas all of the entry points in a function procedure must be functions and in a subroutine must be subroutines.

3.3.2.6 EQUIVALENCE Statement

The primary purpose of the EQUIVALENCE statement is to allow the programmer to associate, within a program unit, two or more data types with the same storage region. There is currently no direct alternative for this functionality in the Fortran 8x proposals.

There were two principal reasons for the EQUIVALENCE statement in early versions of Fortran. The first is that memory address spaces were typically quite small, and equivalencing was needed to reuse the available space for different purposes. The second reason is that early versions of Fortran did not have a rich set of data types and structures and transfer functions between types, so that equivalencing was useful in simulating certain data types and structures. Both of these reasons have now largely disappeared.

Much of the same practical effect of reusing physical storage can, in fact, be achieved with the proposed facilities without using the EQUIVALENCE statement. Dynamic local arrays may be of any size, controlled by procedure arguments. Since dynamic arrays "disappear" upon return from the procedure, other procedures may use this space for other dynamic arrays of different shapes, sizes, and types. Similarly, allocatable arrays disappear when they are freed explicitly and on return from the procedures in which they were allocated. The IDENTIFY statement allows part of an array to be referred to as a completely different object (but of the same type).

3.3.3 Redundant Functionality

A number of features are identified as deprecated simply because they are now completely redundant, having been superseded by superior language features. These redundant features are items 1, 5, 6, 8, 12, 13, 14, 15, and 18 in the list above.

Fortran 77 source form -- replaced by the new source form
(Section 2.7 above)

specific names for intrinsic functions -- use generic names

statement functions -- replaced by internal functions

L&T-61

(see 3.3.3.1. below)

arithmetic-if statement -- replaced by logical-if and block-if

computed-goto statement -- replaced by block-case construct
(see 3.3.3.2 below)

DATA statement -- replaced by INITIAL attribute
(Section 2.1), and by assignment
of array constants (Section 2.2)

DIMENSION statement -- use type declarations instead

DOUBLE PRECISION statement -- use precision control attributes
(Section 2.3)

Fortran 77 DO statement -- replaced by block-do construct
(Section 2.8)

3.3.3.1 Use of Internal Functions for Statement Functions

The statement function definition

```
function-name(dummy-argument-list) = expr
```

may be completely replaced by the internal function definition:

```
INTERNAL FUNCTION function-name(dummy-argument-list)
  type-specification-of-dummy-arguments
  function-name = expr
END INTERNAL
```

The use of the internal function in the executable code is the same as the use of the statement function.

3.3.3.2 Example Replacement of the computed-goto Statement

The code sequence controlled by the computed-goto:

```
GOTO (label1, label2, ..., labeln), integer-variable
...
GOTO labelz
label1 CONTINUE
...
GOTO labelz
label2 CONTINUE
...
GOTO labelz
...
labeln CONTINUE
...
GOTO labelz
...
labelz CONTINUE
```

may be replaced by the block-case construct:

```
SELECT CASE (integer-variable)
  CASE DEFAULT
  ...
  CASE (1)
  ...
  CASE (2)
  ...
  ...
  CASE (n)
  ...
END SELECT
```

3.3.4 Other Deprecated Features

The remaining obsolete features (items 2, 9, 10, and 19) in the above list are neither related to storage association nor directly replaced by superior features. They are all in some sense "bad practice" in terms of generally accepted software principles, however, and their effects may be achieved in ways that are generally now considered better software practice.

3.3.4.1 Alternate RETURN

Alternate returns introduce control (branch point) information into argument lists, and then one (the called) program unit can directly control execution sequence in another (the calling) program unit. This tends to complicate the readability and maintainability of software using this feature. Better practice, in terms of readability and maintainability, is to use a data element in the argument list to contain a "return code", which can then be used in the

calling program to explicitly control the execution sequence in the calling program. For example, consider the following use of alternate returns:

```
CALL subr-name(X,Y,Z,*100,*200,*300,...)
GO TO 999
100 CONTINUE
....
GOTO 999
200 CONTINUE
....
GOTO 999
...
999 CONTINUE
```

where labels 100, 200, 300, etc., are the beginnings of the code blocks to be selected from upon return. In many cases, the effect can be more safely achieved with a return code and a block-case structure:

```
CALL subr-name(X,Y,Z,RETURN_CODE)

SELECT CASE (RETURN_CODE)
CASE ('condition1')
....
CASE ('condition2')
....
END SELECT
```

Alternatively, RETURN_CODE could be used to explicitly control direct branches (GOTOs) to the desired branch points. Maintainability is enhanced with the use of RETURN_CODE because a new selection case can be added in a convenient, straightforward manner, without modifying the actual and dummy argument lists.

3.3.4.2 ASSIGN and assigned-goto

The ASSIGN statement allows a label to be dynamically assigned to an integer variable, and the assigned-goto statement allows "indirect branching" through this variable. The dangers of such dynamic indirection, especially when mixed with integer arithmetic operations, have long been recognized, and a comprehensive alternative to this functionality is not provided. An important practical use of such indirect branching, however, is to return from simulated internal subroutines. Consider the following example code, which "calls" one of several "internal subroutines" in the program unit:

```
ASSIGN 120 TO RETURN      ! set up return point
GOTO 740                  ! branch to "subroutine"
120 CONTINUE
```

```
...
740 CONTINUE
...      ! "subroutine" body
GOTO RETURN
...
```

This functionality is provided in a much better way, through the use of internal subroutines:

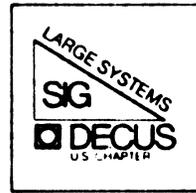
```
CALL subroutine-name
...
INTERNAL SUBROUTINE subroutine-name
...      ! subroutine body
END INTERNAL
...
```

This illustrates the use of internal subroutines to conveniently provide "remote code block" functionality.

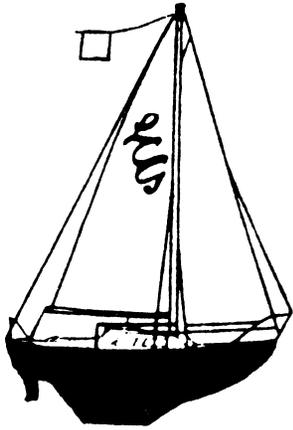
3.3.4.3 PAUSE Statement

Execution of a PAUSE statement requires operator/system-specific intervention to resume execution. In most (if not all) cases, the same functionality can be achieved as effectively and in a more portable way with the use of appropriate READ statements.





DIGITAL EQUIPMENT COMPUTER USERS SOCIETY



AT LARGE



The Newsletter of the Large Systems SIG

LARGE SYSTEMS

SIG Chairperson

Leslie Maltz
Stevens Institute of Tech.
Hoboken, NJ

Networks Coordinator

Richard Janick
Abbott Laboratories
Abbott Park, IL

Symposium Coordinator

Robert C. McQueen
Stevens Institute of Technology
Hoboken, NJ

Systems Software Coordinator

Betsy Ramsey
American Mathematical Society
Providence, RI

Newsletter Editor

Michael Joy
First Church of Christ, Scientist
Boston, MA

Special Projects Coordinator

Osman Ahmad
Assoc., of American Railroads
Chicago, IL

Menu Coordinator

Charles R.T. Bacon
National Institutes of Health
Bethesda, MD

Special Projects Coordinator

F. Berkeley Shands
Emerson Electric Co.
St. Louis, MO

Hardware Coordinator

Clive Dawson
MCC
Austin, TX

DEC Counterparts

Dave Braithwaite
Digital Equipment Corporation
Marlboro, MA

Languages Coordinator

David Edwards
DPEX, Inc.
Fremont, CA

Jack Buckley
Digital Equipment Corporation
Marlboro, MA

Reed Powell
Digital Equipment Corporation
Marlboro, MA

TOPS-20 Coordinator

Pete Gavin
University of Texas at Austin
Austin, TX

IN THIS ISSUE

	<u>Page</u>
SIG Officers	LS-i
Chairperson's Message	LS-2
From the Editor	LS-3
Spring 1985 SIG Menu	LS-4

CHAIRPERSON'S MESSAGE

Just as things are constantly changing in the area of computer technology, so are DECUS products and services. The newsletter which you are now reading is something of a new concept for DECUS. For many years, DECUS, and individual SIGs in particular, have produced publications and newsletters to keep our membership informed about the most recent happenings. We are at the beginning of a new phase in the production of such a newsletter. This is the first edition of what is sometimes called "the Big One", or a single publication including information submitted by all SIGs. There have been many long, emotional debates as to whether to combine all newsletters into a single publication such as this one, or to continue with individual (or groups of) SIG newsletters. Hopefully, those in the decision making positions have made a wise choice. Your comments on this new format are both solicited and encouraged. How do you like the concept? Will you now be able to read about things that might be beneficial to you or your installation that you might not have seen? Perhaps, on the other side, you really have no interest in all this other stuff. Either way, please write and let us know your thoughts.

On Large Systems related business, some interesting things have been happening. We have been asked for our input for suggestions for features for the next major release of VMS. For those of us who will be integrating with VAXen, this input is both timely and important. On another note, I recently participated in meetings at DEC as a followup to sessions in New Orleans. I met with Dick Poulsen (Field Service) and Rich Whitman (LSM) to discuss many of the topics that were identified in New Orleans as areas of major concern to the Large System installed base. The meetings were friendly and productive. I hope that we will see some of the results in the form of communications from each of them in the near future.

Since New Orleans much effort has gone in to the compilation of the Large Systems SIG MENU. Chuck Bacon, our Menu Coordinator, has been busy putting the menu into final form. If you are listed in the SIG membership database, you will (or perhaps by the time you read this, already have) receive(d) a copy. Please complete it and send it to us as soon as possible. The results will be compiled and published shortly after the returns are in.

Based on many comments heard in New Orleans as well as those expressed in phone conversations, I'd like to reassure our membership that we will continue to be supporting several types of activities and services. We will continue to provide information and symposium sessions for those installations who will be keeping and running their 36 bit systems until the end of time. We are not abandoning either our systems or original purpose. In addition, we will be supporting activities that relate to the integration of VAX systems with our 10/20 systems. In fact, we are taking a strong position that these VAX systems are part of an integrated environment that should be considered as part of a "data center" concept. This idea seems to describe a certain overall way of doing business, and sets an expectation for the way we expect DEC to maintain and support our systems. In many ways this is similar to what used to be called a "mainframe mentality". More will probably be said about this concept in future editions and at symposia.

We hope you find this new format informative, and will write to us to share your views and concerns. We will attempt to keep you informed of the latest as quickly as we hear about it. Please copy the SIG on correspondence to DEC that you think would be of interest to others in similar situations. We all can benefit from the efforts, triumphs, and/or problems of others. Let us know what seems to be working and where we can help make it right for all of us.

See you in December!

LS-2

---Leslie Maltz

FROM THE EDITOR

Well here it is! The first edition of the now-famous "Big One". As Leslie stated in her Chairperson's Message there was a lot of thought and discussion (albeit sometimes heated) that went into the decision to go with one large issue covering all the SIG's. We would like to get your initial feelings on the newsletter - both the whole concept of the "Big One" and your thoughts on the Large Systems SIG portion. Send your thoughts and suggestions to the address shown below. We will print as many comments as possible in a subsequent issue of the newsletter.

Because this will be a monthly publication the SIG will attempt to give you even more timely information than we've been able to with the previous newsletter which averaged about once a quarter. We expect we will have perhaps fewer pages per issue but overall the data can be more current.

The officers will frequently contribute articles of interest for all but we need your input to the newsletter as well. This is your opportunity to voice your feelings, contribute your ideas, pass along your systems tips, etc. It's also your opportunity to get your name in print. We hope each of you will help make the newsletter a success by your contributions.

In this edition you will find several letters regarding AMAR. We expect this particular problem will soon have a satisfactory answer, but as of this date it is still unresolved. Informative letters of this nature addressed to DEC or from DEC in response to your problems can be helpful to others by alerting them to potential pitfalls they should avoid or solutions to problems they too may have encountered. In some instances it may also point out problems that on the surface appear to be unique but with exposure may prove to be more universal. Bringing out problems of this type can be beneficial to DEC when they can see that a given issue is of concern to many.

The Spring 1985 SIG menu is just about ready for mailing. We thought you might like to see a preview of the menu items so that you can research any that you are unfamiliar with. This is also an opportunity for you VAX SIG members to review integration issues as well as become familiar with advanced TOPS features you might want to consider requesting for VMS.

Well that's it for the September issue. Again, we welcome your input to your newsletter. Submit articles, comments and letters to:

Michael D. Joy
The First Church of Christ, Scientist
Christian Science Center, A-41
Boston, MA 02115

--Michael Joy

LS-3

SPRING 1985 SIG MENU

CONTENTS

Introduction	1
1. Marketing	3
1.1 Local sales staff need AI-based configuration tools	3
1.2 Strong national or regional LCG sales support in lieu of local	3
1.3 Continue providing TOPS training courses	3
1.4 Bring back the RP07. Reason: performance	4
1.5 When TOPS-20 development ends, provide sources to all customers	4
1.6 Reduce HSC-50 price to TOPS customers	4
1.7 TOPS third party license fee of \$80,000 is too much	4
1.8 Extend KL trade-in offer to other parts of the system	5
1.9 Extend KL trade-in offer to other upgrades, not just 8600	5
1.10 Extend KL trade-in offer annually	5
1.11 Separate maintenance charges for MF20 into its component parts	5
1.12 Improve PULSAR reliability	6
2. Monitor/Network	7
2.1 Implement global ENQ/DEQ on clustered TOPS-20 systems	7
2.2 Add /COLUMN, /LINE, etc. to TOPS-20 TYPE	7
2.3 UDA-50-type interface for KL's; Cheaper than HSC	7
2.4 Finish DECNET-10 DDP device implementation, to save DN20 cost	8
2.5 TOPS-20 DELETE should allow DIRECTORY-like selection	8
2.6 Accounting functions on -10 should be kept in a single program	8
2.7 Provide for EXEC based on username when directory is created	8
2.8 Prevent ctrl-C or modification of LOGIN.CMD on -20	8
2.9 TA78 support from TOPS-20 clusters	9
2.10 Allow TOPS systems to boot from and swap to RAXX disks	9
2.11 Supply MIC and PCL sources to source sites	9
2.12 Network support from COPY command on -20	9
2.13 Partial recognition of leading parts of filenames	9
2.14 User groups on 20 as SIXBIT words instead of integers	10
2.15 Enhance date-time parsing in TOPS-20 to equal of TOPS-10	10
2.16 Don't drop FACT files in TOPS-10	10
2.17 Relative directory addressing in TOPS-20	10

2.18 Customer-defined terminal support in EDT-20	10
2.19 Implement PURGE command on TOPS-20	10
2.20 Allow GALAXY queues on non-PS: structures	11
2.21 Add AUTHOR specifier to DIRECTORY command	11
2.22 Enhance Datatrieve-20 to support DBMS-20	11

3. Layered Products 12

3.1 COBOL-8x on TOPS-10	12
3.2 Provide structured programming constructs in COBOL-10	12
3.3 Language Sensitive Editors for TOPS similar to VMS LSE	12
3.4 Extended addressing in TOPS-10 COBOL, FORTRAN and LINK	12
3.5 Extended addressing for TOPS-10 utilities like SORT, LINK, DDT	13
3.6 Tape and disk catalog for TOPS-20	13
3.7 Multiforking DDT for TOPS	13
3.8 Add network file access support to RMS-10	13
3.9 DBMS-20 should support remote access from DATATRIEVE-32	13
3.10 Provide support for long symbols in TOPS compilers, DDT	13
3.11 Autopatch should have REL files for vanilla sites	14
3.12 LINK-20 should be made native	14

4. Integration 15

4.1 Support a foreign tape utility for VMS	15
4.2 Create a HELP system on VMS based on TOPS keywords	15
4.3 Create a COMND JSYS equivalent as part of VMS	15
4.4 Implement a TOPS-20-like CLI for VMS	15
4.5 Develop and support a TOPS-to-ULTRIX integration effort	16
4.6 Produce an Integration Planning Guidebook	16
4.7 Create a GALAXY-like environment for VMS	16
4.8 Create Micro-VAX clusters on Ethernet or other medium	17
4.9 Implement COMPIL-class commands in VMS	17
4.10 DEC-supported TCP/IP on VMS	17
4.11 Add workload capability to SPM	17
4.12 Give VMS MAIL the power to send to tree-structured usernames	18
4.13 Project accounting on VMS with independent billing identifiers	18
4.14 Extend VMS usernames to 39 characters	18
4.15 Improve tape/disk management in VMS	18
4.16 Enhance VAX TYPE command to include line and column ranges	18
4.17 Provide timestamps in VMS batch log files	19
4.18 VMS memory management in real time	19
4.19 DEVELOP a TOPS-20 shell for ULTRIX	19
4.20 Support TCP/IP over TTY lines	19
4.21 Low-cost VAX attachment for TU72 tape drives	19
4.22 Support IBM standard label tapes (EBCDIC) in RMS	19

Introduction

SPRING 1985 MENU

This is the official Spring 1985 Menu of the DECUS Large Systems SIG. It is based on the issues identified at the Spring Symposium in New Orleans. The specific items were suggested during the scheduled sessions, and most were discussed at the Town Meeting session. A straw vote on the items was taken at that time, but now all installations are requested to participate in this mailed ballot. Your participation in the voting process is essential if the results are to be considered statistically valid and representative of the installed base of sites with DECsystem-10 and DECSYSTEM-20 computers. We hope that you will participate by indicating your priorities for issues identified.

The results of the Menu will be presented to Digital as a timely indication of the needs of our installations and the direction we wish Digital to pursue in future development to meet our needs. The results of the Menu will be available to all members of the Large Systems SIG and will be published in our newsletter. The results will also be presented at the Fall Symposium in Anaheim.

The Menu consists of four parts. The first part contains questions that pertain to marketing issues of interest to current DECsystem-10/20 installations. The second part consists of issues related to the operating systems and networking. The third section relates to questions and requests for layered products. The final section deals with issues related to the integration of 36 and 32 bit systems. Many of these items will be necessities for the next and future releases of VMS based on the needs of current 36 bit installations. Some issues could really be identified with other or multiple sections of the menu, but have been listed in only one place for the sake of time and space.

Each item in this menu has a number, a title, and some text. The text is meant to provide a description of the item, and does not represent a position taken by the Large Systems SIG. It is included for explanatory purposes only. The ballot lists the items in the same order with the same descriptive titles for ease of voting. Please take this menu seriously, as it will provide a basis for actions by Digital which will affect every DECsystem-10 and DECSYSTEM-20 site.

One menu item, number 1.7, has an additional "Suggested amount" to be filled out in the ballot, if the voter feels that this item is important enough to deserve votes.

We ask that every ballot be filled out with a maximum of 25 votes. A

maximum number of votes of five (5) may be cast for any individual menu item. There is no such thing as a fractional vote.

The ballot should be completed and mailed as soon as possible. All ballots must be returned by September 28th. They should be sent to the Large Systems SIG MENU, DECUS, 219 Boston Post Rd. (BP02), Marlboro, MA 01752. Remember, only the ballot should be returned. We look forward to receiving your ballot and thank you for participating in the Menu. Further discussion of these items and other issues of interest is encouraged via letters and articles in our newsletter. Thank you for your participation.

Leslie Maltz, Large Systems SIG Chairperson

Charles Bacon, Large Systems Menu Coordinator

Category 1

Marketing

Marketing and Field Service share the characteristic of being the customer's liaison with Digital. The sections labeled Networks/Monitor, Layered Products, and Integration have to do with what Digital sells, and not so much with what Digital does. Items in this section request actions by Digital which may fall more in the category of business decisions, and are of a less technical nature. As noted in the Introduction, many Menu Items may be considered to belong to a different category than the one in which they are placed.

1.1 Local sales staff need AI-based configuration tools

Local Digital sales people and customers alike would benefit from the availability and use of DEC's expert system configuration tools XCEL and XCON.

1.2 Strong national or regional LCG sales support in lieu of local

This item attacks a problem of LCG sales, noted above. Sales personnel are not sufficiently knowledgeable about the LCG/LSM product lines to perform their tasks for customers properly. The suggestion is therefore made that LSM create a national, or possibly a very few regional, sales support groups, to replace local LCG sales support. The purpose would be to provide a better level of expertise in all LCG matters.

1.3 Continue providing TOPS training courses

DEC Educational Services has cancelled TOPS-10 and TOPS-20 courses. CSC in Colorado says "...don't know when they will schedule a course..." Please regularly schedule these courses, or possibly provide videotaped or comparable training materials for TOPS-10 and TOPS-20.

1.4 Bring back the RP07. Reason: performance

The RP07 has been dropped from Digital's product line, while the older, slower RP06 churns on. Business reasons for dropping the RP07 may yield to customer preference for this higher performance, higher capacity disk drive.

1.5 When TOPS-20 development ends, provide sources to all customers

Digital has promised five years of development, followed by five years of maintenance. At the end of this period, Digital should supply sources to all frozen software free of charge. The five years of maintenance would then be carried out with a great deal of customer assistance.

1.6 Reduce HSC-50 price to TOPS customers

The HSC-50 controller for the RAXx series disk drives is priced high enough that KL customers must think twice before ordering them to expand their 36-bit systems. RP06 and RP07 drives are available at winning prices, and even the shadow of obsolescence does not justify the cost difference.

If the HSC-50 were cheaper, a more natural integration/migration path for hardware would encourage 36-bit customers to take integration more seriously.

1.7 TOPS third party license fee of \$80,000 is too much

Without mentioning S***** C***** by name, this menu item asks for an extended domain for the TOPS operating systems, by allowing these operating systems a new lease on life aboard another manufacturer's hardware. Digital presently does lease the rights to the TOPSes, but at a high price per CPU.

This menu item has a special feature. If you allocate votes to it, we ask that you supply an amount (also on the ballot form) which you feel is tolerable as a TOPS license fee.

1.8 Extend KL trade-in offer to other parts of the system

In New Orleans, Digital announced an offer of \$90,000 apiece for KLs traded in for VAX 8600s. However, no mention was made of other components of a KL10-based system which might be subject to the same offer, like for instance, MH10 memory, RP20 disks, and so forth. This menu item asks for a schedule of trade-in values for all components of a KL10-based system.

1.9 Extend KL trade-in offer to other upgrades, not just 8600

The \$90,000 trade-in offer for KLs should be broadened to include other upgrades and not just 8600 processors. The one-for-one rule may prevent some customers from benefiting from this offer.

1.10 Extend KL trade-in offer annually

The \$90,000 KL10 CPU trade-in offer was given a time limit. The submitter of this menu item felt that its expiration date was too soon for some KL10 sites, which will not be able to commit to a Digital replacement system until after that date. Digital is encouraged to make a continuing commitment to this trade-in offer, to enable longer-range planning for those 36-bit systems which are to remain in service for a greater time.

1.11 Separate maintenance charges for MF20 into its component parts

Maintenance charges may be saved, if the customer can perform maintenance by himself. In the case of MF20 memory, the actual memory circuits require a different order of maintenance from that required to keep the power supplies and controller alive. Digital is requested to break maintenance charges down into components, so that MF20 customers with an excess of extra memory parts can use those to maintain parts of their memories, and get Digital to perform the maintenance on the remaining parts of the MF20.

1.12 Improve PULSAR reliability

The interface between PULSAR and TOPS-10 does not handle errors by PULSAR very well. A PULSAR glitch will often cause the tape subsystem to become unusable or crash the system. This item asks for special attention to this important problem.

LS-10

Category 2

Monitor/Network

This section consists of menu items relating to the software of the TOPS-20 and TOPS-10 operating systems, and to DECNET-related issues. Issues in this part of the menu are not limited to requests to add functionality into the software, although that is the main thrust of most of these menu items.

The ANF-10 network is restricted to TOPS-10 systems alone. Other network issues include DECNET, Ethernet support for various networks, TCP/IP and other foreign network components.

2.1 Implement global ENQ/DEQ on clustered TOPS-20 systems

The ENQ/DEQ JSYSs are used primarily to provide process- or system-wide lock management. TOPS-20 V6.1 does not extend these JSYSs to provide cluster-wide lock management. One application for which ENQ/DEQ is frequently used is database record locking. DEC should extend ENQ/DEQ functionality to TOPS-20 clusters.

2.2 Add /COLUMN, /LINE, etc. to TOPS-20 TYPE

TOPS-10 systems have long enjoyed a TYPE program, from Catholic University, which can select part of a file. Vertically, it can confine its output to a starting, ending, or range, of disk blocks, pages, or lines. That is, one may TYPE/PSTART:10/LRANGE:12:15 for instance, if desired to type out only those four lines on page 10. Horizontally, one may select columns. Thus, TYPE FOO.LOG/COL:33:999 might type out a batch log file without the timestamps. These selection features should be added to the TOPS-20 TYPE command.

2.3 UDA-50-type interface for KL's; Cheaper than HSC

The UDA50 Unibus controller can drive RA60 and RA81 disks, and is cheaper than the HSC-50. For this reason, this controller is desired for KL systems which will not be clustered.

LS-11

2.4 Finish DECNET-10 DDP device implementation, to save DN20 cost

In 7.02, ANF-10 offers a DDP device type, which is a form of direct access to DDCMP packets. This means that any network built on top of DDCMP might be implemented in TOPS-10, and not just ANF-10. In particular, the DDP access to DDCMP might serve as a very inexpensive route to DECNET, bypassing the expense of a 128KW DN20.

2.5 TOPS-20 DELETE should allow DIRECTORY-like selection

Cleaning up a directory can be a morning's work, especially if the DELETE command is unfriendly. It would be very useful if the TOPS-20 DELETE command allowed the same sort of selection subcommands as DIRECTORY does. In particular, DELETE should allow users to specify files using parameters like BEFORE, SINCE, LARGER, SMALLER, etc.

2.6 Accounting functions on -10 should be kept in a single program

Account management functions have historically been housed in REACT. Now, however, we find some new functions in OPR. This menu item asks for the removal of these functions from OPR, putting them back into REACT.

2.7 Provide for EXEC based on username when directory is created

When one logs in to a TOPS-20 system, it would be highly desirable to have a system feature whereby different EXECs could be provided depending on the particular user logging in. This would permit students, for instance, to see a different EXEC from faculty. At present there is just one EXEC.

2.8 Prevent ctrl-C or modification of LOGIN.CMD on -20

There should exist a directory parameter that will cause TOPS-20 to trap control-C's while the LOGIN.CMD file is executing, and prevent the user from modifying the LOGIN.CMD file in that directory. The aim here is to protect the LOGIN process against a security gap by forcing a fixed LOGIN.CMD file to execute to completion.

2.9 TA78 support from TOPS-20 clusters

The TA78 tape drive is controlled by an HSC-50 normally, but the current HSC-50 support in TOPS-20 doesn't include the TA78. The older tapes are still supported, but they cannot be pooled in the cluster, as the disks can. This item requests TOPS-20 support for TA78 tapes on HSC-50 controllers.

2.10 Allow TOPS systems to boot from and swap to RAXx disks

With a bit of effort, the BOOT software should be able to read RAXx disks. Similarly, operating systems should be able to swap on these devices. We expect that over the next two or three years new still higher capacity disks will come along for the CI; it would be to everyone's advantage if the older disks could be completely phased out.

2.11 Supply MIC and PCL sources to source sites

TOPS-20 source sites currently do not get sources to MIC and PCL. Sources are desired, so that customers may have the power to repair and enhance the EXEC.

2.12 Network support from COPY command on -20

File transfer using the COPY command seems preferable to use of network-specific file transfer commands. The EXEC should understand that the presence of a node identifier implies a network file transfer and do the appropriate thing.

2.13 Partial recognition of leading parts of filenames

In TOPS-20, file name recognition is widely appreciated. It permits one to specify a file by its first letter or two, even when the name is long, if there is no other file with the same first letter or two. However, filenames whose names are identical except for the last few letters must be spelled out completely. It would be no great trick to enhance filename recognition so that the ESC key could be used to fill

in the early parts of a file name. Given two files named AB12345C.FILE and AB12345D.FILE, it would be useful to be able to name the latter by keying AB<esc>D<esc>. The present recognizer requires one to key in everything up to the point where the names differ.

2.14 User groups on 20 as SIXBIT words instead of integers

User groups on the DEC-20 are specified as decimal integers. It is not easy to relate these integers with specific user groups. Implementing user groups as SIXBIT words would allow meaningful names to be assigned to user groups.

2.15 Enhance date-time parsing in TOPS-20 to equal of TOPS-10

This is a request to enhance TOPS-20 software to understand date and time fields like YESTERDAY and NOON, in a manner familiar to TOPS-10 users.

2.16 Don't drop FACT files in TOPS-10

Reason: not everyone wants Usage file accounting.

2.17 Relative directory addressing in TOPS-20

It is easier to type <..foo> than to type <current.directory.foo>, and just as unambiguous.

2.18 Customer-defined terminal support in EDT-20

EDT-20 will be a very important piece of software on any TOPS-20 system. However, not all terminals are VT-100 compatible. Most functionality of EDT could be retained, over a much wider range of terminals, if terminal characteristics could be customized.

2.19 Implement PURGE command on TOPS-20

In order to be more consistent with DCL, TOPS-20 should provide a PURGE command that will perform the same function as DELETE with the KEEP subcommand does at present.

2.20 Allow GALAXY queues on non-PS: structures

TOPS-20 GALAXY maintains its queue file on the PS: structure. This structure is crowded enough. This item pleads for GALAXY queues on other disks.

2.21 Add AUTHOR specifier to DIRECTORY command

Give DIRECTORY the power to select only those files written by the designated AUTHOR.

2.22 Enhance Datatrieve-20 to support DBMS-20

Allow Datatrieve-20 to access DBMS-20 files.

Category 3
Layered Products

This section contains menu items which pertain to languages, utilities and other software described by Digital as Layered Products. There are in this section several items which might be equally well placed in the next section, Integration.

3.1 COBOL-8x on TOPS-10

Many TOPS-10 sites would like to have an up-to-date Cobol on TOPS-10. COBOL-10 should be brought to the same level as COBOL-20 v13, that is, to Cobol-8x standards.

3.2 Provide structured programming constructs in COBOL-10

If Digital does not feel that they can commit to implementing a full Cobol-8x level of COBOL-10, they should at least provide a subset of the modern structured programming constructs provided in the newer Cobol standards. This enhancement would give a great boost to the lifetime of COBOL-10.

3.3 Language Sensitive Editors for TOPS similar to VMS LSE

VMS now has a Language Sensitive Editor (LSE) that is based on the new TPU editor. This is a request for a similar product for TOPS.

3.4 Extended addressing in TOPS-10 COBOL, FORTRAN and LINK

The 7.03 TOPS-10 monitor provides for user-mode extended addressing. It is vital that extended addressing support be added to LINK and the languages, if this power is to be put to any productive use.

3.5 Extended addressing for TOPS-10 utilities like SORT, LINK, DDT

There is no question but that LINK must understand user-mode extended addressing, if any languages at all are to use it. This item asks for extended addressing support for SORT, DDT and BACKUP.

3.6 Tape and disk catalog for TOPS-20

This is a request for TOPS-20 implementation of the tape and disk catalog facility which is promised for TOPS-10 v7.03.

3.7 Multiforking DDT for TOPS

It is difficult to debug programs which create subprocesses. Digital should implement a version of DDT that makes this possible.

3.8 Add network file access support to RMS-10

This item requests that RMS-10 be enhanced to support network file access, as for instance from Datatrieve-32.

3.9 DBMS-20 should support remote access from DATATRIEVE-32

At present, only RMS files residing on a TOPS-20 system may be accessed from Datatrieve-32. This item requests that the support be extended to include DBMS-20 files.

3.10 Provide support for long symbols in TOPS compilers, DDT

Many compilers nowadays support long identifiers, in a few cases of indefinite length. If I call a variable LENGTH OF HOSPITAL STAY and another variable LENGTH OF HOSPITAL EMPLOYMENT, I would like to be able to debug using these symbols, in addition to knowing that LINK and DDT haven't confused them.

3.11 Autopatch should have REL files for vanilla sites

This is a simple enough idea for anyone who has spent a day trying to do a straightforward Autopatch on a crowded TOPS-20 system. Having just the REL files to replace earlier versions would make life so much simpler, and probably fit on less tape to boot.

3.12 LINK-20 should be made native

LINK-20 should be rewritten so as not to require the services of the PA1050 compatibility package. The resulting program would execute faster.

Category 4

Integration

4.1 Support a foreign tape utility for VMS

Over the years, TOPS users have managed to decode about all the magnetic tape formats there are. This has taken a lot of uncoordinated effort on the parts of many programmers, most of whom had to get an immediate job done when they wrote the programs to handle various tape formats. This item addresses that issue; it would be highly desirable to have a supported software product for VMS which can read (and to a lesser extent write) magnetic tapes written in formats not commonly found on Digital systems.

4.2 Create a HELP system on VMS based on TOPS keywords

The purpose would be to help attract TOPS users to VMS. Concepts such as TRMOPs, GALAXY, MOUNT requests, sub-directories etc. would be described in terms of their VMS counterparts.

4.3 Create a COMND JSYS equivalent as part of VMS

The amenities which TOPS-20 users hate giving up include the command completion and recognition capability of the COMND JSYS. These features have been imitated and given to the personal computer world through KERMIT, thus proving that it can be done. VMS should provide this feature, and in so doing avoid the proliferation which made such a mess of SCAN/WILD.

4.4 Implement a TOPS-20-like CLI for VMS

A Command Line Interpreter in the style of TOPS-20 would presumably understand abbreviations not only of commands, but also of file names. TOPS-20 users are very definite and vocal in their support of these features. Integration is considered by some an iffy thing if they can't have this comfortable feature on their next operating system.

4.5 Develop and support a TOPS-to-ULTRIX integration effort

For whatever their reasons, not everyone going from a 36-bit system to a VAX wants VMS. There may be other Unix-compatible systems in their shops, or they may simply prefer ULTRIX. Now that this seems to be a viable operating system, Digital might be persuaded to mount an integration effort similar to what they have done for TOPS-to-VMS. Votes for this item may persuade Digital to divert some integration resources into this area.

4.6 Produce an Integration Planning Guidebook

Integration has become such a corporate activity within Digital, and within many customers' shops, that it merits its own roadmap. After all, it is more than just a euphemism for migration. Sales literature may help convince, but it does little to help one organize one's planning efforts.

A great deal of documentation is available, on CUSP and tools tapes, in glossy publications, Digital newsletters, and paperbacks. Unfortunately, most such documentation is specialized, and is discovered only by accident.

Digital is asked to expend some literary effort to help customers get through the integration maze. A genuine integration planning guidebook might contain tables of relative capacities and speeds, of disks and tapes, CPUs and networks; language and other software compatibility charts; scenarios concerning the introduction of various configurations of VAXs alongside TOPS systems; use of DECNET and DIL to connect dissimilar systems; choice of tools for conversion of application software written in various languages; conversion of databases; translation of tape libraries; and construction of timetables. Helpful information like this would make for a stronger customer integration plan, and probably even make sales support people look better.

4.7 Create a GALAXY-like environment for VMS

Operators of large systems benefit strongly from facilities like GALAXY. A system whose users have no access to the computer room may have print queues, tape and disk mount requests, batch jobs and other work performed by an operator; almost unheard of in a small minicomputer shop. Large systems managers would like to see an organized design for management features, so that operators would not have to learn many different command processors, each of an ad hoc design. This menu item asks for a facility for VMS, comparable to GALAXY in purpose, power and cleanliness of operator interface.

4.8 Create Micro-VAX clusters on Ethernet or other medium

The clustering concept is so good that it ought to be an option on all sizes of VAX, not just the big ones.

4.9 Implement COMPIL-class commands in VMS

The TOPS systems support COMPILE, LOAD and EXECUTE commands which compile; compile and link; compile, link and run; respectively. The appropriate compiler is invoked based on the extension of the specified source file. For example, "LOAD TEST.FOR" would invoke the FORTRAN compiler, and subsequently LINK. The generic command qualifiers that may be specified with these commands (such as /DEBUG, /LIST, or /MAP) are passed to the compiler and/or LINK. These commands also check the default directory for existing object (.REL) files with the same first name as the source file. If an object file exists, and was written after the source file, the compilation step is skipped.

These commands are very convenient for both novice and experienced users. VMS should provide a similar functionality.

4.10 DEC-supported TCP/IP on VMS

TCP/IP has been available on TOPS-20 for years, and one or two homegrown versions exist for TOPS-10. There is a third-party version available on VMS. This item asks Digital to build and support such a product for VMS, since it is so useful. TCP/IP is used in private networks as well as on the ARPANET. Having these protocols available in a strong, supported form on VMS would be a real attraction to those considering a VMS system.

4.11 Add workload capability to SPM

SPM is a performance monitoring and capacity planning tool for VMS. It collects mostly system statistics such as CPU utilization, memory usage and I/O statistics. It collects some process information as well.

For capacity planning, a tool that measures, reports and summarizes the system workload is of vital importance. To adequately manage a large number of diverse users, a tool which can report system statistics grouped by process, image, UIC or a combination of these is needed. DEC should add this capability to SPM.

4.12 Give VMS MAIL the power to send to tree-structured usernames

It is not possible to send mail from VMS to a TOPS-20 user whose user name contains a period (as in MATH.SMITH). This problem should be fixed.

4.13 Project accounting on VMS with independent billing identifiers

VMS accounting software at present is quite a bit simpler than it is on TOPS systems. This menu item requests enhancement to the accounting system so that it can handle accounts with subaccounts.

4.14 Extend VMS usernames to 39 characters

At present, VMS usernames are restricted to twelve characters. There are users with longer names who feel slighted by the resulting truncation. Usernames might also serve as group or project identifiers, if the maximum length permitted it.

4.15 Improve tape/disk management in VMS

VMS has no media library management software. As a result, MOUNT requests offer no help to the operator in locating or validating tapes and disks. Apart from checking ownership of a tape, there is nothing to tell the operator where the disk or tape is located (yes, there are libraries containing tens of thousands of reels). If any media identifier is used other than a simple serial number, finding a reel may be quite a chore. Additionally, lack of complete facilities for aliasing and other operator dodges during mounting make it harder for instance, to mount an old retired disk pack, to copy something from it onto its replacement with the same name.

4.16 Enhance VAX TYPE command to include line and column ranges

Elsewhere, this same menu item appears on behalf of TOPS-20's TYPE command. It is a useful, businesslike item, asking for a facility which many users would find convenient. Among features which should be considered are the vertical ranges by block, page, and line.

Ranges by block are desirable chiefly in huge files, to save the CPU time needed to hunt through a file looking for page marks. A horizontal range by columns (adjusting for tabs), is very useful for files which get subjected to wraparound on 80-column terminals.

4.17 Provide timestamps in VMS batch log files

When a batch file is executed on a TOPS system, each line of the resulting .LOG file contains a time stamp which indicates when that line was written. This is a very useful feature that should be implemented in VMS.

4.18 VMS memory management in real time

VMS memory management appears to rely heavily on working set parameters, set for an account and for queues. Working set size should be automatically determined by the operating system according to process needs, current usage and overall system demands.

4.19 DEVELOP a TOPS-20 shell for ULTRIX

Support the question mark and use of guide words. Make it user changeable so it would be as powerful as PCL.

4.20 Support TCP/IP over TTY lines

TCP/IP is becoming an important network protocol. Support of TCP/IP over TTY lines would place a powerful networking tool at the disposal of a great many users who must now content themselves with KERMIT or worse. This plea is on behalf of all systems, TOPS, VMS and ULTRIX.

4.21 Low-cost VAX attachment for TU72 tape drives

Provide a way to add TU72 or other high-performance tape drives to a VMS system. Cost should be much less than for a DX20.

4.22 Support IBM standard label tapes (EBCDIC) in RMS

Power to read 'em and write 'em, translate to and from EBCDIC (with user-definable translation tables), and particularly to understand EBCDIC labels.

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

Mumps is Catching
by Mark Berryman
Copyright (C) 1985 Seldin Publishing, Inc.
Reprinted by permission

"Mumps? Isn't that just for medical applications?" I've heard that statement so many times I'm beginning to hear it in my dreams. But, if you think about it, a system that can keep track of complex medical records efficiently and inexpensively should be able to deliver the same results in countless other applications as well. Converting to Mumps may be a way to save a lot of money, while obtaining substantially more performance from your current hardware.

Mumps was developed in the late 60's at Massachusetts General Hospital by Neil Pappalardo and Curt Marble. Its name is in fact an acronym for Massachusetts General Hospital Utility Multi-Programming System. When Pappalardo and Marble left Massachusetts General around 1969, they formed a company called Meditech and unleashed their new language to the world. In recognition of the language's potential, a plethora of Mumps dialects sprang up, but in adding features developers also added incompatibility. This situation led to the formation of the Mumps Development Committee around 1972 whose task was to develop a standard Mumps language and submit it to ANSI.

By the time ANSI released its 1977 standard, the best of each of the Mumps dialects had been combined into one language and accepted as ANSI Standard Mumps. Unlike the confused swap of Unix versions for example, ANSI Standard Mumps has emerged as a reliable, consistent standard. To date, more than 35 vendors are selling ANSI standard Mumps for some 50 different computer systems. The Veterans Administration's File Manager, a data base management system available in the public domain, runs on every one of those systems without a single modification. Its doubtful that any other package written in any other language that can make the same claim.

When Pappalardo first conceived of Mumps, he envisioned an interpreted language that would execute as fast as a compiled language and one that treated data on disk as a simple extension to local memory. That second idea, the way Mumps uses the disk, is the biggest difference between Mumps and a typical business language. Mumps does not use files; there is no opening and closing of files, no space allocated in your program for buffers, no gets or puts. In effect, it has none of the overhead or coding headaches typically associated with file I/O. Instead, Mumps stores its on-disk information in the form of arrays called globals.

Globals are sparse arrays, permitting both numeric and string subscripts. They are stored on disk using binary trees or multiway balanced trees. If that sounds confusing, try the following approach: Start with your typical Fortran or Basic array. Now add the ability to have any number of subscripts. Next, add the ability to use string, or text, subscripts. Now remove the need to pre-allocate or DIMension the array, thus creating the sparse array.

Because space for a global is not pre-allocated, the global handler of a Mumps system only allocates space for nodes that are actually defined. So if you have a global named ABC with only the two nodes ABC(1) and ABC(10000) defined, you have not created an array 10,000

elements in size - rather the space required for the two defined nodes has been allocated. Global access is further enhanced through the use of a disk cacheing algorithm.

Let's take a look at a typical example. Suppose that you are a manufacturer and you want to set up a parts inventory database. Let's further assume that each part should be broken down into its subassemblies. Your main product is a widget, part number WID001, which consists of three subassemblies, part numbers SUB01A, SUB02A and SUB23C. The first few nodes in your global will look similar to:

```
INVENT("WID001") = DATA
INVENT("WID001","SUB01A") = DATA
INVENT("WID001","SUB02A") = DATA
INVENT("WID001","SUB23C") = DATA
INVENT("WID001","EB2") = "The elastic band that holds it together".
```

If this were an array you had built in memory, and you wanted the data associated with the widget itself, you would simply execute a statement similar to SET X=INVENT("WID001"). If the array was on disk the statement would be identical except that the array name would be prefixed with a caret (^) to indicate that the array resides on disk: SET X=^INVENT("WID001").

As the foregoing demonstrates, an index in a Mumps global is roughly equivalent to a key in an ISAM file. The data portion of the global can contain whatever information is appropriate and can be in as many fields as you need. Note that there is no pre-allocation of field sizes. The data is simply tucked away in the global while powerful text manipulation tools provide easy retrieval. If a given datum is only one character in size, then only one byte of space is allocated to store it. This fact, combined with the fact that most Mumps implementations use a data compression technique on the global subscripts, makes Mumps highly efficient with disk space.

Mumps is widely used throughout Europe, but seems to be confined mainly to medical sites here in the United States. This user base combined with its medical sounding name have contributed to the mistaken impression that Mumps is a medical language. Mumps really is a database system. In the real world it may be the closest thing to the ideal database system. Mumps manages your data base for you; all you need concern yourself with is the relationship between each datum. And because of the global structure and the disk cacheing techniques used by most Mumps implementations, a Mumps system is rarely I/O bound.

To keep all Mumps users in touch with each other and with the continuing development of the language, there is a nation-wide Mumps Users Group. This users group also co-ordinates with the user groups of other countries to effect a worldwide dissemination of Mumps-related data.

Surely, if Mumps is really as wonderful as I've portrayed it, shouldn't it be more widespread? Rather than losing points for its abilities, I believe Mumps has been shortchanged by political and financial considerations that often run counter to what's best for the user. Let's say you call up your local computer salesman and tell him you need a computer system that will meet the following criteria:

- a) It must support 30 - 40 online users

b) It needs to be able to maintain a parts inventory, accounts receivable, and sales order tracking.

c) You require a typical response time of two seconds.

The salesman will probably tell you that, in order to support that many users, and to allow for future growth, you will need a VAX-11/780. To support your data processing requirements, he will probably suggest you buy a package deal of his company's proprietary software.

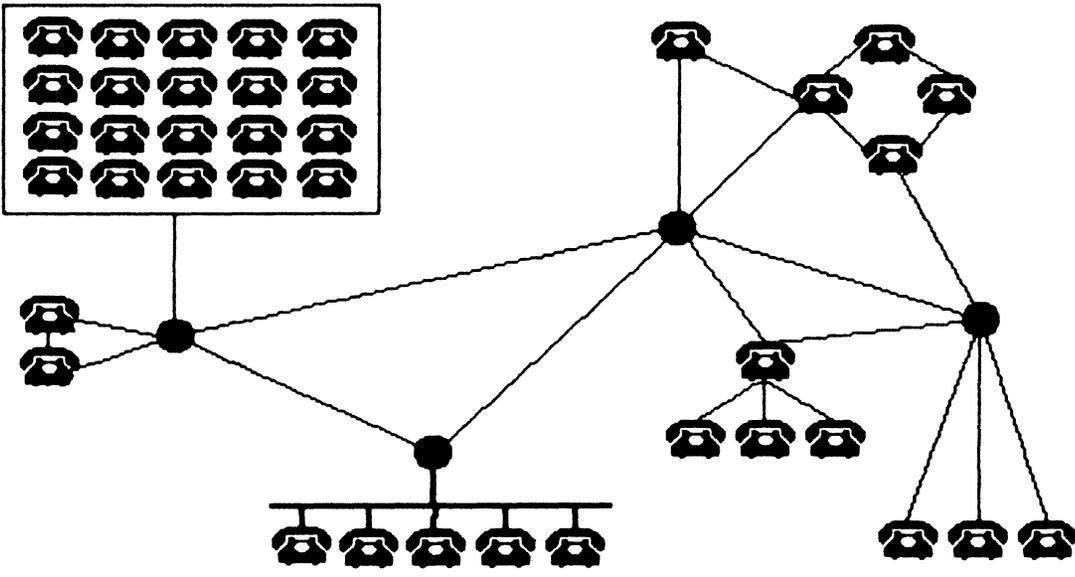
What the salesman won't tell you is that with Mumps you can accomplish the same thing using a much cheaper MicroPDP-11/73, a good performance disk drive of reasonable size and speed, and the VA File Manager package which, being in the public domain, is available for the price of a tape and a stamp.

He also doesn't usually tell you that if you purchase his VAX package you are locked into buying another VAX computer when it comes time to expand. With Mumps, on the other hand, there are numerous other vendors to chose from. When you compare how much a computer company makes when it can sell its own proprietary system with how much it makes when it sells a Mumps system, you get a clue as to why your sales representative would stay mum about Mumps (or maybe why his company never told him about it, either).

For years, Mumps has been one of medicine's best kept secrets, healing all sorts of data management ills. Like some miracle cure, it's time to let the world tap into the advantages of Mumps.

NETwords

The NETWORKS SIG Newsletter



Networks SIG Steering Committee

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

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

Gail Poulter
DEC Counterpart
Digital Equipment Corporation

Carol Greenfield
DEC Counterpart
Digital Equipment Corporation

Sandy Traylor
Symposia Coordinator
Target Systems, Inc.
(714) 779-0275

Bill Hancock
Technology/Standards Coordinator
Ki Research, Inc.
(214) 960-5674

Vickie Hancock
Newsletter Editor
(214) 495-7353

The Networks Special Interest Group (SIG) is one of 23 SIG's within the 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!

In This Issue:

NETWords Cover	NTW-1
Networks SIG Steering Committee	NTW-3
Index	NTW-4
Editor's Cobwebb	NTW-5
Vickie Hancock, NETWords Newsletter Editor	
Who is Doctor DECnet?	NTW-6
Network Troubleshooting and Other Mystical Things...	NTW-7
Doctor DECnet, The Ultimate Networker	
XEDRIVER Commonly Asked Technical Questions	NTW-11
John Heffernan, Digital Equipment Corporation	
DECnet and the High Energy Physics Network	NTW-17
Greg Chartrand, FERMILAB	
Philip Demar, FERMILAB	
Datagrams	NTW-20

The Editor's Cobweb



By Vickie Hancock

Welcome to the first combined issue of the DECUS SIG Newsletters! This particular issue you have received is for promotional purposes to entice you to subscribe to the newsletter. We on the Networks SIG Steering Committee urge you to subscribe to the newsletter and continue to receive the benefits of NETWords every month. NETWords, as you will see in this issue, contains many useful items for the network user, programmer, manager, or for those with an interest in getting to know more about computer networks. NETWords regularly includes technical articles, information on ECO's and FCO's, bug fixes and a wide variety of other network-related items of interest.

In this issue, you will find the first issue of Doctor DECnet's column of helpful hints, ideas, trivia, methodology, and surprises. We think that you will enjoy Doctor DECnet's unique methods of presenting technical information and encourage you to send in your comments, questions, and ideas to Doctor DECnet at the address listed in Doctor DECnet's summary.

Finally, we need your articles, ideas, and input. We rely heavily upon you, our readers, to provide us with publishable material for other DECUS members to learn from and enjoy. Send us your articles on any media (we're very resourceful) to:

Vickie Hancock
NETWords Editor
2510 Limestone Ln.
Garland, Texas 75040

See you next issue!!



NTW5

Who is Doctor DECnet?

By now, if you are a loyal NETWords watcher, you will have noticed that we have been mentioning that Doctor DECnet would be joining our publication soon. Well, soon is now! In this issue, you will find the first article from the ultimate networker, Doctor DECnet. The Networks SIG has commissioned Doctor DECnet to provide our readership with the best in network information, tips, guidelines, technical updates, and humor. But, you may ask, WHO is Doctor DECnet? A good question and worth answering.

Doctor DECnet was born of humble beginnings in Twisted Pair, Arkansas. He spent his formative years attending the Southern School for Advanced Network Design (located in Token, Florida), a rigid school known for its military strictness and reasonably good pepper steak on Saturday night. During this time, Doctor DECnet developed a severe allergy to IBM networks, which tends to affect him to this day, although he has been prescribed pharmaceuticals to help him when working with SNA and bisync protocols.

Following Doctor DECnet's preparatory training, he attended the University of the Divine Network in Boston. During that time, Doctor DECnet received undergraduate and graduate degrees in Advanced Network Quantitative and Qualitative Network Fixing, and culminated with a Doctor of Science in Network Design and Advanced Tinker-Toy Architecture.

Doctor DECnet spent many of his years as a professional network and systems hacker in Digital where he was responsible for popular products such as TRAX, the Pro-350, the PDP-2, DECnet Phase I and II, the 3270 protocol emulator, DMF-32 synchronous communications and many other useful and user-friendly products.

Now that Doctor DECnet is a full time consultant, writer, entrepreneur, and generally, a nice guy, we felt it useful to tap his vast wealth of knowledge and expertise. In return, the Networks SIG has offered to keep Doctor DECnet well supplied with Dr. Pepper, peanut butter, and friendly females. Doctor DECnet has graciously accepted our offer and will be appearing in NETWords now and in the future.

Doctor DECnet welcomes your questions, comments, ideas, and used comic books (preferably Archie and Spider Man). Good, useful, and thought-provoking questions will be answered in future issues (provided that certain bribes are included). Please send them to:

Doctor DECnet
c/o Vickie Hancock
NETWords Newsletter Editor
2510 Limestone Ln.
Garland, Tx. 75040

Finally, those of you who think that the individual responsible for this insanity is Bill Hancock, well, you're WRONG and we ain't tellin' who it is. So there!

NTW6

Doctor DECnet

Network Troubleshooting and Other Mystical Things...

Doctor DECnet Observation of the Month:

Noel Coward once defined a truly sophisticated woman as one who drives a car with her legs crossed.

Many times in my internship in networking (as interns are wont to do), Doctor DECnet would rush headlong into a problem - changing this network parameter and that network parameter without knowledge of the change effect or stopping, if only briefly, to think about the problem. These actions included wild brainstorming activities followed by radical "improvements." Time was lost from mistakes which occurred as a result of this brainstorming activity. It was during this time that the true meaning of NCP painfully came to view - the "Network Clobber Program."

As time passed, Doctor DECnet came to recognize the Rush Head Long (RHL) syndrome in himself as well as in others. This syndrome is well described in Kepner and Tregoe's (K&T) book [1] and is characterized by the following symptoms:

- o Nobody really knows what the problem is
- o Everyone has a different explanation of the network's problem, especially when the ubiquitous "NETWORK PARTNER EXITED" message is concerned
- o Another reason is suggested other than the fact causing the problem (unless it is DMF-32 microcode revision level problem, X.25 problem, or some other cheap shot that I could take)

After a rather long internship (at the customer's expense, of course) and many mistakes later (again at the customer's expense), K&T's approach to solving and analyzing network problems became quite appealing and cost effective for the good Doctor's customers. Utilizing the analysis techniques that all good doctors use, Doctor DECnet began to find symptoms and problems that appeared and proceeded to isolate, define, and categorize them. Through this methodology, certain problems came to the forefront and could be further refined as occurring in the following areas:

- o During installation (both hardware and software)
- o DECnet operations (i.e. during file transfer, task-to-task communications, terminal activities (PHONE, MAIL), etc...)
- o Performance degradation (e.g. system is slow and DECnet task synchronization is lost due to non-response by the partner within a certain time frame)
- o Hardware component failure

With the four general areas identified, the next step was to develop a systematic, stepwise

approach which could be applied to each area. In this article, Doctor DECnet will fill out a prescription for you so that you can use this methodology and not have to suffer another day of "network interruptus."

Evolving systematic approach in hand, Doctor DECnet approaches problems with the sanity of a well-trained emergency room doctor. The purpose is clear: not to give into the pressure of rapidly developing complex problems and pressures; rather, take a deep breath, wait 30 seconds or so, and start to apply each step of the four steps necessary in analyzing and solving the problem at hand. In this manner, each step refines the problem definition and makes the cause more apparent by eliminating extraneous, misleading, and contradictory information.

The first step, identifying the deviation or the user's lament into something useful ("It's broke; fix it!") is the simplest of the steps. The network problem to be solved is concisely and briefly stated. As an example of problem definition, consider the situation of a user performing a SET HOST operation to a remote node with an error message of 33 generated just prior to the operation being terminated. In the definition of the problem, you should identify the facility causing the problem and the effect upon that facility. In our example, SET HOST is the facility that is affected so a brief statement reflecting this needs to be made. How the facility terminated, error message 33, describes how the facility was affected. Thus, step one is complete: the problem facility identified, concisely and briefly, and what the known outcome is.

Step two is a bit more complex and is frequently known as the "Columbo" or "Sgt. Friday" approach: specify the problem in depth. This step is typified by clear, patient questioning in which you, the detective, are looking for "the facts, ma'am, nothing but the facts." Not causes or hints of causes. Simply, the FACTS. For instance, someone might say "the SET HOST problem has been occurring since the DEUNA was installed." The implication that the DEUNA could be causing the problem is clear in the statement and should not be considered as a potential problem specification as it identifies a potential CAUSE, not the PROBLEM, which is what you are after. In the SET HOST example, clear and patient questioning, much like is seen on TV's Columbo series or the Doctor's old favorite Dragnet, should center on the WHO, WHEN, WHAT, WHERE, and HOW of the problem. "Who is having the problem," "What is the type of account," "What are the quota limits," "What, if anything, could be found in the default non-privileged account's NETSERVER.LOG (if the node you are SETting HOST to is a VAX) that would give information about the termination of the operation," and "What was the user doing" are examples of the questions that should be asked to refine the problem in depth. WHERE is important as well - which processors were involved when the SET HOST was attempted. It is necessary to determine the path the user was attempting to link from the local node to the node the user was attempting a SET HOST operation to. In other words, the node the user was attempting to connect to might not be the one next to the node the user originated from. The path from one node to another might take the user through many nodes (how you figure out the path will be discussed in future articles given that path isolation can get tricky as paths can change frequently and radically). Other general questions might include the time of the day in which the problem occurred, load on the network and the processors involved, and other related topics. Remember that peak loading times can have a lot to do with many network problems, so keep in mind that many useful questions to ask may not have anything "directly" to do with the problem.

K&T discuss the concept of "bracketing" in their book. Bracketing is determining the IS NOTS as well as the IS's. For instance, YMS accounts with BYTLM of over 20000, such as the system manager, do not get the SET HOST termination error ("Columboing"), probing, virtual crumpled raincoat and stogie in hand). Also, it might be noted that accounts with lower quotas performing

DCL commands DIRECTORY, PRINT, or similar simple commands did not have the problem. So, identifying the IS NOTS helps to clarify and bracket (bound) the ISs. Just like Columbo, much time is spent defining, clarifying, and specifying the problem without jumping into the possible causes.

Step three deals with finding an expeditious way of dealing with the possible causes. Examining the distinctions and recent changes related to those particular distinctions only acts to speed the process of finding the possible causes of the problem(s). In our example of the SET HOST problem, the problem did not occur at any particular time or place. The major distinction between those accounts that had the problem and those that did not was BYTLM, BIOLM, and FILLM. The listing of the ISs and IS NOTS in case is fairly clear. In regard to changes, it would not be practical to list all changes that might have occurred and investigate each one. It is very expeditious to look at changes associated with the distinctions that have been listed. In the SET HOST example there happened to be no related changes to the distinction of account quota limits. Changes, as a cause, was eliminated. A specific statement should evolve from this step listing the possible causes, i.e. "Account quota differences may cause the problem when invoking utilities such as MONITOR and PHONE."

Step four is the roughest on the 'ole Doctor's ego. This step is the quest for truth and we all know how hard that can be. Possible causes are investigated to see if they account for each diminution of the problem. The SET HOST example has simply one possible cause. The quotas explain the IS and IS NOTS and are, in fact, the only explanation (go ahead and break that one down!).

Some pitfalls in searching for the truth include, first and foremost, the fact that many times too many assumptions are made - and we all know about "ass y me"ing. Secondly, not picking apart the possible causes for contradictions by thoroughly questioning all possible aspects. In searching for the truth in a galaxie, far, far away, one must not accept the explanation for the problem at face value. You should try and make it clear by noting it down. **DO NOT** look for a more complex rationalization to explain contradictions - eliminate it as a cause instead.

The final step is to verify the possible cause(s) that one settled on after the above stepwise refinement. K&T detail three possible methods:

- o Proof by logic (as you know, this does not necessarily apply to networks!)
- o Reality verification
- o Results verification

Proof by logic is pretty self-explanatory. Reality verification, however, requires a little more understanding. Reality verification is observing the network problem first hand. Watching the user perform a SET HOST and see what he may be doing to cause the problem can be very useful if the person might be contributing to the error. Results verification can also be observed by making a change that could result in fixing the problem and observing that the effects of the change actually fix the problem. Many folks start here as I did in my early days of networking, skipping the stepwise refinement. I remember one wintery, late Friday night when I received an urgent request for a house call. The network was down in a factory and all attempts to revive it were totally unsuccessful. This particular network was a DMP11-controlled multipoint network with DMY11's on the target processors. After many questions, the good Doctor DECnet remembered a similar problem and the solution - power down the YAX and PDP-11's (effectively resetting the communication hardware) and power them back up. At this point, Doctor DECnet did not really know what the problem was; only a possible and once-proven

approach existed and the good Doctor's bed was calling in the distance. Though it was late at night, Doctor DECnet has his wits about him, realizing that if the approach worked, the customer's first question would be "what caused the problem" or "what fixed it?" So, in the instructions to the customer, a step was included in which the customer was to perform a slow dance around the YAX system chanting "DECnet, DECnet, how great thou art!" before powering the YAX and PDP-11's back on. Later, that early Saturday morning, a joyous and relieved customer called back to say the network was working. His first question was "what fixed it?" and, knowing that I did not know (but not wanting to admit it), I told the customer that the dancing and chanting, of course! We all laughed and the subject was dropped. By the way, before I forget, the problem of the SET HOST was corrected by modifying the quotas of the user accounts and, upon observation of the results, the problem was corrected and the SET HOST facility worked correctly for all users.

So, to summarize:

- o Briefly list the problem
- o Specify the problem, especially the IS and IS NOTS of who, what, where, when, and how
- o Develop possible cause(s) by investigating the distinctions specified above in the problem specification and changes that may be related to these distinctions
- o Search for the truth - subject the possible causes to a thorough beating, all the while looking for contradictions and avoiding assumptions
- o Verify the cause(s) by observation or making a change and observing the results.
Remember that proof by logic in a network is frequently a contradiction in concepts - a fatal disease in California (I mean, like, totally).

Finally, remember Doctor DECnet's helpful hints in problem isolation:

On a YAX System -

DIR 0:: is the same as going into NCP and doing a NCP> LOOP EXE

DIR REMOTE:: of a remote node is the same as an NCP> LOOP NODE REMOTE

Finally, Doctor DECnet has been using these techniques for some time now and has found them to be useful and a very good way to find the "bug" in the ointment. I wish you success in your search for your problem.

Next issue: The installation and DECnet operations troubleshooting trees

[1] Kepner and Tregoe, Problem Analysis and Decision Making
New York: Princeton Research Press, 1979

XEDRIVER Commonly Asked Technical Questions

John Heffernan
DEC/ Networks and Communications
TWO/F18
1925 Andover Street
Tewksbury, MA 01876-1297

ABSTRACT: The VAX/VMS DEUNA driver is a complex driver with many capabilities. This paper answers some of the questions asked when starting to use the driver for advanced applications.

The author would like to thank Rod Gamache for his generous technical help utilized in preparing this document. Also, thanks to Gary McCoy for getting this paper out the door.

1. How does a read work? More specifically, must you have a read QIO posted all the time? My interpretation of the documentation is that up to `NMA$C_PCLI_BFN` packets will be buffered in system memory. In other words, no read QIOs are needed ahead of time. If this is true, when is the driver able to accept packets? At set mode QIO time? Are `NMA$C_PCLI_BFN` IRPs queued to the UCB's IRP queue at this time?

It is not necessary to have a read posted all the time. The driver will queue up to `NMA$C_PCLI_BFN` buffers to a receive queue of the UCB (Unit Control Block). When you issue a read, the first packet will be "moved" from the receive queue of the UCB to the IRP queue. Therefore, no read QIOs are needed ahead of time. The driver is able to accept packets from the wire after set mode QIO time. The IRPs are not queued to the UCB's IRP queue until a read QIO is issued.

There many ways to handle the read operation. One way is to use the AST mechanism. After you issue the set mode QIO, issue a read with AST. When a packet arrives, the system will deliver the AST to the process. This removes the overhead of moving the packet from the receive queue to the IRP queue. In the AST service routine, issue another read QIO with AST. Note that the process can hibernate waiting for the AST to be delivered. In the AST service routine, issue a wakeup request. In this way, the process can resume execution when the AST has been serviced. You can also have two read QIOs with AST posted to close the window in the AST process from the time the AST service routine gains control and

the time the next read with QIO is issued. This would probably only be needed if you expect a lot of packets coming in off the wire. Note that this approach is more efficient than keeping a long queue of packets in the UCB's receive queue and processing them at your leisure since `NMA$C_PCLI_BFN` times `NMA$C_PCLI_BSZ` bytes of nonpaged pool must be allocated.

In the multiple AST approach, watch out for using the same IOSB and user receive buffer. A common error is to reuse the same IOSB or receive buffer for multiple reads. The recommended value of `NMA$C_PCLI_BFN` is 3 or 4. Too many buffers result in heavy usage of system resources while too few might result in dropped packets. Also, note that the AST approach is better in an event driven type of application.

Another way to handle the read operation is to issue the read and wait for the specified event flag to be set. Finally, the IOSB can be polled for completion (however, this wastes CPU resources).

2. Since `NMA$C_PCLI_BFN` is specified on a per protocol type basis, is there buffer space for each individual protocol type? My assumption is that there is a static set of buffers in system dynamic memory that the driver and DEUNA share as their transmit/receive rings. In addition to this there are buffers allocated for each protocol type opened (number specified by `NMA$C_PCLI_BFN`) that are filled as the driver filters received packets by protocol type. Is this assumption correct?

There are buffers for the per protocol `NMA$C_PCLI_BFN` packets and 9 static buffers for receive data buffers. The 9 hardware buffers are queued off the CDB. These so called hardware buffers are in host memory and are the place where the packets first arrive (they sent via DMA from the DEUNA). There is also a pending receive queue off the CDB. The ring descriptor's data field points to a hardware buffers. On a read, after the own bit is toggled, the driver will be activated and "move" the buffer from the hardware buffer queue to the pending receive queue. Since the receive ring has 8 entries and the hardware buffer has nine, we see that the driver always tries to keep one spare buffer. Later, the packet is moved from the pending receive queue to the UCB. In this step, the driver figures out to which UCB or UCBs to deliver the packet. The driver does buffer shuffling, trying to stay ahead of incoming packets.

Note that the CDB is really an extension of the CRB. It is a structure defined by the XEDRIVER that describes the DEUNA controller. This is needed because there are many UCB's per

controller but only 1 CRB (and therefore CRB extension - called a CDB). There are many UCB's per controller because there is 1 UCB per protocol type, and there are many protocol types.

3. What happens to packets that cannot find a home? These include packets that were received with a protocol type that was not active and packets that could not be buffered due to lack of buffer space for the protocol type of the received packet.

Packets without a protocol type are dropped on the floor (assuming no promiscuous mode user). The same goes when there are not enough UCB receive buffers. The receive buffer unavailable condition refers to the CDB hardware buffers and could be noted by enabling the attention AST.

Note that the driver does drop packets - but it always increments a counter. So if a packet is dropped because of no protocol type, then the Unrecognized Frame Destination (UFD) counter is incremented. If the buffer is dropped because the user didn't specify a large enough BFN value, the User Buffer Unavailable (UBU) counter is incremented.

4. Is there any problem with having multiple protocol types within one image?

No, the driver had no problem handling this.

5. What is the meaning of the NMA\$C_PCLI_MLT set mode parameter? Is this a match all multicast parameter only or do you need to assert this parameter if you want the whole multicast system turned on? Is ten the limit for the number of multicast addresses that can be specified or does the driver handle the case of exceeding the hardware limit? What is the interaction between the NMA\$C_PCLI_MLT and NMA\$C_PCLI_MCA parameters, if any?

These are separate parameters. The NMA\$C_PCLI_MLT parameter is a match all multicast function. In this release, the driver can handle only up to ten multicast addresses altogether (among all users of the device). This is because the DEUNA can only filter ten addresses. If you want more, set NMA\$C_PCLI_MLT. This enables recognition of all multicast addresses without regard to the multicast list for this UCB. Likewise, NMA\$C_PCLI_MLT does not have to be set for recognition of those multicast addresses specified via the NMA\$C_PCLI_MCA parameter.

6. Is there any difference between an IO\$_SETMODE and IO\$_SETCHAR other than the fact that you need physical I/O privileges for IO\$_SETCHAR and only logical I/O privilege for IO\$_SETMODE?

The only thing the driver does is make the privilege check. The driver checks that the user has physical I/O privilege to go into promiscuous mode. The QIO system service checks that the caller has physical I/O to do a IO\$_SETCHAR and at least logical I/O for a IO\$_SETMODE. The I/O subsystem allows different characteristics to be set depending on the privilege. Usually, you need no privilege to set those parameters that affect only you (for example, terminal parameters) but do need privileges for those affecting the whole controller and hence other users.

7. Who can issue an IO\$_STARTUP to modify the device specific parameters? Does this have any relationship to the answer to the last question?

You need at least logical I/O privilege to do issue a startup.

8. How are protocol type and addresses passed to the driver. The picture on page 6-4 of the I/O User's Guide appears to me to be incorrect. True? I think that the high byte of the Ethernet address goes at the low byte of the VAX buffer. The picture shows the opposite view point. What is the best way to specify the these addresses in MACRO, in BLISS? Is the physical/multicast bit the zero bit on the VAX?

The picture is incorrect. See the programming example for specifying a Ethernet address in macro. Bit zero is the physical/multicast bit on the VAX. In MACRO the address AA-00-03-00-FC-01 is defined as shown below.

```
address: .BYTE ^XAA
         .BYTE ^X00
         .BYTE ^X03
         .BYTE ^X00
         .BYTE ^XFC
         .BYTE ^X01
```

In Bliss, the address AA-00-03-00-FC-01 can be defined as shown below (either way).

```
BIND
address = UPLIT WORD (%X'00AA', %X'0003', %X'01FC') ,
same     = UPLIT BYTE (%X'AA', %X'00', %X'03', %X'00', %X'FC', %X'01' ) ;
```

Note that an Ethernet address parameter is specified by giving the parameter constant word (for example, NMA\$C_PCLI_PHA), then the counted string (which includes the modifier word). That is, the modifier word is part of the counted string.

9. What happens if the NMA\$C_PCLI_BFN parameters is omitted? Why does the driver default to 1 if the parameter is required? What is the meaning of called a parameter required if it defaults with no error?

The parameter is not really required (no error will be returned) and will default to 1 in the version 4 driver.

10. A brief overview of driver operation would be useful. What does the driver do? What does the DEUNA do? What happens on a read operation? What happens on a write operation?

See the answer to 1 and 2 for a discussion of the read operation. A write is more straight forward. The packet is sent to the DEUNA via the transmit ring descriptor. The own bit is flipped and the driver requests that the DEUNA poll its transmit ring for messages to be transmitted. If DEUNA is busy, the packet is given to a transmit wait queue until another request completes.

11. A brief overview of the type sharing capability of the driver would be useful. How does it work? What are its limitations? What does the sentence "The shared default mode is the default user of a shared protocol type" mean?

This capability was originally provided for starting up DECnet MOM processes by the NETACP. Generally, a shared default user is started first. When a request from a node comes in, the shared default user sets up a process with the limited mode (shared with destination) channel. Note that one UCB is used for each protocol type. The driver filters on the received packet's source address. The shared default user is a catch all receiver. The limited mode users receive packets first if they are enabled for that address.

12. Is there any way to change the physical address of the DEUNA without shutting down? Must you always shut down to change parameters? Is this parameter specific?

The physical address can only be changed by shutting down. Some parameters may be changed without shutting down. They are promiscuous mode, multicast address and multicast state parameters.

13. What is the algorithm for assigning new unit numbers in the \$ASSIGN system service? Are they predictable?

Generally, the number is incremented. However, a scan operation is done since the counter could wrap around.

14. Any problems having multiple DEUNAs in a system serviced by the same driver?

This is no problem due to the I/O database being "separate" from the driver code. That is, there is a separate database for each module.

15. Under what conditions does the driver run self test? How long does it take? What happens if it fails?

The first user to startup the DEUNA will cause self test to run. This takes 8 seconds (6 for self test and 2 seconds in the driver to eliminate a race condition). If it fails, controller error or medium off line errors are returned in the IOSB.

16. The protocol type is a required parameter. What meaning, if any, does this have in promiscuous mode?

This has no meaning in promiscuous mode.

17. If we want to communicate with other devices, which have a different padding mechanism, do we turn the DEUNA padding off and pad the data ourselves? Is length part of the data field computed and added by the driver?

Padding must have to be turned off in the case of communicating directly other with devices that pad differently. Also, it must be turned off in promiscuous mode. Also note that promiscuous mode packets are delivered on a best try basis. When padding is enabled, the driver looks in the first two bytes of the data field for the packet length in received packets. For transmitted packets, the driver fills in the first two bytes with the data field length. If padding is disabled, the user is responsible for not sending runt packets (packets less than 64 bytes total).

SUMMARY: In this paper, the questions that typically come up when starting to code advanced applications of XEDRIVER are answered. XEDRIVER is complex piece of software and the driver has many capabilities. This paper should help those using those advanced capabilities by expanding on the documentation, and also by providing some insight into the internals of the driver.

DECnet

and the

HIGH ENERGY PHYSICS NETWORK

GREG CHARTRAND / PHILIP DEMAR
FERMILAB
P.O. BOX 500
BATAVIA, IL-60510

SUMMARY

The High Energy Physics community requires DECnet area numbers 41, 42, and 43 for its needs in physics research. In order to avoid conflicts in node identification we need your assistance and cooperation. A discussion and rationale for this requirement follows.

DISCUSSION

Over the last 5 years there has been a growing need for High Energy Physics (HEP) experimenters and theorists to interact and collaborate. For this purpose, university HEP departments have been installing computer networking circuits and facilities between each other and the major HEP laboratories across the country. This networking need has grown as the size and scope of physics experiments have grown. It is not unusual to have an experiment that involves a dozen different universities and laboratories located across the U.S., Europe, and Japan. This diversity of geography and the need for coordination among all parties has created the need for HEPnet.

DEC VAX systems have become very popular in the HEP community. Most of the university HEP departments own or have access to a VAX system. Additionally many experiments have VAX systems that are used in the actual experiment in an on-line sense. Some future proposed experiments consist of a hierarchy of VAX systems starting out with banks of MicroVAX'es networked to larger VAX systems. The utility of having the university VAX networked directly to the experimental VAX is obvious. HEPnet's role is to provide the backbone carrier for multiple users and protocols. DECnet's role is to serve as one of the major protocols that will be carried and distributed through HEPnet.

Currently many university HEP VAX'es have been DECnet'ed together on a cooperative basis to form an effective informal network, dubbed "PHYSnet". PHYSnet consists primarily of leased 9600 baud circuits linking approximately 50 locations and 300 nodes across the country. PHYSnet thus exists as a functional homogeneous network (utilizing DECnet) within a logical heterogeneous network called HEPnet.

The current topology of PHYSnet reflects the manner in which HEP research is conducted in this country. Major universities active in HEP research typically conduct experiments at large research facilities constructed solely for that purpose. This allows numerous experiments to share the same particle

accelerator facilities, permitting research that would otherwise be prohibitively expensive. The flow of data between physics departments at those universities and their experimental sites thus tend to "hub" at the major research facilities where those experiments are located.

The present PHYSnet contains two "hubs". One is Fermi National Accelerator Laboratory (FNAL) and neighboring Argonne National Laboratory, both located just west of Chicago, Ill. The two laboratories are connected by a high capacity microwave link, and for purposes of PHYSnet, may be viewed as one hub. The second hub is the Stanford Linear Accelerator (SLAC) and the University of California Lawrence Berkeley Laboratory (LBL), both situated in the San Francisco Bay area. As before, the facilities are connected by a high capacity microwave link, and may conceptually be viewed as a single hub. The two hubs are linked, creating what one might classify as a dual-star topology for the network. A third high energy facility, Brookhaven National Laboratory (Long Island, N.Y.) will shortly be connected to FNAL, effectively forming a third hub with an additional 20 - 30 nodes.

The existing 300 node PHYSnet is presently defined as one area with default address of area 1. Although the present allocation of nodes represents only one-third of the area capacity, several factors support the move to multiple areas within the immediate future. First, given the limit of 1023 nodes per area, the move to a multiple area network appears to be inevitable. Projecting node expansion based on past PHYSnet growth alone leads one to that conclusion. DEC's introduction of the MicroVAX II (and forthcoming MicroVAX III) intensifies that pressure for nodal expansion. Future on-line experiments are expected to utilize large numbers of these relatively low cost supermicros. Once the necessity for a multi-area network is recognized, the desirability of splitting the network as soon as possible is apparent. The transition from a single area network to a multiple area network would undoubtedly be easier with a 300 node network than with a network two or three times as large. Another factor to consider is the desirability of early establishment of PHYSnet's area designations. PHYSnet would then be in a more effective position to establish its area number designations in the world of scientific research, discouraging others from adopting the same numbers, and reducing the potential for future area number conflict.

Recently, a conference of the computer network departments for the "hub" facilities assembled for the purpose of addressing the issue of area routing within PHYSnet. The conference agreed that a transition to a multiple area network was both timely and in the best interests of the PHYSnet network. The principal discussions involved the practical facets of that transition. It was decided that the generalized structure of the multiple area PHYSnet would be a three area network, with the two major hubs of the existing PHYSnet (FNAL/Argonne and SLAC/LBL) each being assigned a specific area designation, and Brookhaven serving as a third hub with its own area. Individual facilities within the present PHYSnet would adopt the area address of the hub to which they presently connect (area designation represents logical connection, not geographical location). Existing node numbers and names will remain unchanged with the exception of the new area designation.

Discussion was given to "reserving" one or more area numbers for future use either by the European and/or Japanese physics communities, or to be held in reserve for future needs. It was decided that it would be impractical to reserve but not utilize an area number. European and/or Japanese area

designations would be addressed as they became necessary. Furthermore, the need for reserving area numbers for future use will hopefully be alleviated by the anticipated expansion of area numbers from 63 (current) to 255 or more.

One potential problem is the desire voiced by several large universities to have their own area designation, and still remain linked to PHYSnet. It is anticipated that this desire for personalized areas and PHYSnet links will spread to most of the major universities engaged in HEP research. This creates several problems for PHYSnet. First, there presently exists only 63 area numbers. As the number of universities arbitrarily designating themselves to be areas increases, so does the probability of a conflict between area/node numbers. Furthermore, other nodes within a given university network may access other networks (ie. the Chemistry VAX accessing CHEMnet), thus compounding the probability of such conflicts. This would create strong possibilities for traffic being partially or wholly misrouted and/or lost. Second, the problems of network security/integrity increase sharply if PHYSnet becomes accessible to university-wide networks. As a result of these anticipated problems, it was decided that PHYSnet should strive to maintain its homogeneity. Universities desiring to access PHYSnet will be required to utilize a PHYSnet-assigned node name and address (including appropriate PHYSnet area number) to connect to the network. If they choose to designate their own university-wide area, the cost of transmission between the PHYSnet area and the university area will have to be set to preclude transmissions either into or out of the PHYSnet network. This will provide at least some level of isolation.

The area designations for PHYSnet are listed below:

SLAC/LBL	41
FNAL/ANL	42
Brookhaven	43

Conflicts with other network area numbers will be increasingly difficult to avoid as the use of areas grow. For that reason, it is highly advisable that DECnet networks having contact with scientific research facilities and research oriented university environments avoid these numbers when selecting area numbers. In that manner, conflicts resulting from duplicate area addresses can be minimized.

DATAGRAM

1-86

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: SYSGEN PARAMETER TTY-DMA SIZE (VAX/VMS)

Message: The description in the manual (v.4) is correct. The cost for doing DMA matches the cost for interrupt servicing at about 32 characters on the DMF-32, according to some tests DEC has done. For fewer characters, the overhead of allocating a datapath and UMR's is too high; character interrupts are more efficient. The default parameter setting is OK. DECNET is normally configured using the sync line on the DMF-32. It uses a different driver, XGDRIVER, not the terminal driver. (Ref. Backup use). For screen applications, some users find better performance at 4800 baud, so that no XON/XOFF processing is required. QIO's with the noformat option always use DMA.

Your Name: Louise Wholey
Address: Measurex
1 Results Way, Cupertino CA 95014
Telephone: (408) 255-1500

If this is a reply to a previous DATAGRAM, what #? 2-85

Signature: Louise Wholey Date: 7-12-85

DATAGRAM

2-86

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: RMD Network Pages Crash System

Message: _____

Problem: If, on a RSX-11M v4.1 PHASE IV non-routing node, RMD is built with the network pages supplied with the unsupported software kit, asking to see the 'R' page will crash the system. Note that the 'R' page seems to work very well on a routing node, or with M+.

Answer: I SPR'd the problem (SPR 11-73063) and received a corrected version of [200,200]RMM.OLB and RMX.OLB, the object libraries used to build RMD with the network pages. The RSX-11M system no longer crashes, and the 'R' page now also correctly identifies adjacent nodes.

Your Name: Michael E. Mazzone
Address: Process Control Systems, Inc.
1300 S. Calhoun Rd. Brookfield, WI 53005
Telephone: (414) 782-3945

If this is a reply to a previous DATAGRAM, what #? _____

Signature: *M. E. Mazzone* **Date:** June 27, 1985

NTW-21

DATAGRAM

3-86

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: RAW Input Under VMS

Message: We have the VAX C compiler but have not found a way to allow RAW input under VMS. We need RAW input to allow control codes to be trapped and used under program control. We currently use RAW input for our microcomputer applications and want to migrate them to VMS.

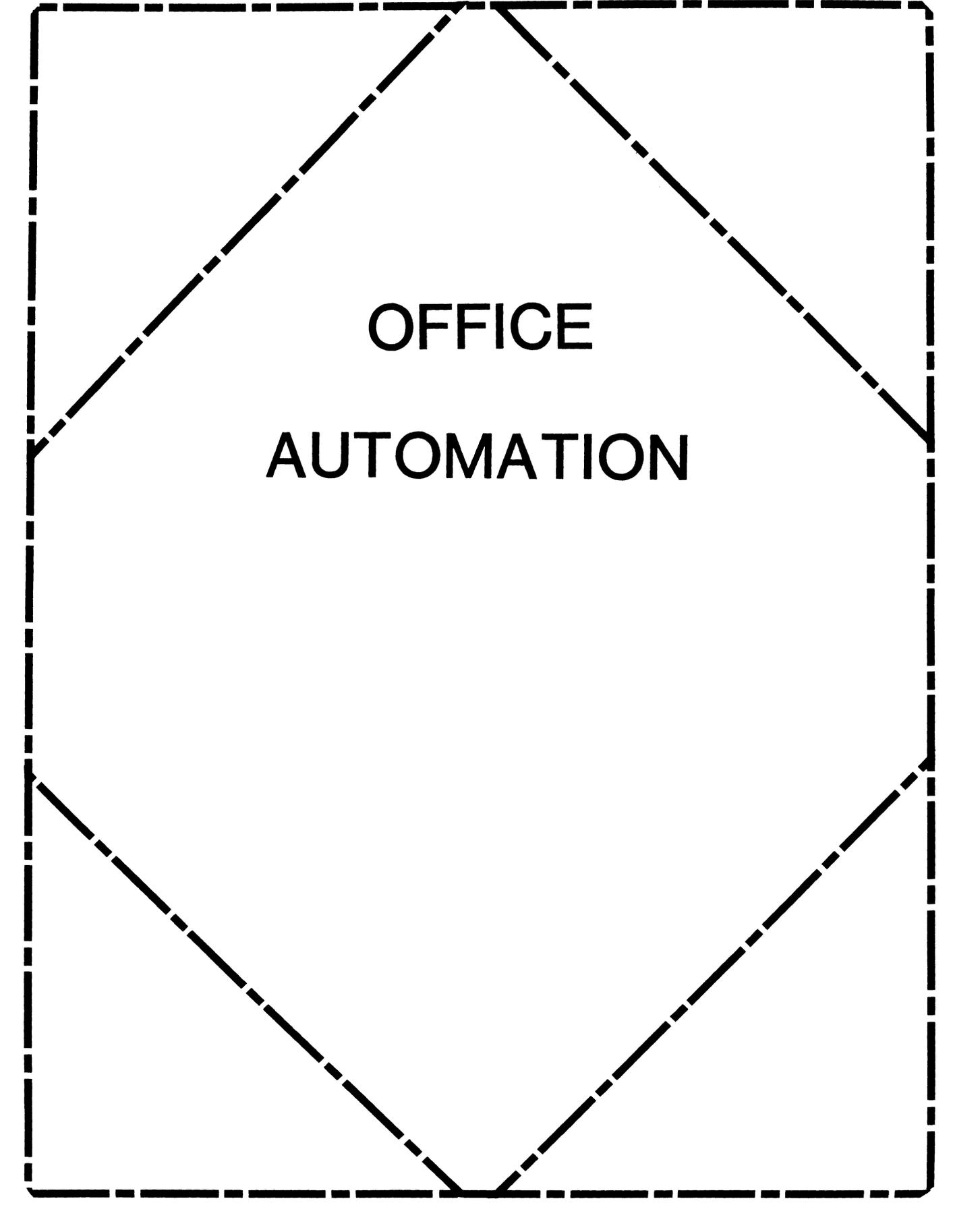
Please advise if this is possible and how we might proceed. There must be some VMS settings to allow this mode of input.

Your Name: Dr. John A. Jackman, Extension Entomologist
Address: Rm. 411-Entomology Bldg. [2475], TAMU
College Station, TX 77843
Telephone: 409/845-7026

If this is a reply to a previous DATAGRAM, what #? _____

Signature: *John A. Jackman* **Date:** 7/9/85

NTW-22



**OFFICE
AUTOMATION**

OFFICE AUTOMATION SIG STEERING COMMITTEE

Chairman

Thomas J. Orłowski
HQ Dept. of Army
Alexandria, VA

ALL-IN-1 Working Group

Leon E. Ottley
Evans and Sutherland
Salt Lake City, UT

Vice Chairman

Ralph Bradshaw
Johnson and Johnson
Raritan, NJ

Symposia Assistant

Sal Gianni
Northeast Utilities
Hartford, CT

Communications Committee Representative

Margaret Drake
University of Texas Health Science Center
San Antonio, TX

Symposia Activities

Mike Jackson
Air Force Operational
Test and Evaluation Center
Kirtland AFB, NM

Symposium Coordinator

Mitch Brown
Gen Rad, Inc.
Waltham, MA

Birds-Of-A-Feather

Larry Eisenberg
Encino, CA

Katherine "Kit" Trimm
Pivotal, Inc.
Tucson, AZ

Personal Computer SIG Liaison

Cheryl Johnson
Grinnell College
Grinnell, IA

Programs

Gene LeClair
HQ Dept. of Army
Washington, DC

Networks SIG Liaison

Gene LeClair
HQ Dept. of Army
Washington, DC

Newsletter Editor

Ray Kaplan
University of Arizona College of Business
Tucson, AZ

DECUS Europe OA SIG

Andreas Verbay
Telinco AG
Spiegelstrasse 20

Library

Bob Hassinger
Liberty Mutual Research Center
Hopkington, MA

Digital Counterparts

Les Agigian
Digital Equipment Corporation
Merrimack, NH

Feedback to Digital

Catherine Ditamore
ARA Services
Philadelphia, PA

Geof Bock
Digital Equipment Corporation
Merrimack, NH

FROM THE EDITOR

Just a brief note from your editor...as you can see, this is the first issue of a combined newsletter representing all the DECUS SIGs. Since we have been publishing four times a year and now have the opportunity to publish monthly, the size and perhaps the format of the newsletter will be changing. Please let us hear from you; with the new monthly format, we can have a much shorter turnaround time for getting news, views, and questions out to the OA SIG subscribers.

Thought for the day--What is the use of running when we are not on the right road?



CHAIRMAN'S CHAT

By

Tom Orlovski

The Office Automation (OA), Special Interest Group continues to provide a variety of programs to facilitate information exchange on topics in this exploding area of interest. There are already over 5,000 members of our group including senior executives in large corporations as a growing segment of our primarily management focused organization. We also provide technicians and implementors with a forum which meets their needs.

Using different tools, such as this newsletter, we provide each group with the opportunity to establish or expand his or her personal OA information network. In conjunction with over fifty seminar presentations at the semi-annual DECUS Symposia we use these pages, an office automation program library tape, working groups, like our All-In-1 group, and our Senior Executive/Management Days (co-sponsored by Digital) to provide a continuous flow of information. Whether your needs involve multinational corporations or just a single person behind a micro-computer we'll help answer your questions. In addition, work has begun to form the first office automation local users group.

We use many different forums for information exchange because office automation has so many facets which impact on everyone from senior corporate executives to the operator of a single person business. Because of this diversity, the best forum for most people is this newsletter. It is your place to get the latest information on Digital's Office Automation world and the place to ask and answer questions or establish your personal network. I encourage you to join us as we meet monthly in these pages.

OFFICE AUTOMATION SIG NEWSLETTER

IN THIS ISSUE

FROM THE EDITOR.....	2
--margaret drake	
CHAIRMAN'S CHAT.....	2
--tom orlovski	
SYMPOSIUM COORDINATES.....	3
--kit trimm and mitch brown	
THE FORUM	
ALL-IN-1 QUESTIONNAIRE.....	4
OASIG SWAP TAPE.....	7
--ray kaplan	
THE OASIG SIR.....	9
--catherine ditamore	
ALL-IN-1 IN ACTION.....	21
--ray kaplan	
FEATURE ARTICLES	
ALL-IN-1 V2.0 SYSTEM MANAGEMENT.....	24
--dave shove	
12-BIT WORLD.....	28
--bob hassinger	

ALL-IN-1 QUESTIONNAIRE

1. How long have you had your ALL-IN-1 system?

- Less than one year
- One year
- Two years
- Greater than two years
- On order, not yet installed

2. What overall objective is your ALL-IN-1 system designed to achieve? Check the one that best applies.

- Increase sales
- Reduce costs
- Provide better customer service
- Make better business decisions
- Facilitate communications
- Other Please specify: _____

3. About how many people use your ALL-IN-1 system?

4. What is the primary application for which the system was purchased? Please check all that apply.

- Text/Document processing
 - Electronic mail
 - Information management
 - Decision support
 - Application solution
- Check area of solution:
- Engineering/Research
 - Business/Administrative
 - Manufacturing
 - Other Please specify: _____

5. What is the secondary application for which the system was purchased? Please check all that apply.

- Text/Document processing
 - Electronic mail
 - Information management
 - Decision support
 - Application solution
- Check area of solution:
- Engineering/Research
 - Business/Administrative
 - Manufacturing
 - Other Please specify: _____

(Please turn over)



6. How did your company justify purchasing the ALL-IN-1 system?
Check the one that best applies.

- Cost savings
- Productivity increase
- Office automation pilot
- Other Please specify: _____

7. Order the purchase decision influencers below from highest to lowest ("1" is highest, "2" is second highest, etc. Enter "0" if no influence).

- Department management
- Users
- MIS
- Corporate management
- Other Please specify: _____

8. Please rate the following aspects of ALL-IN-1 on a scale of 1 to 10, with "10" being the best rating and "1" the poorest.

- Functionality
- Number of users
- Response time
- Ease of use
- Documentation
- Support

9. What areas of ALL-IN-1 would you most want addressed by Digital in the future. Use a 1 to 10 scale, with "10" being most important and "1" being least important.

- Functionality Specific areas: _____
- Number of users
- Response time
- Ease of use
- Documentation
- Support

10. What is your affiliation with ALL-IN-1?

- Systems management
- Technical user
- Office/Commercial user
- Management
- Other Please specify: _____

We appreciate and thank you for your cooperation.



The OASIG Symposium Swap Tape is Alive!

By Ray Kaplan
Self Appointed OASIG SIG
Symposium Swap Tape Person

The New Orleans DECUS U.S. Chapter Symposium marks the production of the first rendition of an Office Automation SIG Swap Tape. As with other DECUS activities, this one is running on a stream of volunteer energy!

Jon Arnold, a DIGITAL Senior Software Specialist, Mitch Brown of GenRad, and Ray Kaplan of PIVOTAL, Inc., combined forces in late April to begin the promotion of an Office Automation SIG Symposium Swap Tape effort. The initial promotion effort was simply a bit of a "word of mouth" campaign soliciting contributions of material for the tape from people interested in things that surround the DECUS U.S. Chapter Office Automation SIG. As it turns out, most of the current interest surrounds DIGITAL's office automation product, ALL-IN-1. As a result, the first OA SIG Symposium Swap Tape contains only ALL-IN-1 software. It is hoped that future tapes will contain software from all of the various office automation interests that make up the OA SIG.

Contributors to the first OA SIG SIG Symposium Swap Tape include:

Martha Rudkin
GMF Robotics
ALL-IN-1 V1.4 to VAX/VMS V4.0 Mail transfer
"Set Host" ALL-IN-1 application
ALL-IN-1 Shared Mailing List application
ALL-IN-1 attached printer application

Mitch Brown
GenRad Corporate MIS
"Print it there" ALL-IN-1 application
CMI Net Mail monitor for ALL-IN-1

Jon Arnold
DIGITAL
From FORD Motor Co.
Watchdog idle job killer
From Jon himself
Local LA50 printer routine
Version 1 file cabinet utility

E. Catherine Ditamore
ARA Services
Access restriction procedures
IBM/XT/AT and ALL-IN-1 file transfer
Search, delete, and refile of ALL-IN-1 documents
LA100 and LQPO2 printer control routines
Version 1 file conversion routines

Michael Jackson
U.S. Airforce
"When read" mail message system
Escape, and sheet feeder procedures

A big cheer of thanks to Jon Arnold for his efforts in running around and collecting submissions from his customers, and helping to promote the tape. Jon is a Senior Software Specialist in the Framington Hills, Mi. DIGITAL office. Without his efforts, the first OA SIG Symposium Swap Tape effort would have been a lot thinner, and a lot harder to make. WE HOPE TO CONVINCe OTHER DIGITAL FOLKS (New Hampshire, Massachusettes, North Carolina, and Redding, England) TO CONTRIBUTE TO FUTURE TAPES.

At this writing, the SIG tape is being edited and assembled by Ray Kaplan, and is expected to be put into the National LUG Organization distribution tree the week of June 17, 1985.

If you are interested in submitting to the Anaheim rendition of the tape, just bring your submissions along to the Symposium! The Library Committee reminds us that you should take the time to make an "official" DECUS Library submission out of your OA SIG Symposium Swap Tape submission. The Library requires an abstract, a directory listing, sources on magnetic media, and a signed Library Submittal Form. Since you get a nice placque for your Library submissions, it is worth your time to make a Library submission out of your OA SIG Symposium Swap Tape submission!

If you can't come to the Symposium, you can submit your software to:

Ray Kaplan
PIVOTAL, Inc.
6892 E. Dorado Ct.
Tucson, Arizona 85715

Remember, like everything else in DECUS, if you have to contribute to it to keep it working.

See you in Anaheim!

ANNOUNCING: THE OA SIG SIR

By: E. Catherine Ditamore
ARA Services
Philadelphia, PA

What is it? A terrific new opportunity for us and Digital! The OA SIG has implemented the System Improvement Requests (SIR) process. It provides us, the user community, with a formal mechanism to tell Digital what we need. Digital is interested in our input and will be giving us some feedback!

Since you'll find SIR submission forms and ballots in this newsletter, I've tried to describe how the whole process works. It's very similar to the one used by the VAX SIG -- my thanks to Gary Grebus at Battelle Columbus Laboratories for his good ideas.

The SIR process consists of several phases: the distribution and collection of forms; consolidation of the forms for publication in the newsletter; prioritization of SIR's via user balloting; tallying of those ballots; submission of the prioritized SIR's to Digital; and, response from Digital to the SIR's.

Each phase of this process is discussed in detail below, starting with the content of the SIR form itself and concluding with Digital's response to the SIR's.

SIR FORMS

The SIR form falls into four categories: Submitter Information, SIR Category, SIR Definition and SIR Description.

The Submitter Information will provide a means of researching or clarifying SIR's.

The SIR Category is intended to assist us in classifying the SIR's. The OA SIG is fairly unique among the DECUS SIG's in that it covers a variety of product types, ranging from hardware to software. In preparing the SIR's for balloting and submission to Digital, it will be necessary (for clarity's sake) to categorize them. Further, the submitter may not make clear which device or software product the improvement is intended for; for example, an improvement to WPS could be intended for the DECmate, Rainbow or ALL-IN-1! Therefore, the category section is intended to minimize confusion once we receive the SIR's.

The section marked ABSTRACT is provided to allow for a brief definition of the improvement.

The section marked DESCRIPTION is provided for a detailed description of the improvements, as well as examples of usage. If possible, we'd like the submitter to justify its usefulness. It is important (as seen in the INSTRUCTIONS) to be specific, rather than assume that we are familiar with a variety of other office automation products.

SIR DISTRIBUTION AND COLLECTION

The vehicle for SIR distribution will be the OA SIG Newsletter and the Symposium. The SIR form will be included in the newsletter on a regular basis. All you have to do is fill it out and mail it!

The SIR forms will also be available at the Symposium during the Roadmap Session and at the OA SIG Campgrounds. The campgrounds will also have a collection box for completed forms. Additionally, the ideas presented at the now twice-popular Symposium session "Wish List" will result in SIR's.

SIR CONSOLIDATIONS

As the SIR's are received they will be numbered and grouped by category. The number will be used for balloting and will be discussed below in the section entitled SIR Ballots. Within each category, the SIR will be reviewed against previously received SIR's; if it is similar to another SIR, the two will be consolidated into one. Additionally, the SIR's will be reviewed for completeness and clarity; if anything is questionable, the submitter will be contacted.

SIR PUBLICATION

The consolidated SIR's (with assigned numbers) and the ballot will be published in the newsletter at least two months prior to the Symposium. This will allow time for the users to review the SIR's, mark their ballots, and return them for tallying. The tally must be completed such that the prioritized list of SIR's can be forwarded to Digital four weeks before the Symposium.

SIR BALLOTS

The SIR Ballot requires three pieces of information: your DECUS membership number, SIR numbers and points. The DECUS membership number will be used to ensure that each member submits only one ballot. Points are assigned to SIR's by putting each SIR number of interest in the column labeled SIR Number and the desired point value in the corresponding column labeled Points.

Each voter is allowed a total of 100 points; these points may be allocated in any quantity desired, and in either a positive or negative sense. A high positive value would strongly encourage the SIR; a low negative value would discourage the change. For example, if the positive points total 80 and the negative points total 20, the voter has fully utilized his allowed 100 points.

SIR TALLY

As the ballots are received, the positive and negative values for each SIR will be recorded. The net of the two values will be used as the rank for each SIR. On the cut-off date, as specified on the ballot, the SIR's will be re-ordered, from the highest rank to the lowest.

Ballots received after the cut-off date, without a DECUS membership number or exceeding the 100 point total will not be included in the tally.

SIR SUBMISSION TO DIGITAL

The complete prioritized SIR list, resulting from the SIR Tally, as well as the tally itself, will be forwarded to Digital four weeks prior to the Symposium; this will allow Digital sufficient time to formulate responses to the SIR's.

DIGITAL RESPONSE TO SIR

Digital has agreed to review the SIR list and take it as input to their own development process! Further, at the Symposium, Digital will indicate which items on the list have been incorporated into new products that will be shipped or announced by the date of the Symposium. Subsequent newsletters will carry this information to keep those DECUS members not attending the Symposium current with the OA advancements.

CONCLUSION

Now you know how the SIR process works -- it all depends on you. So, start filling out submission forms with all those great ideas I know you have! And, check out "Time to Vote!" also in this newsletter for the first SIR ballot.

TIME TO VOTE !

The SIR process is just getting off the ground; therefore, the SIR list is fairly abbreviated. Also, our only SIR's are the result of the Wish List session at the New Orleans Symposium. You will notice that there is only an abstract (no description) and that the SIR's are heavily skewed to ALL-IN-1. I envision these characteristics changing as you, the user community, begin to submit your own SIR's.

But, in the meantime, your vote NOW is important! Please return your ballot as soon as possible to allow Digital time to respond. Ballots received after NOVEMBER 8 cannot be counted. The results of the voting and Digital's responses will be given at a session at the Fall 1985 DECUS Symposium in Anaheim.

The SIR's are grouped by category. You have 100 points to allocate among the SIR's on the ballot. You may assign any point value in either a positive or negative sense: a high positive value would strongly encourage change, and a low negative value would discourage the change. For example, if the positive points total 80 and the negative points total 20, the allowed 100 points have been fully utilized. Remember, only one ballot per DECUS member will be accepted!

ALL-IN-1 (Electronic Mail/Messaging)

SIR #	Abstract
185001	The capability to delete or file before reading or selecting a message.
185002	The capability for system-wide nicknames.
185003	A mass delete function for the mail messages.
185004	Provide the ALL-IN-1 system manager with the ability to manipulate mail messages.
185005	Add an integrated phone message function.

ALL-IN-1 (Editor)

SIR #	Abstract
185006	The capability to change or add tab stops in the DPE editor.
185007	A search capability in the editor that will allow you to search for embedded punctuation, tap stops, etc.

ALL-IN-1 (Printing)

SIR # Abstract

- 185008 Incorporate the following printer enhancements:
1. Full LNO3 support.
 2. Standard type fonts for the LNO3.
 3. Stop print queue.
 4. Host control of your DECmate printer.
 5. A selection form for printing to select various options (ie. what's available on the DECmate Print Menu).
- 185009 Treat DPE or WPS+ documents the same when printing (i.e. the same print settings, the same printers available, etc).

ALL-IN-1 (Time Management)

SIR # Abstract

- 1850010 Do not allow meetings that you have declined to attend to continue to be posted on your calendar.
- 1850011 The capability in your calendar to schedule conference rooms, audio visual devices, etc.
- 1850012 Provide a timer or alarm system to remind you of meetings or appointments. This timer or alarm should go off a few minutes before the meeting or appointment.

ALL-IN-1 (File Cabinet)

SIR # Abstract

- 1850013 Search capabilities that allow you to search through your memos and documents:
1. By phrase or full text database search.
 2. By time frame: either within a range of dates, before a date or after a date.
 3. For the addressees on the memo.
- 1850014 Provide real shared file cabinet facilities, especially with a thought towards teams or groups.
- 1850015 Provide an option on your file cabinet menu to allow the transfer of an entire file cabinet folder to another user.
- 1850016 Provide a DEC-supplied archival system for ALL-IN-1.

ALL-IN-1 (Customization/Applications Integration)

SIR # Abstract

- 1850022 Expand the user defined process (UDP) capabilities to incorporate loops, and the ability to stop during processing to obtain user input.
- 1850023 Include an additional 80-bytes in the user profile for customer specific use.
- 1850024 Promote the ALL-IN-1 applications development capabilities by providing the following:
1. A DSAB for Rdb applications.
 2. Easy end-user application development facilities.
- 1850025 Better integration of ALL-IN-1 with the following product:
1. DECslide
 2. DECgraph
 3. DECcalc

- 1850026 Have other applications that integrate with ALL-IN-1 use the ALL-IN-1 keypad. (i.e. VTX).
- 1850027 Provide customization capabilities for the MicroVAX, rather than forcing customization on a larger VAX for downloading to the MicroVAX.
- 1850028 Offer a pre-DECUS symposium on ALL-IN-1 applications integration and development.

ALL-IN-1 Aids

SIR # Abstract

- 1850029 Provide a conversion utility telling you what's been changed in your current ALL-IN-1 system.
- 1850030 Firmly establish a DEC policy that there will be no uni-directional conversions. (i.e. from the old DX to the new DX). In other words, provide a bi-directional conversion for all products where conversions are required.

ALL-IN-1 System/Security

SIR # Abstract

- 1850031 Access control on the business applications menu that would do the following:

1. Allow a user to know what applications are available to him/her under ALL-IN-1.
2. Provide a mechanism to specify the security levels available to each user.

1850032 Develop a networkable pocket VAX that has ALL-IN-1 running on it.

WPS

SIR # Abstract

1850017 Clean up and simplify the CX interface for the DECmate; make it more user friendly.

Generic Communications

SIR # Abstract

1850018 Provide an updated office interconnect guide that will clearly and easily outline what capabilities work in what environments.

1850019 Provide full intelligent modem support, including looping and if-then-else capabilities.

New Office Workstation Software

SIR # Abstract

1850020 Put the PRO Office Workstation software on the MacIntosh.

1850021 Full office workstation support for the IBM PC.

Miscellaneous

SIR # Abstract

1850033 The Atlanta Hotline should provide comprehensive user support. If the Atlanta Hotline cannot answer a question and feels it is a responsibility of Colorado Springs, Atlanta should contact Colorado directly and pursue the problem with Colorado, rather than tell the customer to call Colorado.

1850034 The Atlanta Hotline should provide comprehensive user support. If the Atlanta Hotline's response to a customer problem is that there is "no work around" then Atlanta should generate the SPR for the problem.

OFFICE AUTOMATION SIG SYSTEM IMPROVEMENT REQUEST SUBMISSION FORM

Name _____ Address _____
 Firm _____
 Telephone _____

INSTRUCTIONS: System Improvement Requests (SIR) can be either hardware or software; please check the category addressed by this SIR. Under **ABSTRACT**, give a brief definition of the capability you would like. In the **DESCRIPTION** section, give a detailed description and examples of what you want. Be specific; don't assume that we know how other products function. Justify the usefulness of the capability and give an example of its use.

HARDWARE IMPROVEMENT

SOFTWARE IMPROVEMENT

DECmate _____	ALL-IN-1 _____	WPS _____
PRO-Series _____	CP/M (DECmate) _____	P/OS _____
Rainbow _____	CP/M (Rainbow) _____	MS-DOS _____
Other _____	Other _____	

ABSTRACT _____

DESCRIPTION _____



E. Catherine Ditamore
ARA Services
Corp MIS
Independence Square West
Philadelphia, Pa. 19106



E. Catherine Ditamore
ARA Services
Corp MIS
Independence Square West
Philadelphia, Pa. 19106



ALL-IN-1 IN ACTION

ALL-IN-1 Hints and Kinks

by
Ray Kaplan
PIVOTAL, Inc.

ALL-IN-1 Hints and Kinks? Sure! This is number next in a series of an indeterminate length. This is intended to be a collection of things that you can use to make your "ALL-IN-1 life" more productive and fun.

Since I keep on getting bills in the mail, I know that my mailbox is not broken. I will just assume that you haven't taken the time to drop me a line with your favorite questions/comments/problems. Please do. We all want to hear your side of things.

Please send things that you have:

Ray Kaplan
PIVOTAL, Inc.
6892 E. Dorado Ct.
Tucson, Arizona 85715
(602) 886-5563

As usual, I continue to "relive" the New Orleans Symposium through my notes, the Audio tapes of the various sessions, and various followup work such as the OA SIG Tape production issues.

News You Can Use

One of the best things about version 2 of ALL-IN-1 is the new documentation. DIGITAL has invested over 17 person-years in it, and it shows. The scant version 1.X documentation was rumored to have been done in 6 person-months!

In case you missed it, the documentation folks came out to the New Orleans DECUS Symposium and gave a great PreSymposium Seminar on using the new Customizable Documentation kit to generate custom documentation. Sue Franklin, Donna Duncan, Lydia Velez, and Sarah McGillis presented a 1 tutorial full of details about how the kit works. Thanks folks, a great job.

You should keep your eyes on the OA SIG Pre-Symposium Seminars. They are advertised in the Symposium registration packet that comes in the mail to all DECUS U.S. Chapter

members. You should be getting one in the mail for the Anaheim meeting in December very soon. Based on the rumors floating around, we stand to have an unparalleled selection of absolutely dy-no-mite seminars in Ahaheim. We expect to see some user presented seminars, and some ALL-IN-1 developer presented seminars. I am going to try to offer one called "Why ALL-IN-1 is NOT Office Automation".

As an example of other things that are offered, the VAX SIG is sponsoring my friend Louise Wholey from Measurex Corp. in Cupertino, Ca. in an all day VAX/VMS SYSGEN parameter discussion. In all, these little jewels are some of the best training opportunities that I know of.

Get This Book!

There is a complete "Integration" documentation set in the works. It will eventually consist of three volumes. A User Interface Standards manual, a TEXT DSAB manual, and a DATA DSAB manual.

The User Interface Standards are the guidelines that were used to maintain the consistent look of the version 2 form faces.

DSABs (Data Set Access Blocks) are the way ALL-IN-1's functional code can remain independent of the actual data and text that it is dealing with, and as such are the very heart of interfacing ALL-IN-1 to just about ANYTHING that is outside of ALL-IN-1. A good example is the possibility of having an ALL-IN-1 user "editing" what they think is a "local" file, when in reality they are actually "editing" a stream of bytes which is coming/going to/from some other system over some communications link. Exciting stuff.

So far, ONLY the User Interface Standards manual is available. I have a copy, and the order number of the book is AA-EG03A-TE. It is called The Office Menu User Interface Standards manual. I called DECdirect to try to get a price only to find that it is not listed. I called the local DIGITAL office, and they could not find it in the July, 1985 documentation catalogue. When I called my DIGITAL OIS friend that sent it to me, he indicated that it was available with an order number of QLA05-GZ. I called DECdirect with that number and they quoted me around \$50.00. I think that this QL number might be the order number for the whole "integration" documentation set. So, I suggest that you do some digging around to see what is going on. I'm going to call my local salesman. Maybe we can figure it out.

At any rate, I am sure that it is available somewhere! We expect to see the rest of the "integration" document set (the DSAB manuals) soon.

Two is Better Than 1

Since we all have version 2 of ALL-IN-1 up and running, we are all moving toward having it "in production", right? Well, probably not. Since it is such a big change, most folks that I know are putting up a copy to "play with" so they can learn it's ins and outs before they try a conversion effort. Good idea!

Following my version 1 experience, I want to run two versions of version 2 on the same system. One for "production", and one for me to "play with." Given that you have disk space, I think that this will work just fine. The only things that you need to worry about appear to be the logical name conflicts.

Chapter 4 in the Office Menu Installation Guide discusses the concept of running version 1.X and version 2.X briefly. I think that this discussion will be all that you need to have a version 1 and a version 2 system coexist. Two version 2 systems are a bit more complicated, in that there are a set of system wide logical names assigned by the standard ALL-IN-1 startup. At this writing I am reasonably sure that you can just make sure that a new set of logicals are defined by the process that is running the second version of version 2. I think that this can be done very simply by taking a copy of the version 2 ALL-IN-1 startup procedure and modifying it to point to the disk/directory where the second system "lives".

An additional neat trick might be to put in a new DCL verb that activates the new system for you, given that you have run your modified ALL-IN-1 startup command procedure in the context of the current process. I am going to take a copy of the ALL-IN-1 Command Definition, and change the "ALLIN1" specified verb to be "ALLIN2" to identify the second system.

Those are some rough ideas. I will be exploring them at length in the next few weeks. Since the airfreight driver is here waiting to pick this up, I better stop here and let them take it away! We are all interested in your conversion "hints and kinks"/"war stories"! Please send along your thoughts on paper, or give me a call to chat so we can get the information out so people can see it and make use of it.

See you next month! (Monthly, now? - At last!)

Happy ALL-IN-ling (VAXing)!

FEATURED ARTICLES

ALL-IN-1 V2.0 System Management

From

The New Orleans Symposium session of the same name

Complements of

Dave Shove
Principal Software Engineer
Integrated Office Systems Group
Unified Office Systems Engineering
DIGITAL Equipment Co. Limited
Reading, England

Transcribed by
Ray Kaplan
PIVOTAL, Inc.

THANKS, Dave!

① ALL-IN-1 (V 2.0)

SYSTEM MANAGEMENT

- o Account Maintenance
- o Mail (Electronic Messaging)
- o Master File Maintenance
- o File Cabinet Maintenance
- o Shutdown

② CREATE USER

- o Makes Entry in User Profile
 - Name, Title, Address ...
 - Privileges
 - Preferences
 - VMS Username, Directory ...User Can Change Some Things.
- o Creates VMS Username (Optional)
- o Sets Up VMS Quotas Correctly
- o Creates Disk Directory and Sub-Directories
- o Copies Prototype Files
 - File Cabinet Index
 - Nicknames File
 - Calendar File

③ DELETE USER

- o Removes Entry From User Profile
 - o (Optionally) Removes VMS Username
 - o Deletes User's (Private) Documents, Calendar, Nickname File, ...
 - o (Optionally) Deletes User's Other VMS Files
- Could Use Backup Function First

BACKUP & RESTORE USER'S FILES

- o Makes A Private Copy Of All User's Stored Documents
- o Uses VMS Backup Utility
- o All Document Only, At The Moment. No Selective Restore.
- o Can Be Used To Transfer A User To Another Machine

④ OTHER ACCOUNT MAINTENANCE FUNCTIONS

- o Move A User's Files To A Different Disk Device
- o Copy A User's Profile Entry From Another ALL-IN-1 System.
- o File Cabinet Janitor Function.

OA-25

⑤ MAIL SYSTEM MANAGEMENT

- o Control Of Fetcher And Sender:
 - Start
 - Stop
 - Hold
 - Status
 - Run Now
- o Display Sender Queue
- o Cancel Outgoing Message Of Any User
- o Show Status Of Any Message Of Any User
 - Includes Message Router ID
- o Change Message Router Password
 - Changes AI's Idea Of It.
 - Must Then Use MRMAN To Change MR's Idea Of It.
- o Change Time Zone
 - Name
 - Offset From Greenwich (Reading ?)
- o Purge All Waiting Incoming Mail For A User
 - Can Also Use To Purge All Waiting Outgoing Remote Mail FOR ALL USERS.
 - Only For Emergency Use.
- o Display System Batch & Device Queues.

⑥ MAIL SYSTEM MANAGEMENT (Continued)

- o System Distribution Lists
 - Like V 1 (Held In User Profile)
 - Create, Edit, Delete, Read
 - If A User Has List Of Same Name As System List, Will Get User's List.
 - Can Only Be Manipulated By System Manager.
- o Network Profile
 - NETWORK.DAT
 - Subset Of Data In PROFILE.DAT
 - PROFILE Searched First, Then NETWORK
 - Can Be Same On All Nodes
 - Simple (Entry) Form To Update
 - Or Can Update Via Network Profile Update.
- o Logging Control
 - Switch Logs: Just Renames The Files (Mail Will Create New Ones)

⑦ MASTER FILES

- o User Profile (Changes Go To Network Profile)
- o Network Profile (Users On Other Nodes)
- o Printers
- o Format (Document Handling)
- o Multi-Node (List Of Other Nodes)
- o MASTER (List Of These Files)
- o Functions To
 - Add Entries
 - Edit Entries
 - Delete Entries
 - Read Entries
 - Print (Some Or All) Entries
 - List ("Index") Of Entries
- o Can Add Extra Files (For Customized Applications)

⑧ FILE CABINET AND DATA FILE MAINTENANCE

- o Janitor
 - Empties User's WASTEBASKETS
 - Optionally, Reorganizes Data Files
 - Users Can Also Empty Their Own
 - o Reorganize
 - Runs VMS CONVERT Utility
 - Recovers Space
 - Reduces Index Fragmentation On files:
 - Shared File Cabinet Index
 - Pending Mail File
 - Time Management Files
 - User's File Cabinet Indexes
- Separate Function To Reorganize User Profile File
- o Verify File Cabinet
 - Checks Consistency Of Indexes
 - o All Run As Batch Jobs

OA-26

⑨ NETWORK PROFILE

Two Files:

- o Network Profile
 - All Users On All Nodes
 - Addressing Information
 - Subset of User Profile
- o Node List
 - All Nodes, 1 Entry Per
 - Name Of Node, Name Of "Source" Node, Time Range, Control Data.
- o Allows Various Kinds Of Topography:
 - All Nodes Connected
 - A B
 - C D
 - Or Otherwise
 - A F
 - B D E
 - C G

⑪ SHUTDOWN

- o Gets Time and Duration (With A Form)
- o Types Messages On All ALL-IN-1 User's Terminals
- o Prevents Any New Users Running ALL-IN-1
- o At The Time, Displays List Of Users Still Running ALL-IN-1
- o Asks Whether To Delete Their Processes
- o No Auto-Restart: Duration Is For Info Only

⑫ ROUTINE JOBS

- o File Cabinet Janitor
 - Can Be Done Together
- o File Cabinet Reorganize
- o Check Mail Error Log
- o Check Sender Queue Isn't Too Long
 - Check Sender Is Turned On
- o Network Profile Update (If Used)
- o Login As POSTMASTER And Check INBOX
 - Messages That Failed To Be Sent/Fetchd 10 Times Are Here.

12-BIT WORLD

DECUS 12 BIT NEWS

=====

NUMBER 50

SEPTEMBER 1985

Contributions and correspondence should be sent to:

Robert Hassinger, 12 Bit Coordinator
 c/o DECUS ..or.. Liberty Mutual Research Center
 249 Northboro Rd (BP02) 71 Frankland Road
 Marlboro, MA 01752 Hopkinton, MA 01748

(Please include reference to Newsletter number and page when inquiring about material published.)

NEWSLETTER SUBMISSIONS

Submissions for the 12 Bit News are accepted at all times and are normally used in the next issue to go to press regardless of date of receipt.

Material submitted in machine readable form is particularly desirable because it can be edited and incorporated into the newsletter format more easily. Higher quality reproduction is also possible this way. Contact Bob Hassinger for further details on acceptable media and formats if you plan to make a submission in machine readable form.

WELCOME 12 BIT USERS: PDP-8, PDP-12, DECSTATION AND DECMATE TOO

This issue of the Newsletter is scheduled to be the first published under the new "One Big One" policy - that is all the Newsletters combined into one large, monthly issue rather than separate, less frequent issues for each Special Interest Group. As a result many who have lost track of the 12 Bit Newsletter over the past few years or who are new to the 12 Bit world will be getting this issue and wondering about it so here are a few words about what it is and why.

You will notice this 12 Bit Newsletter is number 50. It started in 1970 when I was asked to start something called the PS/8 Newsletter, which later grew into the PS/8 Special Interest Group (the first DECUS SIG). The name changed to OS/8 when DEC changed the name of PS/8 which by then had become the principle operating system for PDP-8 based systems. As time went on the Newsletter and SIG evolved to cover all 12 bit related topics, using the 12 Bit SIG name. It now covers all PDP-8, PDP-12, DECstation, and DECMate systems. Hardware as well as all the software systems including the OS/8 family, WPS, COS and the multiuser systems are included. Once in a long while we even get questions about the LINC-8 and we still do our best to help. If you use a 12 bit systems, this Newsletter is where you will find a chance to exchange information, ideas and help.

A few years ago DECUS decided the level of activity of the 12 Bit SIG had tapered off to the point where it should be converted into a subordinate activity of the Office Automation SIG because that was the only product area where DEC was still actively marketing new 12 bit based products (DECMate WPS systems) and

⑩ NETWORK PROFILE FUNCTIONS

- o Entries On User Profile Are Automatically Propogated To Local Network Profile
- o Network Update:
 - Batch Job
 - Accesses All Nodes Listed On Node List
 - Reads Their Network Profile, Adds All Changes Since Last Run To Local Network Profile.
- o Can Also Make Entries On Network Profile By Hand
 - Users With No ALL-IN-1 Account
 - Users On Nodes You Don't Get Updates From
- o Network Purge:
 - Batch Job
 - Removes "Deleted" Entries
- o Network Initialize:
 - Use After Conversion From V 1

OA-27

OA-28

so that was where DECUS thought DEC support would come from. It has not worked out quite that way but here we are any way.

The new combined newsletter format has been a highly controversial issue within the DECUS leadership this Spring, with many bad feelings and problems caused by the way it was pushed on the rank and file workers with poor leadership. I guess after all that has happened in the past I can give this latest brainstorm a chance even though it would be easy to say that fifty issues and fifteen years is enough for anyone, but, if you want the 12 Bit News to continue you need to let me know and you need to pass along your thoughts, ideas, problems and solutions. A Newsletter without news is not very interesting and it is certainly hard to edit. -- Bob Hassinger

12 BIT COMMITTEE

Robert Hassinger
- address above -
(617) 435-9061

COS/DIBOL and WPS liason
Lawrence H. Eisenberg
17141 Nance Street
Encino, California 91316
(818) 788-0354

Education, Multiuser systems, PASCAL
Father Geoffrey Chase O.S.B.
Portsmouth Abbey School
Portsmouth, RI 02871
(401) 683-2000

Representitive to DECUS Product Planning Committee
Jim van Zee
Lab Data Systems
10320 Ravenna Ave NE
Seattle, Washington 98125
(206) 522-6950

12 Bit Software Project
Wally Kalinowski
The Aerospace Corporation
PO Box 92957
Los Angeles, CA 90009
(213) 648-6940

12 BIT SOFTWARE PROJECT UPDATE

Wally Kalinowski has been working for quite some time on a project to raise money to acquire and place in the public domain 12 bit software that is particularly valuable and worth insuring continued availability for. Recently I received material from Wally that indicates he is making progress on a number of items, particularly the software from Dewar Information Systems including PAGE8, ACID, ICE, HIBOL and VISTA.

We have had very little chance to talk lately but I get the impression the main need now is for additional contributions to purchase the rights to these very useful items. If you are interested, Wally's address and phone number are listed at the beginning of this Newsletter.

Wally also sent information on the progress on CCLX, a new and much improved version of CCL. Wally says it is almost ready for submission to DECUS. I had hoped to be able to review this and report on it in this issue but no time.

OS/278 FOR PRE-DECMATE II SYSTEMS - UPDATE

Ever since OS/278 became available for the DECMate II through the DECUS Program Library, we have been trying to make it available for pre DECMate-II systems. Regular readers of this Newsletter know of the many problems with this. Recently Rudi Stange sent me an 8 inch floppy set with OS/278 on it. There is still a problem however. Because of space problems, Rudi made one of the disks an RX02 double density floppy. He can not test it on his DECMate II because you can only boot from RX50s on it and I can not test it because I only have single density floppys. I think the best thing would be to rebuild the kit as an all RX01 package that all 8 inch floppy systems could work with. The only resource I currently have for that would be through the DECUS office and that option has been tied up in red tape for many months.

Don't lose hope, I am still working on it. I think it will be worth the wait.

UPDATE ON LIBRARY SUBMISSIONS FROM JIM VAN ZEE

In the last Newsletter issue there was an article about new and updated material Jim van Zee had recently submitted to the DECUS Program Library. I recently received a package from Jim containing new media and a note about problems he had with his submission of DIRECT V7C. The new submission is on it's way to the Library and hopefully will be available soon. Others may find what Jim ran into interesting or helpful.

Jim's original submission had included copies on DECTape (real DECTape, not TU58), LINCtape, paper-tape and RX01 (single density) floppy disk along with a full listing done on a good letter quality printer. DECUS returned everything except the floppy disk saying that DECUS no longer supplied DEC-, LINC-, or paper-tape! Well, this is almost true. The Library stopped accepting paper tape submissions some time ago and I think we have now passed the cutoff date for ordering existing material on paper tape. They stopped supporting LINC tape quite some time back because of lack of interest. I am not sure I knew DECTape was no longer supported at the time of Jim's submission but I think it too has been dropped by now for lack of orders and the cost of continuing to maintain the hardware to reproduce it.

In fact, this Spring the DECUS Library decided to get rid of it's PDP-8 system completely. In the future they plan to do all reproduction on a VAX. It should be possible to handle floppy disks this way but not much else from the 12 bit world. They will retain the ability to reproduce RX50 (5 1/4 inch) floppies too of course.

Needless to say these changes in the Library have come slowly and over much objection from people like me but when you are a small minority compared to the vast numbers in the VAX, PDP-11, PC and 36 bit worlds there is just so much you can do. Economics rules in the end and you find yourself outvoted many times over by people who may never even have heard of a 12 bit system and certainly have no feeling for the systems or the needs of their user base. Self interest and politics are very strong, even among the user leadership of DECUS. I think we can take comfort however in the fact that many of those other systems are starting to get old too and before long a lot more people are going to start understanding our problems much better.

The DECUS Library later returned Jim's floppy too because they had trouble getting a proper directory listing from it (that issue will go away when they move PDP-8 reproduction to the VAX where they will not be able to do directories). Jim checked into the problem and found his disk was fine. Eventually he concluded DECUS was using the "kludged up" version of DIRECT mentioned last time. If they had been using the standard OS/8 version of DIRECT or Jim's V7B from the Library or the V7C version off the new submission they would not have had trouble.

The new submission also contains an improved version of HELP. It will handle abbreviated topic names, that is HELP DIR rather than HELP DIRECT. This is nice because now the help can match with the abbreviations CCL allows when you issue the commands. The new version is three blocks shorter than the standard DEC version. Another very handy feature is that it looks for the help file on the device called HELP: first, then it looks on DSKO: (good for MULTI-8 users) then finally on SYS:. This is very nice for saving scarce system disk space, makes it less painful to have extensive help files and it allows swapping help files with a simple ASsign command.

HELP - OS/8 V3D FOLLOWUP

In the last Newsletter issue, Dr. Anthony L. Marchese wrote looking for help on how to get an upgrade from OS/8 V3C to OS/8 V3D which is the current (and last) release of OS/8. This is the release that fixed the dates so they would continue to work after 1977. It started shipping early in 1978.

Dr. Marchese had been having a lot of trouble trying to find a way to get a V3D update for the V3C system that was with the computer that had recently been transferred to him. He could not find anyone at DEC who could help him even though DEC still considers this a product and so public domain distribution outside DEC's marketing channels is still prohibited.

Since anyone who wants to use an OS/8 system needs OS/8 V3D to fix the date problem that had been built into all previous releases, this is still a real problem for people who want to upgrade a system to an OS/8 configuration or who, like Dr. Marchese, come into possession of a system that had not been kept up to date in the past.

In response to this problem I got a call this Spring from Bob Kaplow at DEC. Bob is an "old PDP-8 person" and he offered help on this sort of thing. He says the OS/8 V3D Combined Kit Update order number is QJ-024-H* (*=media code) and the price is \$740. I have since checked the listings on DEC's "Electronic Store" and found this kit listed under QF-024-H* (that is QF rather

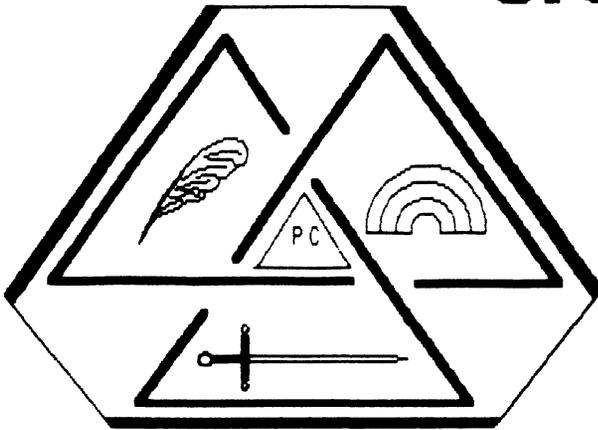
than QJ) so check it both ways if you are looking for it. Note too, this is an update kit for someone who has a license for OS/8 already. The full price to buy the kit new is about four times as much. Note though that DEC may have a lot of trouble trying to figure out at this late date if you had a V3C or earlier licenses. Their records do not seem to be very good that far back.

The combined kit brings together the various parts of the system that had originally been sold as separate pieces - BASIC, BATCH, FORTRAN IV and so on. It is what almost everyone would want if they were ordering OS/8 today. I did find I had a problem with the kit when I got it because mine came on DECTape and there were only TD8E type boot blocks, none for my TC08 as there had been in past releases. That is a very obscure case at this point though and I was able to get around it so others should be able to also.

Bob's phone number is 312-394-6141.

DECUS

**PERSONAL COMPUTER
SIG NEWSLETTER**



PERSONAL COMPUTER SIG STEERING COMMITTEE

Chairman

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

Symposia Coordinator

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

DECmate Working Group Chairman

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

Pro Working Group Chairman

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

Rainbow Working Group Chairman

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

Newsletter Editor

Caroline M. Mack
6415 Adelphi Road
University Park, MD 20782
(301) 927-0108

National LUG Organization Rep

Pierre Hahn (acting)
SUNY HSC-T10-02808101
Stony Brook, NY 11794
(516) 444-1362

Communications Committee Rep

Mark DeMoss
Guinn, Smith and Company
2408 Texas Drive, Ste. 100
Irving, TX 75062
(214) 255-7120

Library Comm. Rep/PC Librarian

Kurt Reisler
Hadron, Inc.
9990 Lee Highway
Fairfax, VA 11030
(703) 359-6100

PC/RSX Liason

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

PC/Graphics Liason

Dr. Khin Maung Yin
Kent State University
1411 Clarindon-Troy Road
Burton, OH 44201
(216) 951-1447

Pre-Symposia Session Coordinator

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

Session Notes Editor

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

Members-at-Large

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

Jim Christine
SPSS, Inc.
1815 S. Cuyler
Berwyn, IL 60402
(312) 329-3580

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

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

Digital Counterparts

DECmate:

Ron Gemma
Digital Equipment Corp.

PRO:

John E. Bennett
Digital Equipment Corp.
ML01-2/C30
146 Main Street
Maynard, MA 01754-2571

Rainbow:

Katrina Holman
Digital Equipment Corp.
LJ02/13
30 Porter Road
Littleton, MA 01460
(617) 486-2451

About this newsletter:

The graphic on the front is a proposed logo for the PC SIG, designed by Tom Hintz. The sword represents the PRO, the quill represents the DECmate, and the Rainbow, of course, represents the Rainbow.

The title page was composed using FNCYFNT, an MS-DOS Public Domain font program by J. Anthony Moushon, using Gothic Bold and Double fonts on an LA-50 printer. FNCYFNT is available on several FIDO net bulletin boards and from some user group libraries.

The Newsletter itself was composed using WordPerfect Version 4.0 with an LA50 printer.

TABLE OF CONTENTS

Personal Computer SIG Steering Committee.....	PC-i
From the Editor.....	PC-2
How To Submit Articles to the PC SIG Newsletter.....	PC-4
Greetings from the Chairperson.....	PC-5
Rainbow Corner.....	PC-6
Sysop's Corner.....	PC-7
Guest Editorial.....	PC-9
RAINBOW SECTION	
Rainbow Software Information and Reviews	
Graphwriter.....	PC-10
Overhead Express.....	PC-11
WordMarc.....	PC-12
Writer's Pack.....	PC-16
Random House Proofreader.....	PC-16
Footnote.....	PC-18
Bibliography.....	PC-19
Grammatik.....	PC-20
Index.....	PC-21
Investment Manager.....	PC-22
Do-It.....	PC-24
Whatever Happened to Concurrent CP/M?.....	PC-27
Comments: GraphTalk.....	PC-30
Comments: Tektronics and Speechmaster.....	PC-30
Hack.....	PC-30
The Fine Points of MS-DOS and CP/M	
Using MS-DOS Batch File Commands.....	PC-32
Screen Controls from the CP/M Operating System.....	PC-37
DOS Formatting a Blank Diskette.....	PC-38
Technical Articles	
Adding Memory: Easier and More Inexpensive Than Ever.....	PC-39
Reinforced Hubs Cause Problems with Disk Drives.....	PC-40
Setting Printer Features.....	PC-41
Lotus 1-2-3 and Printgraph on the Rainbow.....	PC-43
Communications Tip.....	PC-44
Using a FIDO Bulletin Board.....	PC-45
Rainbow Questions and Answers.....	PC-50
Rainbow Wish List.....	PC-52
DECMATE SECTION	
Rainbow to DECmate and Back Using PolyCom.....	PC-53
DECmate II WPS Version 2.0 Fix.....	PC-54
DECmate Notes From New Orleans.....	PC-54
DECmate Questions and Answers.....	PC-55
DECmate Wish List.....	PC-57
PRO SECTION	
PDL: An Online Public Domain Library.....	PC-60
Request For PRO Series Enhancements (Wish List).....	PC-61
ROBIN (VT180) SECTION.....	PC-63
GENERAL INFORMATION	
DEC Specific Computer Magazines and Publications.....	PC-64
What Would You Tell DEC If You Could?.....	PC-66
Ms. Motherboard.....	PC-69
The Grapevine.....	PC-71
Short Notes.....	PC-72
Software and Hardware.....	PC-73
Books.....	PC-75
Letters to the Editor.....	PC-76
USER GROUP COLUMN.....	PC-78
RAINBOW ORIENTED BULLETIN BOARDS	
Setting Printer Features: Printer Samples.....	PC-83
LA 100 Printer Setting Cheat Sheet.....	PC-85
DECUS Commercialism Policy for Publications.....	PC-88
DECUS Personal Computer SIG Questionnaire.....	PC-89
.....in special section at end of Vol.	
PRO 300 Series Wish List Ballot	
.....in special section at end of Vol.	

FROM THE EDITOR

Several comments this time. . . This is the first issue of what has come to be known as "The Big One." The Big One, which combines all SIG Newsletters, is an attempt by DECUS to cut publication costs and to be certain that subscribers receive something for the money they have paid (since not all SIGs publish newsletters when promised, and some never publish at all). The cost of the single combined newsletter will allow those who previously subscribed to numerous newsletters to save money, but will cost others, who previously subscribed only to one or two newsletters, more money than before. (It is \$35/year.) For many DEC PC owners, particularly those who do not use their Rainbow, PRO or DECMate as an adjunct to a VAX or other minicomputer, the new combined version will have far more information than necessary--on totally unrelated subjects. This editor was against the idea of , but, like the other SIG newsletter editors, had absolutely no choice in the matter, which was decided by the DECUS Management Council.

The PC SIG Newsletter will probably continue to be a quarterly publication, although the "Big One" will appear monthly. Those who subscribed to it last year will notice that some of the articles in this newsletter appeared in the last newsletter. This issue of the newsletter is slated to go to all 42,000 (or so) DECUS members, not just the 2000+ people who already subscribe to the PC SIG Newsletter. So a few of the articles have been repeated in this issue.

There is a something which appears to me to be a much bigger problem for DECUS PC SIG members. That is the overly restrictive DECUS commercialism policy (the portion governing publications is reproduced below). While it is laudable to keep DECUS members from being harassed by overzealous salesman (whether from DEC or third party manufacturers), I feel that the policy is far more harmful than helpful. As many of you are aware, I was forced to "white out" all prices in the last PC SIG Newsletter. This meant that, for instance, the article which explained how to cheaply upgrade memory in the Rainbow, which featured a comparison of the prices on DEC and non-DEC RAM chips (DEC's are more than fourteen times as expensive) was ludicrous. The software reviews (written by people who had used the software) were characterized by one zealous DECUS employee as being "practically commercials."

The situation is particularly bad for PC owners because there is no other national forum which focuses on DEC PCs. The non-specific computer magazines almost never mention DEC. (Why should they, when DEC commands less than 4% of the total PC sales?) In the three DEC-specific commercial magazines, the situation is almost as bad! Digital Review, which originally devoted most of its space to the Rainbow and Pro, has dropped PC coverage almost entirely. Personal and Professional has gone from a 6 times a year DEC PC magazine to 24 monthly pages in a special issue of The DEC Professional. Hardcopy continues to have about one article a month, and has added a single page Rainbow column.

While it is not necessarily reasonable for the DECUS PC SIG Newsletter to attempt to take the place of non-existent commercial magazine coverage, there really isn't much else available to the thousands and thousands of DEC PC users who have been left high and dry by DEC's disastrous and capricious marketing policies.

DEC PC users must have an opportunity to compare software and hardware, both DEC's and that of third party manufacturers, INCLUDING PRICE INFORMATION. To continue to "white out" all price information is ridiculous. I urge you to write to the DECUS Board of Directors [send mail to them at DECUS, 249 Northboro Road (BPO2), Marlboro, MA 01752] to ask them to repeal or at least rewrite the commercialism policy so that the PC SIG Newsletter and other PC related publications (and preferably other newsletters) can present a balanced picture to its readers. If you write, please cc me.

More Bad News: DEC continues to pull away from the PC market. DEC Business Centers (at least in the Washington D. C. and Baltimore areas,) will no longer carry Rainbows, Rainbow software, and supplies. For instance, in order to get DEC LA-50 ribbons, I have two choices: either DECdirect or the DEC Electronic Store. Forget a quick trip to pick up a few ribbons. Of course, as many people are aware, it is possible to get C. Itoh Prowriter ribbons, since the LA-50 is really a Prowriter underneath its DEC label, but the ribbon quality is usually not as good.

What a thrill it would be to be able to walk into a computer store, browse around, check out the latest stock of Rainbow software, find a few new books dealing with aspects of the Rainbow, and pick up the latest copy of a magazine or two. Or to find some software suitable for children. . .

For those who are nervous about submitting articles because of the DECUS/DEC copyright on the newsletter, Paula Morin, of DECUS, clarified the issue for Barbara Maaskant. Paula states that the individual author "owns" the individual copyright on his or her individual article. DECUS/DEC own the copyright on the final document: so it is not okay to copy the whole newsletter. If the person is the author of the article, he or she does not lose or relinquish rights to the article by having it published in the PC SIG Newsletter. (That's a relief.) If you are concerned, feel free to copyright articles. If you do, and you submit them to the newsletter, you are implicitly giving permission to publish them, also for the editor to make editorial changes (don't worry, I don't have time to make too many).

Because a number of the articles in this newsletter were originally published in the Washington Area Rainbow User's Group Newsletter, there are a lot of references to the WARUG Public Domain Library, which currently has over 65 volumes of public domain software. Most of the software is not in the DECUS library yet. Much of this software is available on FIDO bulletin boards and/or other DEC PC LUGs. If neither of these sources is available to you, contact Jack Ference, the WARUG Public Domain Librarian. (Please send a stamped, self addressed envelope for information on how to use the library and a list of the volumes and their contents. His address is: Jack Ference, 5511 Suffield Ct., Columbia, MD 21044.)

The PC SIG now has a growing and active leadership. We are here to serve you. Let us know who you are, what you want the PC SIG to do, how the SIG can help you, and what you want to see in the newsletter. Be sure to fill out the questionnaire at the back of this issue (it will be in a special section). The SIG does exist between DECUS Symposia. Since the SIG is national, it doesn't function as simply as a Local User Group, where it is easy to call up someone and shoot the breeze (without incurring long distance bills). Networking is a popular concept--both in computers and jobs. It can work for PC SIG members too. If you are an expert--or at least competent--in something related to the Rainbow, PRO, or DECMate (software application or hardware), and would be willing to help others, please say so in the questionnaire. We will develop a contact list if there are enough volunteers.

Consider writing an article for the newsletter. It is dependent upon your contributions! Reviews of commercial and public domain software, technical articles, how-tos, summaries of other meetings you've attended, rumors, complaints, questions, answers, new products, citations of reviews on DEC hardware and software, comic experiences--and whatever else you can think of, are needed. Details on how to submit articles are below.

Despite the fact that many DEC PC owners feel abandoned, there is a large number of Rainbow, PRO, and DECMate users. Get involved with the PC SIG and local user groups. You have nothing to lose, and much to gain!

Caroline Mack
July, 1985

=====

HOW TO SUBMIT ARTICLES TO THE PC SIG NEWSLETTER:

If you write an article for the newsletter, please send me a hard copy by mail (my address is above). Also, if the article is more than 2 pages long, or has anything which is difficult to type, send me your articles in one of the following formats (let me know which it is!):

- o WordPerfect on an MS-DOS formatted floppy disk
- o WordStar on an MS-DOS or CP/M formatted floppy disk
- o Samna on an MS-DOS formatted floppy disk
- o ASCII on an MS-DOS formatted floppy disk--remove all formatting commands first, please, or I'll have to strip them out one by one
- o Hard copy (typed or printed out)
- o Via modem using DECMINI (call first)
- o Via DCS (DECUS)
- o Upload to the WASH-A-RUG FIDO Bulletin Board (703) 359-6549

I expect the next PC SIG newsletter to be published in December unless something unusual happens. The deadline for submissions to that issue will be October 10.

=====

GREETINGS FROM THE CHAIRMAN

=====

As most of you know, the PC Special Interest Group has undergone a lot of change in the past six months. In December, 1984 at the DECUS Symposium in Anaheim, new leadership was elected. I became Chairman of the PC SIG, Rick Eliopoulos was elected Symposia Coordinator, Lynn Jarrett was elected Rainbow Working Group Chairman, and Celeste Markovitch agreed to become Newsletter Editor. Each of us was a novice not only in our PC SIG position, but to DECUS Leadership. Thanks to the three "veterans" who remained, Tom Hintz, Pro Working Group Chairman, Cheryl Johnson, DECmate Working Group Chairman, and Mark DeMoss, who had previously edited the Newsletter, and who became the new Communications Committee Representative, we all left Anaheim with a shot in the arm of enthusiasm and plans to work hard getting the SIG on its feet for the DECUS Spring Symposium in New Orleans.

We met in late February, where plans for New Orleans and building a stronger PC SIG began to take shape. We had just gotten a newsletter issue out when Celeste had to resign her position as editor. Fortunately we succeeded in recruiting Caroline Mack as PC SIG Newsletter Editor in the midst not only of getting the next issue out, but of a DECUS wide fracas over going to a new format which would include all SIG Newsletters in one combined monthly issue. Caroline's reputation as the editor of the Washington Area Rainbow User's Group Newsletter preceded her.

The SIG's primary goal for New Orleans was gaining exposure. A lot of hard work, and, I don't mind admitting, apprehension, preceded the New Orleans Symposium. Both the hard work and apprehension paid off. The SIG sponsored a very successful open house in the SIG suite on Monday evening, and a Software Clinic on Wednesday evening. Special thanks go to our Digital Counterparts, Katrina Holman, John Bennett, and the outgoing Steve Coughlin for coordinating volunteer time from already overworked Digital developers who spent time during the clinic answering questions and solving problems. I am not exaggerating when I say that on both nights, there was standing room only in the suite. In addition, a software raffle, with software donated by each of the counterparts and Jim Butler, of the Peripherals and Supplies group, was held on Wednesday. Six software packages were raffled (instead of tickets, participants offered their business cards). One of the most popular activities was public domain software distribution. DEC made a fully loaded Rainbow 100+ available to the SIG to allow people to view new and recently announced Rainbow products.

When not used for demos, the 100+ was hard at work making copies of the seven Public Domain Software disks which were available. The most popular items were the Rainbow graphics subroutines provided by Marshall Goldberg of Digital, and Tom Jenning's FIDO Bulletin Board System. At least 450 diskettes were copied before the wrap up session on Friday, when to everyone's disappointment, the suite had to be closed. Many thanks to those who volunteered time on this project and particularly to Rick Eliopoulos, whose sustained energy and enthusiasm kept us going.

As you can see from the greatly expanded list of Steering Committee personnel, the SIG successfully filled several positions that it advertised during the New Orleans Symposium. I am happy to welcome the new members.

New Orleans is barely past, and the steering committee is already working hard planning and preparing for the all Symposium in Anaheim. There are several things to look forward to in the Fall. The software exchange will be offered again, and we hope to have diskettes readily available (though you may want to bring some blank diskettes with you). At least one suite party is planned; currently we are leaning toward a "PC PUB" theme. The clinic will be repeated and the SIG is hoping to have a campground for better accessibility and placement of two Rainbows, a DECmate, a PRO, and perhaps even a Robin. Two Pre-Symposia Sessions are planned; one on FIDO Bulletin Board Systems, and one on Graphics. Look for more information on them in the early registration form.

At least one store item will be sponsored by the SIG, and Session Notes will be available. Many more user presented sessions will be offered, including a PC Magic Session and several interesting panels. So save up your favorite questions, wish list items, user of the week, and war stories to share with us in Anaheim. There will be a PC Campground, too! I look forward to seeing you there!

Barbara A. Maaskant
Chairman

=====

RAINBOW CORNER

=====

Katrina Holman
DEC Counterpart

What's new with the Rainbow personal computer? Recent PC Applications available from Digital include Symphony, DESQ, dBASE III, 20/20, AutoCAD 2, Overhead Express, GraFIX Partner, and Do-It. The Rainbow 190 Office System and a Rainbow 100+ to Rainbow 190 Upgrade kit, which includes WPS-PLUS/Rainbow and Rainbow Office Workstation software, are now shipping. DECNET/Rainbow was also discussed at sessions in New Orleans.

A new Rainbow Product Guide (EZ-W4968-85) was created especially for the Spring 85 DECUS and distributed there. It contains descriptions and ordering information for each of the Rainbow software packages and hardware options sold by Digital as of May, 1985.

Katrina Holman, the PC SIG DEC Rainbow Counterpart, is with the Personal Computing Systems Group at DEC.

=====

SYSTEM OPERATOR'S CORNER

=====

by Kurt Reisler

Running on a DEC-donated Rainbow 100B, in facilities provided by Hadron, Inc., the Wash-A-RUG FIDO has been continual operation for the last 4 months. The board runs under the auspices of the Washington Area Rainbow Users Group, and has become a primary clearing-house for technical information about the Rainbow PC, as well as a source of public-domain software. Although heavily used by the local Rainbow population, the board has been used by individuals as far away as California. In addition, users on the board have received FIDOMAIL from other FIDO systems as far away as Hawaii. In its four months of operation, the Wash-A-RUG FIDO has accumulated over 300 regular users, and averages about 200 callers per week. The system has over 5 Megabytes of public domain software available for downloading. This amount keeps increasing (and available disk space keeps decreasing) as users and the SYSOP keep uploading new software to the system.

What are FIDO and FIDONET? FIDO and FIDONET are complementary parts of a public-domain bulletin board system which was developed by Tom Jennings in San Francisco, CA.

What began as a hobbyist's experiment has quickly grown to be a major revolution in communications. Starting with 2 boards in June of 1984, FIDO has spread to over 400 boards. These boards are running on PCs in the US, Canada, England, Sweden and Indonesia. For the most part, these boards are run by individuals on their own PCs. Although FIDO runs on a variety of PCs which support MS-DOS and PC-DOS (including the DEC Rainbow, IBM PC, PC jr, Otrona, Tandy 2000, Victor 9000, and Sanyo) the user interface is standard across FIDOs. Additionally, the FIDO BBS offer a variety of protocols for file transfer. Users can select either KERMIT, XMODEM, MODEM7, TELELINK or raw ASCII protocols.

FIDONET is a scheduled portion of the FIDO BBS which makes it unique among the public bulletin boards. FIDONET allows FIDO systems to provide Electronic Mail (E-mail) services for its users. Using a dial-up network, with regions and hosts, a user can route messages and/or files between distant systems quickly and at a relatively low cost. Routing is done automatically by FIDO, and FIDONET is run during scheduled "windows" on all FIDO systems participating in FIDONET. FIDONET allows FIDO SYSOPs to stay in touch and stay involved in the evolution of the system. In addition, it is an efficient way of distributing the weekly FIDO Newsletter, and weekly updates to the nodelist. Most FIDO systems require that users establish a "credit" balance (usually \$5) to make use of FIDONET to non-local FIDO nodes. The "per-message" cost (usually \$.025 each) is deducted from this balance. The software to establish a FIDO system can be obtained from a variety of sources. You can purchase the software from the author (Tom Jennings) for one hundred dollars. This price was established to discourage people from buying it directly from Tom. You can usually download the required software from the local FIDO systems in your area.

I keep both the IBM and the DEC Rainbow versions available on the Wash-A-RUG FIDO. The other versions can be downloaded from some of the larger systems which have the disk space to spare. Finally, you can obtain the software from the DECUS Library (DECUS Program RB-102 FIDO version 10g) for a cost of twenty-five dollars (\$25). I hope to get the most recent version (V10m) into the library this week.

So, what's new in public domain software on the board? Here is a sample:

ARC.LBR A new file compression utility has become popular.

ARC.EXE combines the functions of squeezing and unsqueezing files, as well as archiving them. It selects one of four compression methods and creates archive files with a .ARC extension. The

library contains the executable and documentation.

MAINT.ARC There is a MS-DOS version of the CP/M "MAINT" utility. The archive contains the executable, documentation and source.

DTC.ARC This archive contains a desk-top calendar program. The archive contains the executable, documentation and an initial data file.

QIX.COM and MONSTA.EXE Both of these are games that use the Rainbow screen graphics. They are both from the Gnomes at DEC.

SCRAM.ARC This is a public domain version of the DEC graphics game. It requires the use of the Rainbow graphics board, and looks best with a color monitor.

JETSET.ARC This is an excellent simulation game that puts you at the controls of a 747. The archive contains executable, documentation and diagrams.

HACK.ARC This is an addictive D&D type game that can be run on any MS-DOS micro, as long as it has ANSI support (a natural for the Rainbow). It does require 256 Kb free memory to play. It can take upwards of 41 minutes to download at 1200 baud.

EMPIRE.ARC This is another generic MS-DOS game. It is slow to initialize, but seems to be interesting to play. Archive contains all the needed files and documentation.

DECMINI.EQE (version 3.2 18 May 1985) This is a fixed version of Tom Jennings's communication program. Earlier bugs have been fixed, and it also will support 2400 baud modems without problems. Versions exist for other hardware configurations as well.

MSKERMIT.EQE (version 2.27) This is an enhanced version of the MS-DOS version of the popular KERMIT program.

This is just a small sample of what is available on FIDO bulletin boards around the country (and the world). [Editor's Note: See the list of DEC related bulletin boards at the end for telephone numbers of these boards.] The amount of public domain software increases daily, as users upload to the boards, and as SYSOPs circulate things through FIDONET. I would strongly recommend that you investigate local FIDO systems as a method of obtaining public domain software, and for keeping in touch with other Rainbow users.

Kurt Reisler is a Software Engineer with Hadron, Inc. In addition, he is the SYSOP for The Bear's Den (FIDO 109/74) and the Wash-A-RUG (FIDO 109/483). He is also the network coordinator for the FIDO systems involved in the Washington DC area FIDONET (net 109 Wash DC Metro). Kurt is also very active in DECUS, as the PC-SIG Software Librarian, and Session Notes Editor for the UNISIG. At the DECUS Symposia, he can usually be found wearing a camouflage jacket covered with buttons.

(c) 1985 by Kurt Reisler & The Bear's Den

=====

GUEST EDITORIAL

=====

HERE WE GO AGAIN. .OR, WE ALL KNOW THAT THE RAINBOW WON'T RUN
LOTUS 1-2-3
by Will Roberts

How many times have you heard or read that LOTUS 1-2-3 is available only on the IBM-PC or that if it is available on the DEC Rainbow that the DEC machine is slow and awkward? I can't begin to count the number of times I've seen this canard in print or heard it from personal computer "experts" in my own office and the offices of friends and business associates.

Well, I for one have been slow in adopting LOTUS on my Rainbow because it's a poorly implemented adaptation of the original LOTUS for the IBM PC. For example, the Rainbow can display 132 column text making it easy to see twelve months of figures on the screen at once. LOTUS doesn't use this feature because it's not available on the IBM PC. And LOTUS color graphics don't use the full 16 color palette of the Rainbow because the IBM PC doesn't have that wide a choice of colors in its basic spectrum. And of course, LOTUS on the Rainbow is rumored to be slow although the Rainbow uses the same processor and clock speed as the IBM PC. The truth here is that the Rainbow version fails to use the fast screen function because LOTUS didn't take the time to adapt its product to the hardware with the same degree of intimacy it did on the IBM PC.

According to the July 8, 1985 issue of InfoWorld, LOTUS will soon be releasing a version 2.0 (due out this fall) which will include string arithmetic and a number of other fixes and enhancements. As a DEC user, I certainly hope that DEC has learned its lesson and will deal with the folks at LOTUS, even if it means a costly subsidy, to get a decent implementation of 2.0 on the Rainbow.

I urge Rainbow 1-2-3 user's to contact DEC's headquarters office in Maynard, Massachusetts, and urge the company to make sure that it does not completely relinquish the desktop to Big Blue.

Will Roberts is a member of the San Francisco Bay Area DEC PC User Group. This editorial was excerpted from an editorial in the San Francisco Bay Area DEC PC User Group News.

=====

RAINBOW SOFTWARE INFORMATION AND REVIEWS

PRODUCT REVIEW: GRAPHWRITER by Dennis K. Fitzgerald

Graphwriter is a presentation graphics package from Graphic Communications, Inc. The Rainbow version runs under MS-DOS V2.05 or V2.11 and is distributed by Digital (Part no. QAX07-C3) at a list price of \$. . . It is well customized to the Rainbow and produces impressive results. It is not, however, a toy. In conjunction with the high resolution output devices that Graphwriter supports, it is much like having a resident graphics artist. For a user who needs to produce a lot of high quality graphics and can afford a pen plotter or Polaroid Palette, Graphwriter can be great. For a small user with nothing but an LA50 or LA100, it is just too much, and an integrated package like Lotus 1-2-3 or Symphony would probably be more suitable.

Graphwriter has two parts, a Basic Set and an Extension Set. The Digital Distributed Software (DDS) version includes both sets. The Basic Set supports a set of basic graph types: six types of bar charts, pie charts, line charts, scatter plot charts, a combination bar and line chart, and a pure text chart. As if those were not enough, the Extension Set adds a combination pie and bar chart, Gantt charts, organization charts, bubble charts, table charts, surface/line charts, line/table charts, and five more types of bar chart. They come on a whopping nine Rainbow disks. Fortunately, they are organized so that you never need more than two at a time. Graphwriter "composes" your charts using defaults that generally ensure high quality results. However, if the default style doesn't suit you, you can vary almost any style attribute including color, type font, type size, orientation, plot size, spacing, and line and bar style (solid, dotted, cross-hatched, etc.).

Graphwriter comes with an extensive manual including introductory, tutorial, and reference sections. It is completely oriented towards the Rainbow and includes precise instructions on resetting the computer, copying disks, and adjusting Set-Up attributes. Examples are given of each chart format, each style attribute, and each menu. In addition to the manual, the package includes a brief Format Selection Guide that lists all of the supported formats with an example of each one and provides brief instructions on how to make effective graphics using Graphwriter.

Graphwriter is completely menu-driven. However, an expert user can anticipate and type responses before the menu appears. In this case, the menus are not displayed. Data can be entered directly from the keyboard, can be saved and recalled from one session to another, or can be imported from SYLK (Multiplan) or DIF (VISICALC) files. Several chart formats are compatible with other formats (e.g. a stacked bar chart with a clustered bar chart). In these cases, you can change formats without reentering the data.

Devices supported include the DEC LA50, LA100 (and LA210) printers, the Rainbow Color/Graphics option (for previewing plots), several pen plotters (HP 7470A, HP 7475A, HP 7550A, and Digital LVP-16), and the Polaroid Palette. An appendix to the manual provides precise instructions for attaching and setting up each of these devices including cables, device switch settings, and Rainbow SET-UP settings.

My personal experience with Graphwriter is somewhat limited, but what I have tried has been very impressive. Some random thoughts include:

o Graphwriter is heavily overlaid. As a result, it runs rather slowly from diskettes. It can be installed on a hard disk, and runs much faster, but uses up about 2.5 Megabytes of disk space!

o I had a lot of hopes for using the Gantt and Bubble charts in conjunction with my work as a software engineer. While the resulting charts can be quite nice, they need so much specification that it is easier for me to do by hand.

o I am quite impressed with Graphwriter's use of the Rainbow Color/Graphics option for previewing displays. (I don't have a color monitor, but the black and white display is quite adequate for preview purposes.)

o The manual is excellent. My compliments to the writers and editors at Graphics Communications, Inc. If all manuals were this good, life would be much easier for all of us.

o Importing files from Multiplan is a snap. You just save the spreadsheet using the SYMBOLIC option, and then tell Graphwriter which rows and columns to use for labels and data.

o The ability to try different formats using the same data is very useful. As the manual points out, different formats highlight different aspects of the data and can be chosen to send the message that YOU want to get across.

Summary: It's hard to justify the price of this package in my own (non-business) situation, but I really like this package. (Now, if I could just get a pen plotter....)

Dennis Fitzgerald is a Senior Software Engineer with Computer Science Corporation. This article appeared previously in the WARUG Newsletter.

(c) 1985 by Dennis Fitzgerald

PRODUCT REVIEW: OVERHEAD EXPRESS by N. Jay Bassin

It seems that whenever I am giving a presentation to an important group, I am preceded and followed by people who seem to have their own graphic arts departments. Their professionally typeset, multi-font visuals tend to be compared to my felt-pen creations (which always look crooked), with disturbing results to my already-dimey artistic reputation. If you were also held back for remedial crayola, but do need professionally prepared visuals (for overhead transparencies, slides or even report covers), have I got a tip for you! The folks at Business & Professional Software, Inc. (143 Binney Street, Cambridge, MA 02142; 617/491-3377) have recently introduced their program Overhead Express for the Rainbow and DECmate II [\$; MS-DOS; dot-matrix printer required, graphics card recommended. DEC itself markets Overhead Express as "Digital Distributed Software" (DDS).]

I have used Overhead Express for a month now, and I recommend it enthusiastically. The program is a cross between a word processor and a relational database program. You enter your information much like a database (I'm reminded of dBase II), and you insert control commands (similar to WordStar) within the file to create special effects like boxes, arrows, lines, symbols, font changes, etc. Overhead Express offers two main creation modes: the "Express Editor" contains twelve pre-programmed "Express Templates"; and the "Custom Editor" that allows you to modify any combination of the express templates or create your own graphic display from scratch. You can also create and store your own "custom templates" for quick production of visuals in your own format.

Overhead Express offers four typefaces, each with regular and italic styles, and each in three sizes from 24 dots to 60 dots (about 5mm to 14mm). One typeface, "Modern," also has two additional sizes, 18 dots (about 4mm) and 75 dots (about 17mm). That's a selection from 32 different font choices. Each font has its own punctuation and special symbols, in addition to layout graphics like customized boxes, lines, "bullets," etc.

While preparing your visual, either in the express or custom editor, you can hit the "DO" key to create a schematic visual interpretation on your screen (if you have the graphics option) to check layout. This schematic does not produce text, but a rather

clever graphics display with characters replaced by proportionally sized polygons. It is very useful, though, to check your program. After you create your visual program (and check it schematically), you must save the file and exit back to Overhead Express's main menu. From here you may generate the "presentation" either on your printer or first check it with your graphics screen. By selecting "Review on Screen," Overhead Express will display a reduced version of your visual as it will appear when printed: all fonts are supported, sizing is accurate, and a neatly drawn box surrounds the graphic to show the size of your page and margins. (I wonder what the combination of Overhead Express's screen display and the Polaroid Palette would do!) This process does take about a minute for the program to load all the graphics drivers and fonts. The final step is to choose "Produce Presentation" from the main menu, in order to generate the hard copy at your printer. I wish that space in this review allowed me to include the printed results--you'll just have to take my word for the fact that they're pretty impressive.

While I haven't tried it very much, Overhead Express also provides a "Screen Presentation" mode that will animate a visual presentation on the screen without operator intervention. This is similar to the self-running demonstrations we all see at computer stores. If you plan to set up a Rainbow at some conference or exposition, Overhead Express can create and display your customized message.

The program supports a variety of dot-matrix printers, including the LA-50 and LA-100, Epson MX/FX 80 and 100, IBM graphics, C. Itoh 8510/Prowriter, C. Itoh 1550/Prowriter II, NEC PC-8023-A-C, Okidata Microline 84-II (also Microline 92 & 93), and the Apple Matrix printers. I would have wished for a device library that supported one or two of the more popular multi-pen plotters, such as the HP 7074A, but I really am grateful that BPS included as many (non-DEC) printers as they did. I hope that this broader printer library marks a trend for Rainbow software.

The manual is excellent. It's customized for the DEC Rainbow, and it's well thought-out, clear, and organized. The only negative thing about Overhead Express is that it's copy-protected: once again an example of inconveniencing the innocent. You can make one or two copies (sufficient for a backup disk and transfer to a hard-disk), but that's it. For \$15, you can buy a backup system disk.

N. Jay Bassin, PhD, a writer and consultant on Environmental Management, is the Chairman of the Washington Area Rainbow User's Group. This review appeared previously in the WARUG Newsletter.

(c) 1985 by N. Jay Bassin

PRODUCT REVIEW: WORDMARC
by Caroline M. Mack

WordMarc is a user-customizable word processing program which runs on the DEC Rainbow and PRO as well as Vaxes (including the MicroVax), IBM PC and compatibles, HP 150 and the TI Professional. Formerly known as Muse, it is a powerful, versatile program.

WordMarc for the Rainbow comes with a four-part manual, a Quick Reference Guide, five disks, keyboard stick-on labels, a Function Key Reference Chart for the IBM PC (useless for the Rainbow), and separate installation instructions for the DEC Rainbow. Installation is not difficult, but it can be very confusing. WordMarc consists of more than 130 files. Many (particularly different printer files) are not necessary to operate the program. The sheer number of files could cause panic in someone uncomfortable with computers.

WordMarc is menu driven and provides a self-teaching guide with exercises to help the user get started. Users who are have experience with other word processing programs may prefer to learn

from the User's Guide which provides information in a more concise form. After WordMarc is loaded, the user types in WM, and the main menu appears:

```

EDIT      an existing document
CREATE    a new document
SPELL     check spelling in a document
PRINT    or review a document
MANAGE    files
GLOSS    glossary operations
AUX      auxiliary functions
LANG     language changes
EXIT     from word processing

```

Users who have learned the commands can choose to work without menus.

To begin a new document, the user moves the cursor to Create, and then presses Execute (the DO key). The user is given a choice of document types (the user can also add custom document types):

```

LETTER    format
MEMO      format
MANUAL    format
TXM       format
WIDE     format (132 col)
ASCII    file format
ND        specific format
RETURN   to Main Menu

```

After choosing the type, the appropriate screen appears. Typically there are up to four lines of information at the top of the screen. The top line lists the name of the document, the mode, the page number, the screen number, and the current prompt (if any). The second line displays instructions related to the current function. The third line is a user response line, and the fourth line is the format line. Text is displayed below the format line. WordMarc provides a choice between 80 and 132 columns on screen, but allows a document width of up to 163 columns.

WordMarc operates in typeover mode: if the user backspaces to make a correction, he will type over the characters already there. The user cannot toggle to insert mode. Thus making changes and corrections requires more strokes than in word processing programs which allow the user to operate in or choose insert mode, where existing text is pushed forward as text is added. In addition, the delete key does not delete backward, so you must make several keystrokes to delete even a single character or space. This slows text entry.

WordMarc prompts allow the user to get in and out of files easily. After the user completes creating or editing a document, he presses FILE to get the following menu:

```

OPTION    select printing options
REFILE   this document as changed
NOSAVE   this editing session
SAVE     and name this editing session
ASCII    save this WordMarc document in ASCII format
WM       save this ASCII document in WordMarc format
RETURN   to this editing session

```

The original document can easily be replaced with an edited version, both versions can be saved, or changes can be abandoned. The ASCII and WM options allow the user to import and export files from other sources.

Choosing OPTION allows the user to set the printing options. When the user has hyphenated, added headers and footers, paginated, and merged, he can review the document on screen as it will look when it is printed. WordMarc supports a long list of printers, and allows the user to define other non-supported printers.

WordMarc is nearly a "what you see on screen is what

you get" word processor with most of the features a user would expect in an expensive word processing program: center, bold, underline, character overlay, super- and subscript (up to 13 levels, with the ability to suppress them where it makes a paragraph look messy), search and replace, a choice of hyphenation styles (soft, hard, automatic, semiautomatic, and manual), automatic repagination, right justification, and merge. Two nice features are document encryption and the ability to pop out to the operating system without exiting first from WordMarc. WordMarc also has a recovery feature which minimizes the loss of text in case of an unexpected power loss (maximum loss is 200 characters). Accidental deletions can be completely recovered if recovery is instituted immediately after they are made. Version 4.0 of WordMarc, the one tested, does not have a footnoting capability; however, WordMarc expects to add that function in the future. Some of the WordMarc functions are accomplished in an awkward manner. Page numbering, for instance, is done with headers or footers.

WordMarc functions are centered around the number and cursor keypads, although a few of the Rainbow's function keys are used. For some reason, WordMarc chose not to use some of the six keys above the cursor key in the way they are marked. All is not lost, though: if the user does not like the way WordMarc set up the keys, or prefers not to pepper keycaps with tiny plastic stick-ons, WordMarc functions can be reconfigured to other keys. The user could, for instance, move WordMarc functions to the function keys at the top (excepting Help and Do), and create an appropriate strip instead of using the stick-on labels.

The ability to reconfigure WordMarc is not limited to the placement of function keys. WordMarc includes the source code for menus, prompts, dictionaries, system, and printer definitions. The Technical Reference Guide's instructions on reconfiguring are geared toward the IBM PC, but an experienced user or programmer can edit and recompile the source code files.

The main spelling checker consists of about 40,000 words. (An additional spelling checker, available only if you have a hard disk, contains 20,000 more words.) It goes through the document and flags misspelled words; the user must use the combination of the Search and Spell keys to find the misspelled words. It is sometimes difficult to tell on screen which words have been flagged. WordMarc allows "Reverse" spelling so that the user can check backwards as well as forwards in a document. The spelling check also produces the statistics necessary to determine the document's FOG index (reading level); total word count, average sentence length, and average word length. In addition to the Main spelling dictionary, WordMarc provides the option of establishing a User Dictionary for specialized terminology, and a Document Dictionary. The Document Dictionary is meant for large documents which go through a number of editing cycles. Instead of marking the same words each time, it avoids marking specific words, terms, abbreviations, and acronyms which have already been accepted in a previous document. The dictionary doesn't check single letters, so a typo, such as "t" instead of "it", would go unnoticed; it does, however, mark capital letters found in the middle of a word or not at the beginning of a sentence.

WordMarc's Abbreviation Glossary is similar to boilerplating and macro functions in other word processing programs. Its most obvious use is for recalling boilerplate phrases or paragraphs with only a few keystrokes, but it can also be used to recall an elaborate sequence of keystrokes. By choosing Gloss from the main menu, the user brings up a menu which enables him to edit for the Glossary:

EDIT	or create a glossary item
ERASE	a glossary item
REVIEW	glossary item abbreviations
LIST	prepare a glossary item listing
LOCAL	select the local glossary
GLOBAL	select the global glossary
RETURN	to the main menu

WordMarc can be configured to provide foreign language prompts, menus, and error messages for Dutch, French, German, Italian, Spanish, or Swedish. The foreign language option, invoked through the LANG option on the main menu, must be separately installed for each language, at a cost of \$45 per language.

For a word processor which advertises that it is available on mainframes, minicomputers and micros, WordMarc's lack of a simple file transfer process is baffling. Like many other companies, WordMarc has chosen to make related programs available only for the IBM PC. There are currently no plans to make a Rainbow version of WordMarc's LinkMarc file transfer program, which allows files to be transferred back and forth between IBM PCs and Vaxes.

Word Perfect's documentation is adequate, but not spectacular. It is divided into four areas: the Installation Guide, the Self-Teaching Guide, the User Guide, and the Technical Reference Guide. It lacks a comprehensive Table of Contents to pull the entire document together and make it easier to find information. The Index is short, incomplete, and located not at the end of the manual (after the Technical Reference Guide), but at the end of the User Guide (p. 155). Lack of a good Table of Contents and Index means that the user spends too much time flipping through the manual in an attempt to find necessary information.

One of the places WordMarc really shines is in its ability to include alternate characters within text. In addition to foreign language symbols, WordMarc will allow the user to type complicated mathematical formulas into the text, see them on screen, and then print them out. This facility requires an alternate character ROM. Two are available: WordMarc supplies a ROM which corresponds to the Diablo ECS Elite 12 Scientific printwheel, and DEC, the DEC Technical chip, which corresponds to the LA100 Symbols 10 character set. WordMarc's ROM is \$.

WordMarc's literature states that it requires MS-DOS and 256K RAM for the Rainbow, but it is a huge program which resides more comfortably on a hard disk. (In fact, a hard disk is required in order to use the 20,000 words in the "Additional Spelling Disk.") WordMarc for the Rainbow is not copy-protected, so installing to a hard disk is fairly simple. The Rainbow version of WordMarc is \$; the Pro version, which requires the P/OS operating system, is \$. Updates are \$ each. Discounts are available for quantity purchases. WordMarc charges \$ a year for technical support on its toll-free hotline. WordMarc is available from Marc Software International, Inc., 260 Sheridan Avenue, Suite 200, Palo Alto, CA, 94306, (415) 326-1971.

WordMarc's extreme flexibility is great for the user who is prepared to take advantage of it, but the program is frequently awkward to use. For those who need a word processor which can handle special symbols, or those who want prompts, menus, and error messages in languages other than English, WordMarc may be an excellent choice. WordMarc may also be a good choice for those who need to transfer documents to a wide variety of different computers. For users who prefer a word processor which is easy to use and economical on keystrokes, WordMarc may not be a good choice. WordMarc offers an unusual option for users who do not whether WordMarc will suit them: a one month trial package, including all documentation, for \$.

(c) 1985 by Caroline M. Mack

=====

=====

PRODUCT REVIEW: WRITER'S PACK
by N. Jay Bassin

Say what you will about WordStar; a real cottage industry exists out there writing supplementary software to be used with that venerable (if somewhat obsolete) word processor. The folks at Digital Marketing (2363 Boulevard Circle, Walnut Creek, CA 94599; 415/947-1000) have packaged a set of six such programs, written by three different sources, into one product. Writer's Pack includes Footnote, Pair and Bibliography (Protom Software); The Random House Proofreader and Grammatik (Wang Electronic Publishing); and Index (Orthocode Corp.). Each of the programs pretty much stands on its own (remarkably, some of them are even incompatible with one another!), and are designed to supplement the capabilities of WordStar, Select, and other CP/M word processors. I will give a brief overview of all six, and then discuss in some detail each of the programs. My personal favorite of the bunch is Proofreader. Writer's Pack [\$ 49; CP/M-86/80 or MS-DOS; 64k] includes five diskettes and an effective manual.

As its name implies, Footnote is a utility that numbers and manages the placement of footnotes. It is fairly easy to use, and allows placement of notes either at the bottom of the proper page or collected at the end of the paper. A particularly nice feature is its ability to run footnotes over several pages. As footnotes are added or deleted, the program automatically corrects the numbering. For someone who uses footnotes extensively, this is a very handy program. Pair is a companion to Footnote only in the sense that it appears on the same disk. Pair checks a WordStar file to ensure that paired control characters (such as bold and underlining toggles, etc.) or parentheses, quotation marks, or brackets match.

Bibliography is a library utility that checks a WordStar file for embedded "key words", matches those key words to entries in a separate (user-created) master bibliography, and constructs a document bibliography of all names cited. There are a number of different formats which may be selected. If you're into literature citations in a big way, give this program a good look.

Grammatik bills itself as a "grammar checker" and "beyond a spelling checker." It checks for common typographical errors such as a "Sticky sShift key," doubled words ("the the"), and proper punctuation. It will check WordStar files against two included "phrase dictionaries" that contain about 600 wordy, awkward or sexist phrases and suggest alternatives. Finally, Grammatik will display a summary table for your document, listing the number of words, sentences, problems detected, the average sentence length, etc. My advice here is to read the phrase dictionaries to see if you are likely to use any of the poor constructions. If not, this program is probably not worth the time.

Index reads a WordStar document file into which you have embedded special "dot commands" and creates an index and/or table of contents of these key terms. The index and table of contents are written to separate files for subsequent editing or printing. The manual for Index is the most confusing of the series, and a WordStar file prepared for indexing may generate problems if you try to use the Footnote program on it. Index does produce nice-looking indices and indented, nested tables of contents. A unique feature of the indexing program is its ability to cross-reference key words. However, it takes a lot of work and thought to place index keys in every instance where a word should be cited. How many 20-page reports do you see with a complete index? If you're writing a text book let your publisher index it on a mainframe.

The Random House Proofreader

The Random House Proofreader is an excellent dictionary program. I used to think that I hated dictionary utilities because my writing is often very technical and the usual spelling checkers

with 30,000 words or so seemed to stop every other sentence to ask about such words as "precipitation." I was therefore quite pleased with the results from Proofreader. I saw in a test run on a 5,000 word technical paper. The Rainbow version contains a giant dictionary of over 84,000 words; one of the largest I have ever seen. It is remarkably fast, much quicker and more precise than either MicroPro's SpellStar or the spelling checker in WordPerfect. The program works on ASCII files as well as WordStar, and generally recognizes the "soft hyphens" and soft carriage returns in the word processing file. Unless you install it on a hard-disk, you will have to switch diskettes to use it: Proofreader takes 234KB of disk space in five files, including 182KB in the dictionary file alone. Proofreader is written for CP/M-80 and therefore may not be accessed from within WordStar version 3.xx with the "R" command from the main menu: you must initiate the program from the CP/M prompt. For the same reason, it cannot be transported to DOS and used with AME86.EXE, although WordStar itself can be.

Proofreader is quite easy to use. Assuming the program is in drive A and your document "FOO.TXT" is in B, simply type "PRF B:FOO.TXT<cr>". Proofreader immediately reads FOO.TXT, displays the total number of words and the number of unique words, and sorts the words alphabetically. For a document of over 1000 unique words (perhaps a 5000-6000 word document), this sorting may take up to ten seconds. After sorting, Proofreader checks each unique word against its master dictionary (PRF.DIC). The 80,000+ word dictionary takes about three minutes or so. Proofreader then searches an auxiliary dictionary (PRFDIC.AUX) that is user-created. (PRFDIC.AUX comes ready-supplied with all the WordStar dot-commands and control characters.) The auxiliary dictionary will normally grow automatically with use of Proofreader. After checking through the dictionaries, Proofreader will display the number of "unknown" words it has found, and will offer a menu for continuing. While you can just dive in with option "C" and correct words in context, I suggest that the first step normally be option "R"--reviewing each "unknown" word in sequence, but not in context. That way, many unknown but correct words such as proper names or intentional misspellings can be eliminated. As each word is displayed, you may acknowledge that it is incorrect or type "N" to indicate it is not correct. After this run, control returns to the main editing menu. If you then type "C" to correct questionable words in context, you will get another menu with 7 choices: (C) correcting the word; (D) asking for dictionary assistance; (L) accepting the word and adding it to the auxiliary dictionary; (A) accepting the word for just the rest of the document; (I) ignoring the word just this time; (Q) aborting the session without saving any changes; and (E) exit and save all changes so far. Proofreader has some pretty nifty features in its correction routine. First is the fact that unknown words are displayed in context of two lines of text, so there is no doubt about what you meant. Another is that a correction itself is checked against the dictionary to make sure you don't compound the error. Run-together words ("theonly") can be corrected and words may be deleted by just hitting RETURN. Finally, all subsequent identical errors are automatically corrected! Proofreader will make corrections to the file and generate a ".BAK" file containing the original document.

What won't Proofreader do? For starters, it won't re-form a paragraph if, for example, the document was right-justified. You will have to go back to WordStar and reform with the ^B. If you are working with a justified document, you can select one of two other options that will just "Mark" unknown words (with an "#") for later searching. For another, it ignores punctuation errors, case ("lake Erie" will pass), unpaired quotation marks, parentheses and brackets, and doubled words. It won't catch improperly hyphenated words or word-pairs. Finally, and most awkwardly, it doesn't recognize "soft hyphens" where they occur at the end of a double-spaced line. All-in-all, I think that The Random House Proofreader rivals the best spelling checkers available. If all you need is a speller, try to get this one without the package deal.

Footnote

As its name implies, **Footnote** is a utility that numbers and manages the placement of footnotes. It is fairly easy to use, and allows placement of notes either at the bottom of the proper page or collected at the end of the paper. A particularly nice feature is its ability to run footnotes over several pages. As footnotes are added or deleted, the program automatically corrects the numbering. **Pair** checks a WordStar file to ensure that paired control characters (such as bold and underlining toggles, etc.) or parentheses, quotation marks, or brackets match.

Footnote (CP/M-86, 64k) runs with two files: FN.COM is the actual program, and OPTIONS.FN contains a user-changeable overlay that defines function keys and control-commands used by FN.COM and PAIR.COM. (Because FN.COM and PAIR.COM are written for CP/M-86, they may be called from the WordStar main menu with the "R" command and/or transported over to MS-DOS and used with AME86. If you use **Footnote** often, it would be advantageous to install it on your WordStar disk.)

Footnote works by marking "calls" in the text wherever you want to refer to footnotes with the symbol "@." As you type your manuscript, you delineate text areas from "note" areas with familiar WordStar non-printing controls embedded in the text. (You may enter the footnote itself either in the text as you go along or in a separate file.) If the notes are within the textfile, notes are delimited by a "R" and a ".PA" on the left margins, with the consecutive notes between marked with an "@." As a result, page layout (as measured by WordStar's page numbering on screen) is confusing. A more important factor is that a WordStar text file with embedded **Footnote** commands cannot be printed cleanly except after completing the "format" procedure. After saving the WordStar file, you run **Footnote** in order to consecutively number and match each "call" to its note. The opening menu offers six choices: N)umber notes; F)ormat footnotes in a textfile; R)emove notes from the textfile to a separate endnote file; M)erge endnotes into the textfile; C)hange options; and e)xit to the system.

The first step is to N)umber the notes and calls to ensure that they match. After doing so, you may re-enter WordStar to check, but **Footnote** will let you know how many notes it created. If you created a textfile with included notes, you may choose to R)emove the notes to an endfile, or conversely you may choose to M)erge the endfile of notes into the textfile. If you want footnotes formatted at the bottom of the proper page, you must then F)ormat the document before you can print it. In formatting, **Footnote** creates a temporary working file on the disk that contains the textfile. You must therefore ensure that the disk is neither write-protected nor too full. After formatting, **Footnote** has created and saved a print-ready document that uses soft carriage returns to "design" each page. It marks each page with a ".pa" that WordStar uses to define a page-break, and fills in any space between the notes and the bottom of the text with carriage returns. Fortunately, **Footnote** creates a ".BAK" file from your non-formatted textfile. **Footnote** allows you to select any of three numbering schemes: Superscripted1; superscripted and underscored2/; or just underscored and slashed3/. You also have a separate choice over how the numbers will appear in the notes.

If I used footnotes regularly, I would really like this program. It's better than the footnoter in version 3 of WordPerfect, and for some additional work in preparing the manuscript, produces professional-looking page layouts. However, it will only work with WordStar (and maybe Select) files.

Pair

Pair reads WordStar (and maybe Select) textfiles, locates any control character (such as ^S or ^B) or text markers--(,),[- ,],(,),"-and tries to match them to a counterpart. If there are any unpaired marks within 200 characters (changeable by the user), **Pair** marks the 200th character with a ")" (also changeable) that

you will have to search for using WordStar's global search function. **Pair** must be configured to search for each mark. In other words, if it's set for ^S (underline), it will not find a missing quotation mark. It can get a bit tedious to run the program a dozen times, reconfiguring each time, just to check for pairs. Realizing this, Pro/Tem included directions for creating "batch processing" commands as a "submit" file in CP/M. By following their instructions, you can create a command file that will search for any number of markers. The catch, though, is that it will search for only one at a time, from beginning to end, and then repeat with the next. **Pair** ain't worth it.

Bibliography

Bibliography is a library utility that checks a WordStar file for embedded "key words," matches those key words to entries in a separate (user-created) master bibliography, and constructs a document bibliography of all names cited. If you do a lot of technical writing with citations to other authors or references, this is for you! The most striking feature about it is its "database" format for the master bibliography. While you can create any number of separate bibliographic files, you can also put your entire reference library on one file, and write any number of separate papers all drawing from the master file! Where was this gem when I was in graduate school? The first thing to do is to create a bibliography (using WordStar or any word processor). Each entry begins with a "%KEY:" heading that identifies a unique "keyname." This keyname will be used in your manuscripts to call this particular citation. The bibliographic entry continues with additional user-defined headings (e.g., author, date, title, publisher, etc.) and an "%Annot:" or comment header that could contain a short abstract. Each entry (record) ends with a "%:".

When you're writing your paper, you include a reference to the appropriate bibliographic record with a citation such as "%Einstein, 1905". The citation is identical to the matching keyname heading in the bibliography except that it begins with a percent sign and ends with one of these four characters:), /, i, or . Thus, all of the following citations will match the keyname "Einstein, 1905":

```
(Einstein, 1905)
%Einstein, 1905)
%Einstein, 1905: p. 235
(%Einstein, 1905;
As Einstein (1905) wrote...
In Einstein's (1905) paper...
```

Bibliography's menu will display the current settings for nine separate functions. For example, you can choose to omit the annotation from the list of references (probably a good idea); number each entry in your list of references; alphabetize your list of references; etc. Once your settings are correct, you hit RETURN and type in the name of your article, the name of your library file, and the name of a new file to contain the sorted list of references. This last file may then be merged into the end of original text file if you wish. If you embed a keyname that is not contained in the library, **Bibliography** will create another file called "CANTFIND.BIB" with the missing citations. With **Bibliography** you have quite a choice over format for your list of references. Rather than try to list them all, I'll just say that every journal style I'm familiar with is included, from alphabetical to consecutively numbered. After you finish modifying your paper and are ready to print it, run **Bibliography** one last time to remove the "%s embedded in the text." (You can also do a global search/replace to delete the symbols within WordStar.) **Bibliography** will create a ".BAK" file each time it modifies the text.

As if this weren't enough, Pro/Tem includes in the diskette three additional utilities: SORTLIB.COM, MERGE.BIB.COM and LISTKEYS.COM. SORTLIB alphabetizes your master bibliographic library. This means you can add records anywhere and sort the

whole thing. Pro/Tem says that with 128k of memory, you can sort about 1200 entries. MERGEIB creates a single alphabetized bibliography from a number of different bibliographies; for example, if you wanted to merge separate chapter bibliographies into a single bibliography for the whole book. LISTKEYS generates a list of all keynames in in library. Bibliography, written in CP/M-86, can also be run from the WordStar opening menu and it is compatible with AME86. The manual is pretty clear--certainly above average for this type of product. My conclusion is that this program is a winner!

Grammatik

Grammatik bills itself as "beyond spelling checking;" a "system that checks for typographical errors and analyzes writing style." Grammatik reads WordStar (or ASCII) files and searches for words or phrases listed in user-changeable dictionary files. In addition, it checks for proper capitalization (for example, the first letter of the first word after a period, or a "sticky shift" key), paired punctuation marks (), " , [] , { } , and doubling of words or punctuation such as "the the" or " ; ; ". Naturally, the program assumes that the words are all spelled correctly to begin with.

The distribution disk comes with six separate files: GMK.COM is the main document analysis program; PHRASES.GMK is the main phrase dictionary; SEXIST.GMK is an auxiliary dictionary of "sexist," or gender-dependent, words; PROFILE.COM is a separate program that will list each separate word in the document together with the number of times it's used; SORTDICT.COM, a utility that alphabetically sorts phrase dictionaries (or any ASCII file with records on separate lines); and SAMPLE.TXT, a demonstration file.

Grammatik's main menu lists 12 choices to configure the program's run. Each execution must have the name of a phrase dictionary and the name of the document file; Beyond that, the user may select the method of error listing, output file, or categories of errors to exclude (if desired). Multiple phrase dictionaries may be "read in" to memory. Each time Grammatik matches a poor sentence construction, it stops (if desired), displays the offending word or phrase together with the sentence number (if desired), category of error ("redundant," "wordy," etc.) and, optionally, one or more alternatives. If an output file has been named, errors will be marked with an "?#e", where "e" represents a one-letter symbol for the type of error. Grammatik won't allow you to correct a sentence interactively. At the end of the run, Grammatik will display a summary table listing the number of sentences, number of words, average sentence length, average word length, number of questions and exclamations, the number of sentences with fewer than 14 words or more than 30 words, the shortest and longest sentences, the number of prepositions, and the number of sentences with conjunctions of the verb "to be." I leave it to you to decide whether these statistics are useful.

How well does Grammatik work? I modified the SAMPLE.TXT file to throw in some additional errors similar in type (but not identical) to those already there. Most of the poor constructs I added (my favorite was "...like, for instance,...") were not picked up because they were not explicitly listed in the phrase dictionaries. This leads to the software's biggest failure. Grammatik does not know the meaning of any words. It only recognizes poor construction if that exact phrasing is located within a dictionary. If you are unaware that "...like, for instance,..." is redundant, Grammatik won't catch it because it's not in the distribution dictionary (which contains, by the way, about 500 phrases). If you are aware of that, you probably wouldn't be using it in the first place. Unlike a spelling checker that questions anything it doesn't recognize, affording you the opportunity to correct it or to have the dictionary "learn" the new word, Grammatik will not question anything it doesn't recognize! Grammatik cannot recognize subject-verb agreement ("I doesn't think so") or even whole sentences ("Works with text files too."). Grammatik is also supposed to check paired punctuation marks. In the 10-sentence test document I tried, there were three intentionally unpaired quotations and parentheses. The program picked them all up, but failed to

properly identify which sentence they were in: all three were attributed to the last sentence (which, coincidentally, did not have any unpaired punctuation at all)! Grammatik is written for CP/M-80, so it cannot be accessed with the "R" command from WordStar's main menu.

For the same reason, it is not compatible with AME-86, in case you're happily using WordStar under MS-DOS. My recommendation is to give Grammatik a miss.

Index

Index reads a WordStar (or ASCII) text file into which you have embedded special command strings that identify key words and creates an alphabetized index and/or table of contents of these key words. In theory, it sounds great. In practice, like other programs in this package, the return may rarely be worth the investment of time. Index, like Grammatik, is written for CP/M-80 and also may not be used from the WordStar menu or from AME86. Because Index treats table-of-contents generation quite differently than index-creation, I will discuss them separately; tables of contents first:

Index uses, as its primary embedded controls, WordStar-like dot commands with three dots. WordStar interprets any line beginning with two (or more) dots as a non-printing comment. When generating a table of contents, you are instructed to put a "...T" command in columns 1 through 4 of the line immediately above the line to be entered in the table of contents. Any WordStar printer controls (such as ^B for boldface) are legitimate in Index also. The table of contents can have indented minor headings if you choose, up to 16 levels. For example, if you have a bolded heading ^BINTRODUCTION^B that you want in your contents table as a third-level subheading, you would type "...T3" at the left margin of the line immediately above the heading. Index will automatically collect all such headings, note the page number each is on, and generate a text file with the formatted table of contents. It gives you a wide selection of formats to choose from and customize. There is one catch, though it's a minor inconvenience: the table of contents heading must be alone on the line. If you format your text with a minor heading that starts a line of text, the entire line will end up in your table of contents and you will have an editing job on the generated table of contents file. Similarly, if your heading is underlined or bolded as in the example above, and you do not want the word underlined or bolded in the contents, you will have to edit the output file. If you use tables of contents, this little program is relatively easy and does pretty work. Since the 3-dot commands are compatible with WordStar, you lose no flexibility in just using Index for tables of contents.

Index's index provision, the primary purpose of the program, is about as straight-forward. However, the accuracy and thoroughness of the index is entirely dependent upon the user, not the program. This means that you will have to mark every single instance of a particular term throughout the document; all Index does for you is to keep track of the page numbers. In practice, the text file is edited to insert index calls either above or below the paragraph containing the word to be indexed. The index call also uses a three-dot command, in the form "...X text" where "text" stands for the word or phrase you want indexed. In this most-simple case, Index will create an alphabetized list of your calls together with the page number that the call is on. For example, if you have a paragraph discussing computers on page 4 of your document, you could put the command "...X Computers" above the paragraph and "Computers, 4" would appear in your index. If you also had that same call next to a paragraph on page 8, the index would show "Computers, 4, 8". If two or more index calls exist on one page, Index will list that page only once. The calls must be preceded and succeeded by hard carriage returns so as not to lose their place on the left margin during paragraph reformatting. For this reason, the calls must not fall within a paragraph or else the paragraph cannot be reformed after text editing. It's important to realize that Index does not "recognize" the key

word: It will not automatically index every subsequent use of the terms "Computers" unless you type the same call into every page where the keyword exists. In revising long paragraphs, it is easy to end up with a call on the page before or after the one with the actual use of the word. In this case the index will be misnumbered. Frankly, I found it extremely tedious and irksome to insert index calls at every paragraph a key term is used. I don't know why Index couldn't have just been made to "remember" key words once, and automatically index every instance of them. Index allows nested, multi-level entries. For example, if you discussed Rainbows on page 6 and a VAX on page 9, you would insert a call of the form "...X Computers, Rainbow" and "...X Computers, VAX" in the appropriate places. Your index would look like this:

Computers, 4,8
Rainbow, 6
VAX, 9

Index allows up to eight nested levels in this fashion. Index also provides separate calls for "reference" entries that produce an index line such as

"Micro-computers, see Computers, Rainbow," or

"Z80 Processor, see also Computers, Rainbow."

Because these reference entries are not page-specific, you need insert the calls only once anywhere in the document.

Index does quite a lot, but in a very labor-intensive way. It produces nifty looking contents and indices, but let's face it: few of us write documents long enough to require an index. If you're writing a 400 page text book, I think you'd go blind before putting in all the necessary calls. Let your publisher do it with his mainframe typesetting computer.

Summary

To recap this review of Writer's Pack, I really like Proofreader and Bibliography. Footnote is a very good program for someone who does a lot of footnoting, but is too complex to bother with if you have only one or two footnotes. I would use Index solely for its table-of-contents provisions and ignore its indexing capabilities. Grammatik and Pair (which comes on the Footnote disk) are not very useful at all. A quick call to Digital Marketing's help line confirmed that each of the programs are available separately: Proofreader for \$; Grammatik for \$; Bibliography and Index for \$ each; and Footnote and Pair together for \$. The bundled price of \$ is a savings only if you can use all the programs. If you skip any two (and I urge you to skip Grammatik), you can get everything else individually for less!

This review appeared previously in three parts in the WARUG Newsletter.

(c) 1985 by N. Jay Bassin

=====

PRODUCT REVIEW: INVESTMENT MANAGER

by Ken Gordon

Have you ever wanted to use your Rainbow to analyse and manage your investments in stocks, bonds, and other financial securities? There is now a program available for that purpose which runs on the DEC Rainbow using MS-DOS (requires 128K memory). I have recently tested Investment Manager, the only Rainbow version of an investment program that I am aware of.

Investment Manager is a portfolio management program designed by the Walton Group of Boston, and distributed by International Microcomputer Software, Inc., (IMSI), 633 Fifth Avenue,

San Rafael, California, 94901, (415) 454-7107. The list price is \$, which is tax deductible if you use the program to manage revenue producing assets.

Although the program was designed for professional investment managers, with full capabilities, it is feasible, although perhaps expensive, for individual investors as well. As an alternative, you can design programs with similar functions using general database programs such as dBase II or Lotus, but it is certainly convenient to have standardized menus, pre-formatted reports, well-tested routines, helpful documentation, and built in communications utilities. If you want to analyze the historical behavior of the stock market to help make investment decisions, you will need a different type of program, usually known as analytical programs.

You start by entering information on each stock, bond, money fund, mutual fund, or other securities in your portfolio into the database. The program asks for name, amount, ticker symbol, date purchased, original cost, and current value. This is done only once for each item. As you make new investments or sell them, you enter them into the database. Whenever you wish, you can enter updated quotes and dividends which modify the existing values (I'll discuss automatic updates later).

The program provides for five reports, which you can display or print at any time. The reports provide the status and performance record of each security, and of the portfolio as a whole. The reports offer an up-to-date valuation, the results of calculating several performance measures (such as rates of return and impact rankings), and lists of appropriate transactions for annual tax and audit purposes. The reports allow investors to compare the performance of their investments with alternative investments, or with the market as a whole. Investment Manager will handle multiple portfolios, and a wild card symbol allows you to combine several portfolios in a single report. For example, you can keep stocks owned by your spouse and children in separate portfolios, but request the the family's securities be combined in a single report.

I particularly liked the automatic updating feature. To use it, you will need to have a modem, and to establish an account with one of the financial data services (I used Warner Computer Systems Data Services). You can use the built-in communications utility to dial-up the service, automatically look up the quote for each security in your portfolio, modify your database, and hang up. Once I became used to the process, it took only 30 seconds at 1200 baud to update a portfolio with about two dozen securities. During non-prime time, the cost was about \$1.00 for the data service, plus a 30 second telephone call to New York City. That sure beats looking each security up in the Wall Street Journal and doing a lot of multiplying and totalling. If you like to "count your money," you can update weekly, or even daily. Warner Data Services includes daily quotes for everything reported in the Wall Street Journal plus several thousand bonds.

The advertisements for this program stress one feature I found less than useful: the ability to handle Beta values, which is a ratio indicator designed to quantify each particular security's volatility, or the riskiness. Only a few sophisticated analysts will find Beta values useful. The rest of us can simply ignore them without affecting the other capabilities of the program.

You will need to be creative to use some of the capabilities the program provides to handle some financial situations you encounter. For example, if you want to use the calculations of compounded rate of return, you must enter all of the dividends or interest you have received since purchasing the security (which could involve a number of years). The program cannot handle a situation where you have your dividends reinvested, or they are used to buy more stock, unless you periodically increase the number of shares owned manually, or you create a new purchase entry covering additional shares.

There are several operational aspects I should mention. The program is not copy-protected, so you can put it on your hard disk without difficulty. The speed of operation was excellent when

using my hard disk. In the original disk I received, the loading/installation function did not operate, but a call to the distributor brought a subsequent version that did operate. When I called later to find out what data the automatic update program required for bonds, I was referred to Robert Walton, the program's designer, who graciously sent me an enhanced version of the communications program. It appears that enhancements to the program continue to be made. Although there is no toll-free 800 number, adequate technical support is provided. I found the user's manual poorly arranged and less complete than is desirable. For example, the significant feature which allows you to combine portfolios is only described briefly at the end of another topic, and it is not indexed at all.

In summary, Investment Manager is a worthwhile program which functions well for those who need to monitor a portfolio of securities. Because you will need to enter data on your own portfolio and to set up accounts/procedures for automatic updating, you should expect significant front end effort before reaching the operational stage. However, the automatic updating feature is one of the plusses of the program. The greatest minuses are the mediocre documentation and the relatively high price. Given the limited number of proprietary programs adapted for the Rainbow, we should appreciate this effort. If Investment Manager meets your needs, I urge you to use it.

Kenneth Gordon, PhD., is an economist with the United States Government Bureau of Standards, and is the co-founder, with his wife, Dr. Barbara Gordon, of the Washington Area Rainbow User's Group.

=====

PRODUCT REVIEW: DO-IT
by Dennis K. Fitzgerald

What is Do-It? Do-It is a program to enhance the usability of your Rainbow. With Do-It, you can suspend the program you are running, perform one of several functions, and then return to your original program where it left off. The functions include a notepad editor, a calculator, save screen, print screen, terminal emulator, and the ability to run one or more MS-DOS commands or programs. You install Do-It by running the DOIT command once each time you reboot your system. (Most people put the DOIT command in the AUTOEXEC.BAT file.) To invoke Do-It, you just type CNTR-SHIFT-ENTER.

Notepad

The Do-It Notepad allows you to edit a single screenful of text. It can be invoked through Do-It or run as a stand-alone program. When you invoke Notepad through Do-It, the screen contents remain just as they were before Notepad was invoked. This allows you to include text from your application screen as part of your "note."

Notepad support Rainbow screen capabilities such as bold, blink, underscore, and reverse video. Double width and double height characters are also supported. The screen can be set to 80 or 132 columns. Notepad also supports the line drawing character set. It allows you to save the screen image in one of two modes: an Image mode that supports all character and line attributes, but is readable only by Do-It, and a Text mode that supports only ASCII text. You can thus save the screen image in whatever mode is most appropriate to your needs.

Notepad makes extensive use of the Rainbow function keys. In particular, PF1 through PF4 and F17 to F20 are used to turn character and line attributes on and off, and the numeric keypad is used to enter line drawing characters. When entering line drawing characters, you can select the direction that the cursor moves following each character. This makes it easy to draw vertical lines or to draw a horizontal line from right to left.

Calculator

The Do-It calculator uses the Rainbow numeric keypad to emulate a Reverse Polish Notation (RPN) calculator. This type of calculator has an avid following, but can be confusing to someone who is not used to it. To add two numbers with this type of calculator, you type the first number, hit the Enter key to put it on a stack, type the second number, and then hit the "+" key to add it to the top of the stack. I personally prefer the more-familiar infix notation where you type the first number, the "+" key, the second number, and finally the "=" key.

When the calculator is active, the screen shows a picture of the calculator with each of the keys labeled. Each key has two meanings with the PF1 key acting as the prefix key (a usage that is familiar to users of other DEC systems and of the WPS and EDT editors). The calculator supports a stack of four numbers, a data entry area, and one memory. One advantage of this calculator over a real, hand-held calculator is that the contents of the stack and memory are displayed on the screen at all times. Functions include the usual =, -, *, /, and %. In addition the calculator supports 1/x, square root, and square functions. Memory functions include store, recall, clear, add to memory, and divide memory, by display value. Stack functions include roll, exchange, enter, change sign and clear. A special function is provided to specify the precision of the display. Another function, Undo Last, allows you to recover from an erroneous operation. The Help key provides an expanded explanation of each key.

I find this calculator sufficient for my needs, but other users have complained that it needs more functions (eg., sine, cosine, etc.) to be more useful. My suggestion to someone who needs these extra functions is to get GW-BASIC or something equivalent and run it as a second application using the "Run a Program" option from the Do-It menu.

Save and Print Screen

The Save Screen function allows you to save the screen to a disk file for future use. As with Notepad, the screen can be saved in Image mode (with attributes) or in Text mode (ASCII only). I use Save Screen when I am generating user documentation for programs. When I want an example of what the screen looks like at a particular point in the program, I just use Do-It to save the screen in Text mode. It is then a simple matter to include the text file in my program documentation. Save Screen is available from the Do-It menu and also can be invoked within Do-It Terminal and Notepad using the F4 function key.

The Print Screen function allows you to print the contents of your screen on the printer. The only fully supported printers are the DEC LA50 and LA100, but any printer should work except when double width characters and/or line drawing characters are present on the screen.

Terminal Emulator

The Do-It Terminal emulates a VT102 terminal with the added ability to save the text of your terminal session to a file or the printer. A key feature of the emulator is the use of a buffer between the communications port and the display. This buffer stores all undisplayed data (within the limits of the memory you allocate to the buffer when Do-It is installed) even when you are not in terminal mode. This allows you to start a terminal session, return to your application, and then return to the terminal session. You can then display or save the contents of the buffer. You can also save the contents of the screen to a file. Printer echo can be turned on/off using the Addnl Options function key.

Command/Program Execution

Probably the best feature of Do-It is the ability to run an arbitrary program or programs and then return to your original

application. Do-It provides two menu options to support this feature: the "Run a Program" option to execute a single command, and the "MS-DOS Command" option to execute a series of commands. In the latter case, you type the MSDOS EXIT command to return to Do-It and your original application.

General

Do-It has on-line help for all Do-It functions. Help is invoked through the use of a series of screens that can be sequentially or randomly accessed. The help screens were created using Notepad and can be customized and/or extended very easily. A two-page help screen is provided that includes an ASCII table with decimal, hexadecimal, and octal equivalents of each standard 7-bit ASCII character. Unfortunately the table has several (more than ten) typographical errors. It was easy to correct the two help files using Notepad, but I expect better from an otherwise high quality product like Do-It. (The two corrected help files, HDI.IT and HDI.US, are available through the WARUG bulletin board and will be available in the WARUG public domain library.)

Do-It comes with an excellent 65-page User's Guide. It is well written, accurate, and makes good use of typographical elements such as size, font, and emphasis to make the guide easy to read and interpret. It includes detailed instructions on installation of Do-It which should be carefully read. For those who can't wait to try out their software, Chapter 2, Off to a Fast Start, provides enough information to get you started. But follow the instructions exactly, including running Do-It from a floppy, until you have read the installation instructions. There are separate chapters on each of the functions, a chapter on customizing the help files, and a chapter that addresses possible problems. The guide has a table of contents, but no index.

Do-It works on any Rainbow 100 with at least 256K of memory, but because Do-It takes up 50K for its resident portion and needs an additional 65K to run Notepad and the calculator, you end up with less than 128K left to run your application programs. Since I've upgraded to 896K, I allocate 256K to a RAM drive, 128K to MS-DOS and Do-It, and split the remaining memory into two 256K partitions. One of these is used to run my primary application and the other is used by Do-It when it runs Notepad, the calculator, or a second program. I can highly recommend Do-It in such an environment, but hesitate to recommend it for a 256K system unless you run small (128K or less) programs.

Do-It requires MS-DOS 2.05 or 2.11, and about 207K of disk space (244K on a hard disk because files on the hard disk have to be multiples of 2K bytes). The four .EXE files that make up Do-It require 82K (84K on hard disk). The twenty help files use up an additional 125K (160K on hard disk). If disk space is a problem, some or all of the help files can be left off.

Make sure you get Do-It version 1.03 or higher. Earlier versions had a "feature" which prevented you from putting any commands after the DOIT command in your AUTOEXEC.BAT file. This caused a problem for me since one of the public domain programs I use interacted badly with Do-It unless it was installed after Do-It. Until I got version 1.03 I had to run the public domain program (SEARCH) by hand each time I booted the system.

Sometimes when you invoke Do-It, it sends a carriage return to your application (or to MS-DOS) when Do-It exits. According to VuSoft, this is an unavoidable feature of MS-DOS. That is, when your application is waiting in MS-DOS for a character, Do-It cannot get at MS-DOS to file input or output unless the character input request is satisfied. This does not happen with all applications, and is not too serious in many cases as long as you are aware that it is happening. (It happens when I am using my favorite editor--I get an unwanted new line each time I use Do-It.) I can think of situations where it would be disastrous, however, and would like Do-It a lot better if this problem did not exist.

You should also be careful when using Do-It to run another application program. While in the middle of typing this review into my word processing application program, I decided to do some

additional evaluation of the Do-It Terminal emulator. So I invoked Do-It, typed "R" to run a program, and then ran my autodialer program. I intended, once the phone number had been dialed, to return to my document, and then invoke the Do-It Terminal program. My autodialer program, for reasons which have nothing to do with Do-It, occasionally hangs up the Rainbow so that I have to reboot to get it going again. This is annoying, but ordinarily not a major problem--EXCEPT when you use Do-It to run the program from the middle of a word processing session. Of course, it had not occurred to me that this might happen, and I had not saved the last set of changes to the review. Fortunately I had been experimenting with the Do-It Print Screen function and had printed about half of the new material before clobbering everything. The moral here is that, while Do-It was not at fault in any way, it made the consequences of my error much more serious. I recommend that you save whatever you are doing, if possible, before using Do-It--just in case.

VuSoft plans to continue support for Do-It on the Rainbow and that the next version will probably include the addition of file transfer capabilities to the Do-It Terminal.

Conclusion

Do-It is a welcome addition to the software available for the Rainbow. While it is slightly expensive compared to similar products for the IBM-PC, it is, at least, available. Except for the problem with sending a carriage return to your application in certain circumstances, it is an excellent product. The more I use it, the more I like it, and the more uses I find for it. I can recommend Do-It to anyone who either has much more than 256K or who only runs 128K programs. I congratulate VuSoft on a good, well documented, product. Do-It, which was written specifically for the Rainbow, lists for \$199.95 and is a product of Vu-Soft, 248 Tower Road, Lincoln, MA, 01773, (617) 259-0666, and is also a DDS (Digital Distributed Software) product.

This article originally appeared in the WARUG Newsletter.

(c) 1985 by Dennis Fitzgerald

=====

WHATEVER HAPPENED TO CONCURRENT CP/M?
By Willett Kempton

A Brief Review of Concurrent CP/M

Concurrent CP/M is a multi-tasking operating system for the Rainbow. It allows up to four programs to be run at one time. When only one program is running, it will run approximately at the same speed as under CP/M-86/80. With two, and especially three or four CPU-intensive programs active, slowdown can be noticeable. On the other hand, a disk-intensive program and a CPU-intensive program can run at the same time with little slowdown of either. Similarly, a modem program can be downloading through the comm port while you run another program, with each running at essentially full speed. The only comparable facility MS-DOS is printing while running another program. Unlike MS-DOS background printing, CCPM does not freeze up the keyboard for seconds at a time while the other process runs.

DEC took Digital Research's generic 8088 Concurrent and adapted it to the Rainbow. Their port was excellent--probably the best version of Concurrent for any machine. For example, DEC added a "system monitor" screen which can check memory use, status, and resource use (disks, printer, comm port) of each of the four processes. From any process, a "screen print" utility can be run which will print (or save to file) the current screen of any other process. Each process has its own Set-Up parameters, so, for example, one process can be in 80-column mode while the other is in 132-column mode.

Most, but not all CP/M-86 programs will work. Notable exceptions include MAINT and RED. CP/M-80 programs will not work. High speed video works, as long as it is done via the system calls rather than via direct memory access. Graphics programs can build an image on a graphics screen while other processes run non-graphics programs.

To run multiple programs simultaneously requires the sum of the memory required for each program, plus 164K for Concurrent CP/M itself. A 256K machine cannot really utilize Concurrent, whereas 512K does quite well for moderate-size programs. The Rainbow's ability to go to 896K, rather than just 640K, becomes very useful in a Concurrent environment.

Digital Research's Technical Evaluation of DEC's Port¹

"DRI to retail DEC Concurrent with GSX"

In cooperation with Digital Equipment Corporation, Digital Research will market Concurrent CP/M with GSX for the DEC Rainbow and Rainbow Plus personal computers. The product will be available in DRI packaging in January, 1984. DEC has provided "a very professional and sophisticated implementation of Concurrent CP/M," said Allen Beebe, director of DRI's System's Software Division. According to Beebe, DEC's software engineering team "did excellent work. Al Perron, who was engineering manager, and his staff have brought many minicomputer standards to microcomputer software with their XIOS for the Rainbow version of Concurrent." Chuck Carroll, OEM Systems manager for DRI, outlined some of the features of the DEC Concurrent. "It supports five virtual consoles," he said, "one of which can be dedicated to a system status function, so you can always switch to console zero to see what's happening on the system." The DEC implementation also passes the full 16-bit keyboard value back to the programmer, so that ISV's have complete control of the keyboard. Full VT100 compatibility is provided, including smooth scrolling and double size characters. A sophisticated set-up menu controls scrolling, character size, baud rate, and other settings. In addition, an on-line HELP function, accessed by a single stroke HELP key, functions as a tutorial on how to use Concurrent CP/M. Also, Carroll added, the implementation supports a 10 MB hard disk, up to 896 KB RAM, and provides two levels of programmable function keys: a Command level, and a Program level that allows applications to use the keys without affecting the Command level functions. Unlimited characters can be assigned to each of the programmable function keys. "From what we've seen," Carroll concluded, "the performance looks outstanding. And, interestingly, the whole thing has been reviewed by a DEC Human Factors Committee, so that, for instance, all the screen have been designed for a consistent appearance. How many microcomputer software companies have a Human Factors Committee that pays attention to details like that?"

Distribution of Concurrent CP/M

For reasons which have not been publicly announced, DEC declined to market Rainbow CCPM. They instead returned it to Digital Research and asked them to distribute it. From several phone calls to Digital Research and DEC at the time, my impression was that Digital Research was not anxious to market it, and wanted to just license it to hardware OEMs, (like DEC) and let them sell it. After several months, Digital Research began distribution, but never advertised the product. I have never seen a review in a DEC magazine. The only way any DEC user could even know it existed was by the tiny print in an occasional mail-order house's ad.

Not surprisingly, sales were slow. Digital Research announced in spring 1985 that they would discontinue the product. Iron-

ically this occurred just at the time that open-market prices of 256K RAMs dropped dramatically, allowing many more Rainbow users to buy the memory needed to utilize Concurrent CP/M.

In late May and early June, I made several calls to Digital Research to confirm the rumor that they were discontinuing distribution. Several messages were never returned; their current hot new product is GEM, and Concurrent CP/M (for any machine) is low priority. On 12 June, Judy Merbis, Director of Corporate Communications, called. She verified that DRI had decided to drop the product. She explained that Digital Research incurs costs in listing and maintaining a product, that each separate disk format requires its own product codes, and that sales were not adequate to justify those costs.

I mentioned software houses I knew who wanted to continue selling the product, and asked if DRI would consider licensing it. She suggested that anyone who wanted to make such an arrangement contact Mike Lostus, whose title is something like, "Manager for Operating Systems Marketing". Address: Digital Research, Inc., P. O. Box DRI, Monterey, CA 93942, (408) 649-3896.

Conclusion

Rainbow Concurrent CP/M is an excellent product which combines high functionality with continued use of existing software. It is possible that a small company might license it from Digital Research, and could make a profit by operating at lower overhead. Alternatively, it is possible that DEC would decide to pick up the product to provide continuity in the software support for their hardware. Unless DEC decides to port Concurrent DDS to the Rainbow, Concurrent CP/M provides functions needed by some users which are not provided by any other Rainbow product.

Rainbow owners who would like to buy Concurrent CP/M will still (as of June 85) find it in stock at some mail order houses. In the interest of continued availability, I encourage interested users to contact both DEC and potential small distributors and express their interest.

Willett Kempton is a university researcher and DEC user. This article was taken from the VAXsig on CompuServe, and was previously published in the San Francisco Bay Area DEC PC User Group News.

More on the CCP/M situation:

From the CompuServe VAXsig: I'm so bummed out that CCP/M appears to be virtually unavailable that I'll be writing an angry "letter to MA" to gripe about DEC and CCP/M; I might even take it as far as an article for DEC Professional on the great lost operating system, or something! Also, what's all this BS about being unable to use GSX on more than one CCP/M console? (Robert Trelease)

I am relaying to this SIG a message that I just saw posted on the Digital Research SIG: "I talked with our manufacturing department a few weeks ago and discovered that we had just filled an order for 1000 DEC CCP/Ms! This order went directly to DEC, so I guess it will soon be available from them. The only support we will provide for it is through this SIG via other users and our "off-the-top-of our head" knowledge of the product. If DEC is going to sell the product, I assume they will have some sort of support set up for it. (a DRI employee)

A knowledgeable source at DEC's Peripheral and Supply group who told me that the large order for CCP/M was for purposes of a big Rainbow PC deal to a corporate user who will be using CCP/M as part of a custom application. However, if there is enough interest, my source noted, DEC could probably be persuaded to include CCP/M in its DDS (Digital Distributed Software) program. This is the stuff DEC will not warrant or support. (Will Roberts)

All of the above was published previously in the San Francisco DEC PC User Group News.

¹ This section is excerpted from a Digital Research Publication, "Digital Research ISV Forum", Vol. 3, No. 4, November, 1983.

=====
GRAFTALK: A FEW COMMENTS
by Lisa Heinz

Graftalk has lots of handy preformatted settings for common graphics needs. For example, when doing graphs, it already knows the days of the week, months of the year, and years; all you have to do is give it a starting point and an increment; for instance, [months Jan 1] will generate Jan, March, May. . . along the specified axis. Graftalk automatically sets the values of the x and y axes by the range of the data, unless told otherwise. It automatically changes line types, colors, or shading types when you plot several sets of data on one graph. And more. It is fast and flexible, but if you want it to do anything except pie, bar, and line graphs, you're on your own. It has an interactive/joystick mode, so any graphic can be produced; it is just difficult and timeconsuming. Graphics created in interactive/joystick mode are a pain to edit, and nearly impossible for anyone other than the person who created the graphic in the first place, or who hasn't spent many hours in interactive mode. But Graftalk is easy, fast, and fun for bar, pie, and line graphs, and for very simple interactive graphics.

=====
COMMENTS: TEKTRONICS EMULATION AND SPEECH MASTER
by Francis Shepard

We have used software which allows the Rainbow to emulate a Tektronics 4010 series terminal (with communications) and have downloaded drawings from a PRIME CPU via 1200 baud lines. The software allows windowing and expansion of the window to any degree onto the screen. The drawing must be subsequently downloaded onto a disk to be printed (we use an LA50). Unfortunately the window cannot be printed. However, we since the Rainbow emulates the Tektronics terminal, we believe that we can create a window at the mainframe and download it. We haven't tried it yet. The software works very well and the documentation is good. It is called GTM-10 and is available for \$ 1 from Machine Algorithms, 1515 Ninth Street, Boulder, CO, 80302, (303) 442-1072.

We have also used Speech Master, software for making view-graphs, on an LA50 printer. the documentation is terrible--read appendix D first. We found several ways to hang up the Rainbow so that only a cold boot [turning the computer off and then restarting it instead of using set up/control set up] will do. It does make nicely lettered printouts which we have used as report covers. The cost is \$ plus \$ shipping. As mentioned in the January, 1985 newsletter, the address is Graphic M*IXS, Inc., 960 Clocktower Drive, Springfield, IL 62074, (217) 793-0884.

Francis Shepard is a member of the Washington Area Rainbow User's Group.

=====
PUBLIC DOMAIN GAME REVIEW: HACK--EXPLORING THE DUNGEONS OF DOOM
By Rick McClinton

Hack is a screen-oriented Dungeons and Dragons style game similar to Rogue on the Unix/Ultix systems. You, the brave adventurer, set out into the dungeons to slay evil monsters and gain gold, useful items, and magic items both baneful and beneficial. Your distant goal is to descend twenty or thirty levels into the dungeon, find the amulet, and return to the surface again. Along the way, you and your little dog, who can be trained to do all sorts of useful things, will meet up with a wide array of monsters, traps, obstacles, and objects. Monsters like the Floating Eye and Quivering Blob will challenge you to battle. Traps

such as the pit and the teleport will hinder your movements. Secret doors and passageways block you from special areas. Sometimes you will run across a store where you can buy anything for any price (as long as it is very high). But don't give p hope, there is still a chance you will win! (Or so says Two Bacca, the bookie. . .)

Hack is played on your DEC Rainbow, using ANSI escape sequences so you don't need the graphics option. It doesn't use VT100 special characters, either. Every symbol used in Hack is accessible to you through the keyboard just by turning on the machine. Hack has on-line help for both instructions and command lists. It also has a feature which allows you to identify any object with a single command.

Hack is available through the Wash-A-Rug FIDO bulletin board and will be available from the public domain library. Getting it is worth the disk and the trouble!

Rick McClinton is a high school student who, when he wrote this review, had reached level 10 in Hack. He is also a member of the Washington Area Rainbow User's Group.

=====

THE FINE POINTS OF MS-DOS AND CP/M

USING MS-DOS BATCH FILE COMMANDS by Steven Stewart

All of the commands discussed here are documented in the user's guides that come with your operating system. However, the user's guides do not provide many examples of how to combine the commands to create a useful batch file. The example batch files presented in this article, or your own variations of them, will not only provide examples of how several commands work, but also may be of use in your MS-DOS applications.

A batch file, in its simplest and most commonly used form, is merely a series of MS-DOS commands saved in a file with a filename extension of .BAT. Once such a file has been saved you can execute it from the usual MS-DOS command prompt by entering only the filename (without the .BAT extension).

A Simple Menu System

The first example of batch file commands will be a simple menu system. This is an excellent use of batch files, and can be implemented very quickly and easily. Such a system makes it easier for several people in an office to use a single computer. In the following example, one batch file displays a menu on the screen which lists the programs that are available on the computer. When the user chooses which program to run, another batch file automatically switches the user to the proper subdirectory or prompts the user to insert the proper diskette, and executes the necessary commands to run the program. Thus each user does not need to know the details of where the programs are kept or what the syntax is for any commands.

(Note: If you want to enter the following example batch files and try them out you can use the EDLIN program that comes with the MS-DOS operating system. If you use your own word processor, be sure to save your batch files as MS-DOS text files instead of saving them in the word-processor format. For example, in WordPerfect save or retrieve your files as DOS text files rather than using the Save or Retrieve keys. If you use WordStar, open your file as a non-document file. Many simple editors, such as program editors, save the file in the proper format automatically.)

The following is a batch file named MENU.BAT which displays a menu.

```
1: ECHO OFF
2: REM MENU.BAT: DISPLAY MENU.
3: CLS
4: TYPE MENU.SCR
5: PROMPT --$g
```

Note that the line numbers shown above are for reference only. If you try this batch file on your own computer, do not key in the line numbers or the colons immediately following them. To run the batch file you just type MENU and a carriage return at the MS-DOS command prompt.

The first line, ECHO OFF, is the first line of many batch files. The ECHO command may take any of three parameters. If the parameter is OFF, all subsequent commands in the batch file will not be displayed on the screen. The results or output of the commands will be shown, but not the command lines themselves. If the parameter is ON (the default) subsequent command lines will be displayed. If the parameter is neither OFF nor ON then the rest of the line following the word ECHO is taken to be a text string which is displayed on the screen. ECHO is often turned OFF at the beginning of a batch file simply to control the appearance of the screen while the batch file is executing.

The second line of MENU.BAT is a comment to help document the batch file. Comments begin with REM (for REMark). While it is good to document your batch files to some extent, you will find that too many comments can significantly decrease the speed with which your batch files execute.

The third line of MENU.BAT, CLS, is an MS-DOS command to clear the screen. The cursor is automatically placed in the upper-right corner of the screen.

The fourth line uses the MS-DOS TYPE command to display a file, MENU.SCR, which contains the menu. You must create this file to look just as you want it to appear on the screen. (You could also use the ECHO command repeatedly to list the menu to the screen one line at a time, but the TYPE command will display a screen full of information much more quickly than a series of ECHO commands.) MENU.SCR might show that option 1 is to enter BASIC, option 2 is to enter your word processing program, option 3 is to enter your spreadsheet, and so forth. The last line of the MENU.SCR file might say something like "Enter the number for your choice." If you know how, you might even give your menu screen a professional look by inserting the escape sequences in your MENU.SCR file to create double-width characters for the menu title, put the numbers in bold or reverse video, and use the special graphics character set to draw a box around the menu.

The last line of MENU.BAT changes the usual MS-DOS system prompt. By using the PROMPT command you can make the system prompt appear to be just about anything you want, including the time, date, default directory, or any text. The MS-DOS Advanced User's Guide lists the characters to use in the PROMPT command to display special prompts. The prompt shown in the example is a simple right arrow, "-->", which will be displayed instead of the usual default drive, such as "A>". This will provide the impression to the user that the computer is waiting for a response to the menu instead of an ordinary MS-DOS command. At any time you can get back the default MS-DOS system prompt by typing PROMPT followed by no parameters. This command might also be made into a "Return to MS-DOS" option in your menu.

Once this file is displayed, the user will see the cursor at the "-->" prompt and may enter any valid MS-DOS prompt. To respond to the menu, the user will enter the number of the program to run followed by a carriage return. Thus, you will also need a separate batch file for each choice in the menu. The batch files will be named 1.BAT, 2.BAT, and so on. Here is an example of a batch file named 3.BAT which enters Lotus 1-2-3.

```
1: ECHO OFF
2: REM 3.BAT: ENTER 1-2-3.
3: CLS
4: ECHO Put Lotus system disk in drive A.
5: PAUSE
6: CHDIR \LOTUS
7: 123
8: CHDIR \
9: MENU
```

The first three commands in the 3.BAT file are similar to those already discussed in the MENU.BAT file. The fourth line is an example of an ECHO command to type a message to the screen. Since 1-2-3 is a copy protected program, you must place the Lotus system diskette in the A drive even when the actual program is run from your hard disk. Line 4 tells the user to do so.

The fifth line is a special batch command which interrupts processing of the batch program until the user presses any key on the keyboard. The message "Strike a key when ready . . ." is automatically displayed on the screen. Presumably, the user will then insert the Lotus system disk in drive A and strike a key according to the instructions on the screen. Alternatively, the user might type control-C at this point to discontinue processing of the batch file.

Line 6 changes the default directory to the LOTUS subdirectory where the program is kept. Line 7 executes 1-2-3. After the

user exits from 1-2-3, control is returned to the batch file. Line 8 then changes the default directory back to the root directory or to whatever directory that the menu system batch files are kept.

The last command of 3.BAT is MENU. This executes the MENU.BAT file, which once again displays the menu on the screen. It is important to know that, although one batch file may pass control to another in this way, control will not automatically be returned to the first batch file once the second batch file is finished. MS-DOS keeps track of processing only one batch file at a time and does not allow the nesting of batch files. In the above case it makes no difference since the MENU command is the last command of the batch file.

You might also make the MENU command the last line of your AUTOEXEC.BAT file. An AUTOEXEC.BAT file, if it exists in your root directory, is automatically executed immediately after MS-DOS is first loaded. Making MENU the last command in this file will cause the menu to be displayed as soon as you first enter MS-DOS. Incidentally, you should also put the DATE and TIME commands in your AUTOEXEC.BAT file so that you will still be prompted for the date and time to set your system clock/calendar.

It should be easy for you to think of ways to enhance the above menu system to suit your own needs. For example, some menu choices might execute other batch files which lead to submenus. Others might execute commonly performed MS-DOS functions such as listing the files in a particular subdirectory or backing up certain files to a diskette.

Copying Without Fear

The example I will use to illustrate more advanced batch commands may also help to solve a problem you may have come across if you have a hard disk on your system. If you have a diskette-based system you still may find applications for this kind of batch file. Let's say you have a subdirectory on your hard disk in which you keep files pertaining to a particular project, say, the XYZ project. Like all careful computer users, you regularly back up the files on your hard disk onto diskettes. This is easily done from the hard disk subdirectory by using the command "COPY E:*.* A:". (We will ignore the possibility of using the BACKUP command for this example.) Usually if there are already files on the backup diskette in A:, you do not mind overwriting them with the updated versions that are on your hard disk. The COPY command will overwrite the old files automatically. Now, some time has passed, you have backed up the files on your subdirectory several times, and you have erased some of the XYZ project files from your hard disk to save space. But now the XYZ project is starting up again and you need to retrieve all the XYZ project files from your backup diskette. You could use the command "COPY A:*.* E:" from your hard disk subdirectory to retrieve all the files, but by using this command you might overwrite some files on your hard disk with older versions on the backup diskette. Wouldn't it be nice if MS-DOS had a command that only copied files that do not already exist on the destination drive?

MS-DOS does not have such a command, but you can create your own command to do this with the following batch file. I call the file RETRIEVE.BAT.

```

1: ECHO OFF
2: REM RETRIEVE.BAT: GET BACKUP FILES WITHOUT OVERWRITE.
3: A:
4: :TEST
5: IF '%1' == '' GOTO END
6: FOR %%F IN (%1) DO IF EXIST E:%%F GOTO WARNING
7: GOTO COPY
8: :WARNING
9: ECHO Note: The following files are on both
10: ECHO A: and E: and will not be copied:
11: FOR %%F IN (%1) DO IF EXIST E:%%F ECHO %%F
12: :COPY
13: ECHO Retrieving %1 . . .

```

```

14: FOR %%F IN (%1) DO IF NOT EXIST E:%%F COPY A:%%F

```

```

E:
15: SHIFT
16: GOTO TEST
17: :END
18: E:
19: ECHO Retrieval complete.

```

As I mentioned above, the line numbers and colons immediately following them are for reference only and should not be keyed into your batch file. However, the colons that immediately precede the statements in lines 4, 8, 12, and 17 are part of the statements and should be keyed in.

Many commands in the RETRIEVE.BAT file should be familiar from the last example. Line 1 tells MS-DOS not to display the commands in the batch file as they are being executed. Line 2 is merely a comment to document the batch file. Line 3 makes A: the default drive. This is necessary for some of the commands in the batch file to work properly. Near the end of the batch file, on line 18, the default drive is made E: again.

Line 4, ":TEST" is the first unusual looking line. This line is called a label. Its purpose is to act as a reference point so that you may go directly to that line from somewhere else in the batch file. Notice line 16, "GOTO TEST", near the end of the batch file. This GOTO statement, when it is eventually reached, will cause execution to continue on the line immediately following the line labeled :TEST. Thus, most of this batch file is a large loop, being repeated over and over again. (The reason for this will become clear shortly.)

In general, a label is any line in which column 1 contains a colon. Immediately following (no space) the colon is a string of characters which identifies the label. Any other words on the line are regarded as a comment. In the above batch file, lines 4, 8, 12, and 17 are examples of labels. While the initial colon is a necessary part of the label statement, in the GOTO statement the colon should not be typed as part of the label name.

Line 5 is an example of an IF command. There are several versions of the IF command. The one on line 5 simply tests for the equality of two character strings. The format of this kind of IF statement is "IF (string1) == (string2) (command)". In RETRIEVE.BAT, if the string '%1' is exactly the same as the string '' (these strings will be explained next), then the GOTO statement on the IF command line is executed and control is passed to the statement following the label :END (line 17). If the two strings are not equivalent, then the GOTO statement is not executed, and the batch program continues with the statement following the IF command, line 6. In general, the command appearing after the string equivalency test on the IF command line need not be a GOTO statement. Any valid MS-DOS command is possible.

So, how can the string '%1' ever be equivalent to the string '' on line 5? In a batch file, any single digit preceded by a percent sign stands for a parameter originally on the command line that you entered to execute the batch file. Thus, %1 stands for the first parameter, %2 for the second, and so on to %9 standing for the ninth parameter. The %0 parameter stands for the batch command name itself. Presumably, when you execute this batch file you will enter a line something like "RETRIEVE XYZ*.X" to retrieve all of the XYZ project files from the diskette in drive A: to the current default directory on the E: drive. In this example, the XYZ*.X on the command line would be the first parameter. When the IF statement on line 5 is executed, the %1 is replaced by XYZ*.X. Thus, the statement will be interpreted as "IF 'XYZ*.X' == '' GOTO END" and, of course, the GOTO statement will not be executed since the two strings are not equivalent. If you had not included any parameter after the RETRIEVE command, then the statement would have been interpreted as "IF '' == '' GOTO END" and since the two strings are equivalent, the GOTO statement would transfer execution of the batch file to line 18. This is exactly what we want, since if the user of the batch file specifies no

files to copy, then we can skip the copy process and end the batch file.

Incidentally, in an IF statement it is not absolutely necessary to enclose the strings to be compared in single quotes. The statement is actually comparing any two character strings delimited by spaces. Thus, you might have statements like "IF %1 == C GOTO STEPC". This way of coding the IF statement is perfectly valid as long as %1 is not a null string. That is, if the user does not enter any parameters on the command line, then the above IF statement would be interpreted, after substitution of a null %1, as "IF == C GOTO STEPC". This is an invalid statement, since there is no string specified before the == sign. If this happens, then the user would be warned of a syntax error when the batch file is run. To avoid this possibility, I usually append some extra character to both strings to make sure neither string is evaluated as null. Enclosing the strings in single quotes, as in RETRIEVE.BAT, serves this purpose.

Another important type of IF statement tests for the existence of a file. The format of this kind of IF statement is "IF EXIST <filename> <command>". (The word EXIST in this command must be in upper case.) When this statement is executed, MS-DOS searches for the specified file in the current directory. The file specification may include the standard wildcard characters (* and ?), but it may not include a path name. If the file exists, the <command> is executed. Otherwise <command> is skipped. This type of IF statement is seen on lines 6, 11, and 14.

Before looking at the rest of the batch file it will be helpful to introduce the FOR statement. The FOR statement allows you to execute a single command several times in succession, changing the command in some way each time. The general format of the FOR statement is "FOR %%c IN (<set>) DO <command>". The words FOR, IN, and DO must be upper case. The <set> is either a series of character strings separated by spaces or a file specification which may include wildcard characters (although the file specification may not include a path name). The <set> must be enclosed in parentheses. The letter c following the two percent signs can be any single character except a numeric digit. The %%c represents a variable which changes value each time the FOR statement executes the <command>. The FOR command is executed as follows. For each member of <set>, make %%c equal to the <set> member and then perform the <command>. Usually, the <command> will include the %%c so that the <command> will appear different in some way each time it is executed. An example of how to use the FOR statement is: "FOR %%F IN (TYPE PRINT ERASE) DO %%F MYFILE.DOC". This FOR statement will execute three commands in succession: "TYPE MYFILE.DOC", "PRINT MYFILE.DOC", and "ERASE MYFILE.DOC". Another example is: "FOR %%F IN (XYZ*.*) DO ECHO %%F". This command would search the current directory and display on the screen each filename which matches the file specification XYZ*.*. Consult your MS-DOS User's Guide for other examples and an explanation of why the double percent sign is necessary.

Line 6 is a pretty complicated statement. Let's divide it into pieces and look at them separately. The statement starts with "FOR %%F IN (%1) DO ...". Remember that when the RETRIEVE batch file is run, %1 will be a file specification. Since we switched to the A: drive in line 3, this FOR statement will search the A: diskette for each file matching your first file parameter on the RETRIEVE command line. For each matching file that it finds it will execute the command following the DO statement. What is this command? It is an IF statement, "IF EXIST E:%%F GOTO WARNING". Thus, for each matching file on the A: drive, the IF statement will test whether the file also exists on the default directory of the E: drive. If it does, the "GOTO WARNING" statement is executed which passes control to a part of the batch file that displays a warning message. If the same files are not found on both the A: and E: drives, then it is safe to perform the file copy. In this case the "GOTO WARNING" statement on line 6 will never be executed, but instead the "GOTO COPY" statement on line 7 will be executed.

The warning section of the batch file, lines 8 through 11, echo a warning to the user that certain files exist on both the A: and E: drives and will not be copied. Line 11 then displays the name of each such file on the screen. The FOR statement in line 11 is almost exactly like the FOR statement in line 6 which was discussed in detail above. The only difference is that for each file that is both on the A: and E: drives, an ECHO statement is performed instead of a GOTO statement.

The copy section of the batch file, lines 12 through 14, copies from A: to E: only the files that are not already on E:. Again, the FOR statement in line 14 which performs the copy is almost exactly like the FOR statement in lines 6 and 11. This line shows a variation of the IF statement that has not yet been discussed. It is possible to precede the conditional part of the IF statement with the word NOT (must be upper case). The entire FOR statement may be interpreted as follows: "For each file on the A: drive matching the file specification listed as the first parameter on the batch file command line, if the file is not also on the E: drive then copy the file from A: to E:". This one statement does all of the important work of the batch file. In fact, most of the other statements in the batch file are merely enhancements and could be deleted.

The SHIFT statement on line 15 makes the old %2 equal to %1, the old %3 equal to %2, and so forth. Thus, if the user listed more than one file specification to copy on the RETRIEVE command line, the second file specification now becomes %1. Line 16 returns control to :TEST at the beginning of the batch file, and the whole process repeats itself. The test in line 5 will cause the batch file to jump out of this loop as soon as a null parameter is encountered, which means that the user listed no more file specifications on the command line.

Whereas most of the batch files that are commonly written perform a single very specific task, the above example shows how you can create flexible, general purpose commands of your own that can execute differently depending on the parameters you supply. You can probably think of several ways you might enhance the above batch file to suit your own needs a little better. You might also want to create something like an ARCHIVE batch file which backs up selected files to diskette.

Steven Stewart is a Software Engineer with Computer Science Corporation.

(c) 1985 by Steven Stewart

=====

SCREEN CONTROLS FROM CP/M OPERATING SYSTEM
by N. Jay Bassin

We all know that the Rainbow screen can do some pretty nifty things with graphics, bold (high-intensity), underlining, double height/width, direct cursor addressing, etc. If you use a printer with "escape" sequences to change font, spacing, or other attributes, you also know that those features may be toggled by appropriate software (e.g., BASIC or some word processing package or such). It is possible, however, to generate these screen and printer "escape" sequences directly from the CP/M operating system prompt! For example, the VT100 sequence for "clear screen and home cursor" is "<ESC>[2J<ESC>[H". In BASIC, this command would be expressed as "CHR\$(27)+"[2J+CHR\$(27)+"[H". However, if you tried to execute the command from the system prompt, you would get a "beep" and a "?". The trick is to first type a space at the prompt, then the escape sequence using the <CTRL> key thus:

A) <CTRL><ESC>[2J<CTRL><ESC>[H <return>

(the " " signifies a space, not an underline.). If you key this in, you will see echoed on the screen: A) ^[[2J^[[H. The "^[" is the "key" form of the non-printable <ESC>. To exit from the

Rainbow's graphics mode to the standard USA ASCII character set, type <CTRL><ESC>(B" followed by a return. For my printer, I have to send a "decimal-29" (ASCII 65) to toggle 17-cpi printing and an "ESC 1" to toggle letter-quality printing. To generate an "ESC 1" to the printer from the operating system, first type a "CTRL P" (^P) to echo all commands to the printer. Then type a space, <CTRL><ESC>1<CR>. It will appear as "^1". (Don't forget to type another ^P to turn off echoing to the printer.) However, how do you generate a "decimal 28," which is a non-printable character and which has no apparent key? All of the first 32 ASCII characters, non-printing though they may be, are "emulated" by a particular escape-character. Some of the more complete ASCII character code charts contain these keys. (If yours doesn't, you can figure it out easily as follows: add 64 to the decimal equivalent to the character (found in any table), and find the ASCII expression of that value. For example, decimal "7" beeps the keyboard. Add 64+7 = 71, which converts to "G." If you type a <CTRL>G, you'll get a "beep.") OK, following that rule, we find that decimal 28+64 = 92, which is the ASCII "\" (backslash). When I want to toggle 17-cpi on my printer from the operating system, all I need do is type a "space<CTRL>\\" and it would show up on the screen as "\". As soon as I hit <RETURN>, I can hear the printer switch over. You can do any valid escape sequence this way, and even concatenate sequences if you don't put in extra spaces between the expressions (such as the "clear screen, home cursor" commands above).

=====

DOS-FORMATTING A BLANK DISKETTE
by N. Jay Bassin

Contrary to the instructions in the MS-DOS manual and in the FORMAT command instructions, it is possible to format a fresh disk without first running the CP/M format command. The trick is to use the undocumented "initialization switch" ("/I") with the command. Thus, at the DOS prompt, type:

A)FORMAT B:/I <CR>

If you want to transfer the DOS system files (and COMMAND.COM) to the new diskette to make it "bootable", use the "/S" switch with FORMAT in the same manner.

(c) 1985 by N. Jay Bassin

=====

TECHNICAL ARTICLES

=====

ADDING MEMORY: IT'S EASIER AND MORE INEXPENSIVE THAN EVER
by Tom Tugman

New software programs call for more and more memory. DESQ, for instance, can require over 512K of memory in order to operate in conjunction with other programs. Lotus will gladly utilize more memory, if it has the opportunity. And more memory allows you to make a larger RAM disk, which allows programs to operate much faster than they can on floppies.

A couple of meetings ago, one of the Washington Area Rainbow User Group's members, Bob Catt, mentioned that it was possible to get 150 nanosecond 256K chips for as little as \$1 each (it takes 9 separate chips to make 256K), from a company in Beggs, Oklahoma. DEC offers 9 chips (200 nanoseconds) for a bargain price of \$ (the same price as the 256K memory board).

We had purchased the 256K board when we originally bought our Rainbow 100+, but couldn't afford to "populate" the board to the maximum allowed: 896K. (Back then the board was \$1000).

Although I was skeptical--I had never heard of Microprocessors Unlimited [24000 South Peoria Avenue, Beggs, OK 74421, (918) 267-4961], and we have had bad luck with mail order firms before--I finally decided to order 512K worth of chips. By the time we got around to calling, the chips had dropped substantially in price. The day we ordered them, Friday April 12th, Microprocessors Unlimited was actually selling them for \$1 each! [As of the end of July, they are less than four dollars each.] The operator at the company informed us that the price changes frequently, depending on the supply and price in Japan. The two sets, or 18 chips, came to \$18. They shipped them Federal Express for \$6.00, making the total price for 512K only \$24. Microprocessors Unlimited took a credit card number, but agreed that we could send them a check when we received the chips. We received them the next Monday, April 15th.

Installation was easy. Microprocessors Unlimited sent along a bizarre several page long set of instructions, but this is all you really need to do:

1. The chips arrive in boxes wrapped in aluminum foil. Don't open the foil until you're ready to install the chips.
2. This affair is casual dress only. Be sure to wear cotton clothes (no nylon or other synthetics) and no shoes. The static charge generated by leather on a wool or synthetic rug can destroy the chips (not to mention the memory board!).
3. Put a towel down on the table as a cushion, then spread out a long sheet of aluminum foil.
4. Completely unplug your Rainbow from the wall.
5. Unplug the peripheral interface cords from the back of the Rainbow. Remove the system unit cover.
6. Unscrew the back and remove the circuit boards. You will have to unplug all of the internal ribbon cables for the power source, floppy and hard disk, etc.
7. Remove the Memory Extension board. Lay it down on the aluminum foil, chip side up.
8. Open the box(es) of chips. Touch the aluminum foil with your fingers before removing the first chip and plug it into one of the vacant sockets in the memory board, making sure that the notch is in the same direction as the chips already in the board. Make sure that the tines are correctly aligned by pushing them in a little on

each side first, before pressing the chip down. Don't press too hard on the chip or the memory board.

9. After you have installed all of the chips, check the dip switches which are located to the left of the memory banks. If you have installed 256K chips, all of the dip switches should be up. (Switch 4, on the right, is always up.)

10. Put the Rainbow back together. If you have a 100A, you will need to go into SET-UP and manually reset the amount of memory; on the 100+, it will reset automatically. Run Test 1 on the Rainbow diagnostics disk. (If the test fails, open the system unit and check to see if any of the chips are loose.) If it passes, you're finished!

Note: Since this article was first published in the Washington Area Rainbow User's Group Newsletter, I have had a number of calls from Rainbow users who want to know if there is a way to avoid buying a 256K memory board (PC1XX-AD/LZ) from DEC (\$699) in order to upgrade memory [or, in the case of a Rainbow 100A, buying the 256K memory board and the DEC memory adaptor board (PC1XX-AK/LZ, \$99)]. One alternative is to buy a 128K board from DEC (PC1XX-AC/LZ), which is only \$300. The 128K board, which is physically the same as the 256K board, has 2 banks of 64K chips and one empty bank. You can remove the 64K chips and fill all three banks with 256K chips. Use of the 128K board saves approximately \$350, including the cost of a third set of 256K chips. The only third party boards I know of, by Univation, go to a maximum of 448K. If any readers have a better answer, please contact me or the newsletter editor at (301) 927-0108.

FOLLOW UP: AN ADDENDUM ADDING MEMORY CHIPS

Dear Tom,

As a follow-up of our conversation regarding adding 256K chips to the Univation board(s), Univation stated that the boards were not designed to accept the additional capacity, even though the 256K chips would fit in the slots. As a further hindrance to adding to the board without Univation's help, is the lack of dip switches on the board. This function is performed by something Univation calls "memory mapping PAL" which apparently must be furnished by Univation.

Univation stated that they are working on a redesign of the board to accept 256K chips. They were asked to come up with an attractive trade-in arrangement. The current 192K board has 27 slots; filled with 256K chips it would produce 768K.

I would like to mention that Univation repaired my board in one day. That is truly good service! My system was down for less than one week total. I really owe them a vote of thanks.

Bob Saul
Covington, Louisiana

(Tom Tugman is a systems analyst with General Electric. He is Treasurer of the Washington Area Rainbow User's Group. He also furnishes technical advice to the editor of the WARUG Newsletter. This article originally appeared in the WARUG Newsletter.)

(c) 1985 by Thomas D. Tugman

RAINBOW EXPERIENCES PROBLEMS WITH FLOPPY DISK REINFORCING HUBS
by Arthur Wouk (via ARPANET)

I have recently had the following experience with my DEC Rainbow. I received the Rainbow in March, 1984, with two floppy drive units (four disks) and began using Verbatim Datatype floppies at that time. I started using Dyan floppies in September, when

our purchasing department switched to them.

About two weeks ago, I began to have trouble reading from Drive D; later the trouble appeared on Drive A. I tried to format a new box of Dyan disks, but was unable to format them on any combination of drives, one for the X program, the other for the disk to be formatted. At this point I called DEC Support. . . . The repairman came, and he tried the usual. Replaced all possible electronics to no avail. He could get DEC floppies to format from DEC furnished CP/M sources. In all cases CP/M reported that the drives were not in acceptable speed range. Replacement of all four drives (two disk units) made DEC floppies work, but did not permit formatting the Dyan disks.

The word from the DEC phone backup support people is the following: Rainbow drives are intended to be used with floppies WITHOUT reinforcement rings at the central hole. The presence of these rings damages the drives, while the drives damage the rings, to the extent that eventually BOTH are unusable. You must use non-reinforced floppies. There is no statement of this in any DEC literature, but the software which is DEC-supplied is always on unreinforced disks.

Apparently this problem has just begun to surface. Change your floppies as soon as possible to unreinforced, or expect trouble with the drives eventually. With my new drives, I am able to read the VERBATIM disks, but not the 10 or so DYAN disks. I am puzzling over methods to recover the information on the latter disks.

An addendum to my note, "Warning to DEC Rainbow Users." Several floppy manufacturers (Memorex, 3M, Immac, . . .) now sell a disk for the Rainbow with special format. The same disks are also to be used with the DECmate II (not the DECmate I) and the PRO 350 series. Apparently DEC changed their drives about the same time the Rainbow was introduced, without telling anyone.

[Editor's Note: The Digital Hotline hardware people explained that there are two parts to this problem. First, the reinforced hubs are more liable to slip, which can make them impossible to read. Second, the reinforced hubs can cause the plastic fingers which grab the disk while it is in use to be damaged, particularly if the hub is off center. On the other hand, if you use non-reinforced hub disks, they tend to wear out very quickly. The Hotline representative said that this should not happen, and if your drives are trashing your disks, you need a service call. There are tell-tale signs that this may soon occur: look for tiny scratches on the center rings of the disk. There should be none. The primary problem with continuing to use disks with reinforced hubs, other than destroying your disk drives, is that you may not be able to read the information off of them if they become damaged. Of course the same thing can happen to non-reinforced floppies. The best solution: keep at least two back-ups.]

SETTING PRINTER FEATURES
by Paul Vince

Many printers use control sequences set up by the American National Standards Institute (ANSI) to control features such as horizontal and vertical pitch, printing density, and bold and underline selection. These control sequences are also called escape sequences, since most begin with the ASCII "escape" character. A detailed list of your printer's features and the control or escape sequences required to invoke them should be found in the User's Manual.

There are a number of programs available which can set the printer with the desired features. Most word processing programs offer the user the ability to control printer features with internal editing controls (see "WordStar and the LA-50" below). Lotus 1-2-3 and Symphony offer the opportunity to enter a printer setup string (another term for a control sequence) before printing a worksheet. There is an excellent DEC program called "SETPOR",

available from the WARUG public domain library in both MS-DOS (vol. 28) and CP/M (vol.29) which allows you to set many features of the DEC LA-50, LA-100 and LQP102 (plus communication parameters). My vote for easiest to use with the LA-50 is an MS-DOS program called PCLASO.COM available in file area #12 of the Wash-a-Rug bulletin board, and on Volumes 40 and 57 of the WARUG Library.

However, if you don't have any of these handy programs, you can enter control sequences directly from the keyboard. As an example we will use the control sequence which causes the printer to use the compressed pitch, the 16.5 cpi (character per inch) pitch in which this newsletter is printed. The control sequence for the LA-50 (and presumably for other printers using ANSI control sequences) is [ESC][4w (Escape, left bracket, 4, lower case "w"). The lower case "w" MUST be used or the sequence will not work.

In CP/M, at the system prompt, type:

PIP LST:=CON: <Return>

<ESC>[4w <Return>

The monitor displays [4w and the cursor returns to the beginning of the line.

<Ctrl>-Z

returns you to the system prompt. The printer should now type in the compressed pitch.

In MS-DOS, at the system prompt, type:

COPY CON: PRN:

<Interrupt>[4w <Return>

The monitor displays ^[[4w and the cursor moves to the next line. Press

<Ctrl>-Z <Return>

You'll receive the message,

1 file(s) copied

and you will be returned to the system prompt.

If you wish to set other features, just type additional control sequences on separate lines before the <Ctrl>-Z which is the end-of-file message. You can use either upper or lower case for the PIP and COPY commands but the letters in the escape sequence MUST be in the case indicated in the manual.

The simplest way to reset the printer is to turn it off and back on but it can also be accomplished by repeating the above procedures and substituting either a 0 or a 1 instead of the 4 in the second line. If you plan to use an escape sequence often, use a CP/M editor such as RED or an MS-DOS editor such as EDLIN to name and create a file containing the escape sequence(s) followed by ^Z. Then, to set the printer use either PIP or COPY as shown above, replacing "CON:" with the filename. If you view these files after creating them, you should notice that both the <ESC> in CP/M and the <Interrupt> in MS-DOS are displayed as ^[[which is the symbol for Escape.

WordStar with the DEC LA-50 Printer

Both the CP/M (v 3.33) and MS-DOS (v 3.11) versions of WordStar offer the DEC LA-50 printer (among many others) as choices on the installation menu. Selecting the LA-50 installs some pitch selection features which are only documented in the files LA-50.TST (or LA50-100.TST) and LA50ENH.TST found on the distribution disk. The default pitch is 10 cpi (characters per inch), also known as pica pitch, which gives a line length of 80 characters. The dot

commands for setting horizontal pitch and lines height do not work for the LA-50 (at least in my hands). However, the command ^PA (Ctrl-PA) toggles the printer to 12 cpi, known as elite pitch, with 96 characters per line. The "P" is not displayed on the wordstar screen, only the carat and the second character (e.g. ^A) will be seen. Four "user patches" ^PQ, ^PR, ^PE and ^PW, toggle the printer for four additional pitches giving the user access to all 6 pitches of which the printer is capable. (The default pitch is ^PN. To return to the default pitch from ^PQ, ^PR, ^PE or ^PW, type ^PA^PN; to return to the default pitch from ^PA, type ^PN):

```
PRINT CONTROL
^PQ ^PA ^PN ^PR ^PE ^PW
```

```
PITCH (CPI)
16.5 12 10 8.25 6 5
```

```
CHAR PER LINE
132 96 80 66 48 40
```

Other features which control the appearance of the print are enhanced density (^PY), bold density (^PB), double striking (^PD) and underlining (^PS). Neither enhanced density nor bold printing will work with the 16.5 or 8.25 pitches. Bold and enhanced densities cannot be used together. Double strike printing (which closely resembles bold) and underlining will work with all pitches in regular or enhanced density.

Combining the six pitches with enhanced density, bold or double printing gives a total of 24 different print appearances on the LA-50. Although the characters appear normal on the screen, if you mix pitches it will probably be necessary to adjust the page offset to get the left margins to line up. You may also need to reset the right margins to accommodate the new line length (given in the Characters per Line list above). In order to get double strike printing to operate properly, it may be necessary to place the print command(s) above the line to be double printed. However, with a bit of experimentation you can quickly become familiar with the options, and enjoy an unexpected printing versatility with the LA-50.

The "user patches" (^PQ, ^PR, ^PE AND ^PW), also installed for when LA-100 is selected from the Printer Installation Menu and their functions, are documented in the files LA50-100.TST and LA-100ENH.TST on the distribution disk. These user patches are not installed for printers other than the LA-50 and LA-100. However, they can be installed for any other printer (or changed for the LA-50 and LA-100) by following the instructions for "User-Defined Functions" in the "Custom Printer Installation" section of the WordStar Installation Manual.

See pages at end of this newsletter for commands and samples of the print they produce.

Paul Vince is on the steering committee of WARUG.

=====

LOTUS 1-2-3 PRINTGRAPH ON THE RAINBOW
by N. Jay Bassin

Users of Lotus's 1-2-3 on the Rainbow know that the PrintGraph printer library is limited to DEC's LA-100 and LA-50 dot-matrix printers. Lotus users who also use other computers such as the IBM-PC also know that this limited printer library is unique to the Rainbow. For nearly two years I have moaned, complained, written polite letters, telephoned, and generally made a nuisance of myself at DEC and Lotus--to no avail. Not being a "techie," I fruitlessly tried a number of home remedies, and even tried to get some information on writing my own driver for my Okidata printer (I was desperate!).

Then, at the last WARUG meeting, Robert Perry of Lotus listened to my tale of woe and said he'd look into it. Having heard

that before from others at Lotus, I was pleasantly surprised when, within a week, he sent along a new Graphics Printer Library, with the suggestion that I try it. This disk, marked for the IBM and clones, Compaq, TI and Wang computers, contained their recently announced upgrade, adding twenty more drivers to the already-large IBM device menu. Following the enclosed instructions, I copied the single file "LOTUS.DLB" (for "device library") onto my working copy of PrintGraph. Even though the disk was IBM formatted, it was single-sided so the Rainbow read it easily.

I quickly generated and "saved" a Lotus graph (not having a DEC printer, I didn't have any ".PIC" files on hand) and called up the device selection menu from PrintGraph. Adrenaline pumping, I saw my Okidata printer listed for the first time! Oh frabjous day!

I suddenly became the only Rainbow 1-2-3 user who could print graphs on a non-DEC printer. Why couldn't Lotus have told me this a year and a half ago? Any IBM version of 1-2-3 could have supplied LOTUS.DLB, and saved us both a lot of aggravation.

Rob Perry should be the president of Lotus!

Now back to earth, I am providing a copy of Lotus' new Graphics Printer Library to the WARUG library. The documentation included with Lotus' distribution disk states in part, "We hope that you will use the diskette we are supplying you with to update your own PrintGraph Disk and those of other 1-2-3 users in your company. Please feel free to copy the disk for other people in your company..." There is no indication of restricted use.

There is one caveat: this device library does not contain the LA-50 driver, only the LA-100. If you only use an LA-50, this will not help. If you use an LA-50 plus other printers and plotters, you will have to have two separate PrintGraph disks. If you use non-DEC printers exclusively, this is a godsend. [Postscript: On May 31 I got a call from a technician at Lotus, responding to one of my many previous letters. He told me that the Rainbow would only interface with DEC printers (false), and that the Rainbow version of 1-2-3 was only supplied for DEC printers. I cheerfully told him that a local Lotus person had solved the problem, and I explained how. I suggested that Lotus notify other users of this fix, but he said that they couldn't do that.]

=====

COMMUNICATIONS TIP
by Steve Jack

Using a communications program can be fun, especially when the software is free! But there can be some problems. For example, I use LCTERM and a Paradyne 2400 baud modem. For about 6 months I dialed just about every FIDO across the U.S. and I have the phone bill to prove it. I had very few problems until I started dialing the local FIDO. I noticed that some part of the pictures or text would 'fall off' the right side of the screen. I tried changing modems, I tried new software, but no change. I could not figure out what on earth was going on. I even asked some of the members of the LUG (Pacific Rainbow, Covina, CA) but no one could help. I finally got fed up and borrowed a data scope from my office (a data scope is like a CRT except that it won't act on any control characters; it only displays the data). I hooked the data scope up between the modem and the Rainbow and noticed that every time an "HT" character came across the data scope's screen, that segment of data was lost on my Rainbow's screen. (HT is a screen control character and stands for horizontal tab). As soon as I saw that, I went into set-up and checked for tabs. There were none! I put the standard tabs settings back in and that solved the problems. I hope this saves someone out there the weeks of aggravation that I went through. Have Fun!

Steve Jack is Co-Chairman of the Rainbow Pacific LUG in Covina, CA.

=====

=====

USING A FIDO BULLETIN BOARD--NEARLY FOOLPROOF DIRECTIONS
By Caroline Mack

(To those of you who have never done this before: if I can do it, you can do it! Follow the directions and actually download some software!!)

In order to use bulletin boards, you will need to have a modem--just about any will do. You will also need software which allows your computer to communicate with the bulletin board. You don't need to spend a fortune on Crosstalk or PolyCom. The directions which follow are for DECMINI, a program which is available in the WARUG Public Domain Library. If you don't already have it, order Volume 34. (You could also download it from a FIDO bulletin board, but if you know how to do that, this article isn't for you.)

Once you have your modem and DECMINI, you are just about ready to go. Boot MS-DOS. Go into Set-Up, and set the following (some will be the default setting from the factory, and you need not change them if they are the same):

1. Under the statement of memory, there is another line which will say, LINE or LOCAL. Make sure it says LINE:

```
xxxK
LINE
```

2. Use the <Next Screen> key to move to PARAM SET. Use the right arrow to move across the parameters. Use the up and down arrows to change the parameters.

Parameter	Set # to
Emulation (Ansi)	1
Auto X-On/X-Off (on)	1
Local Echo (on)	1
XMT Break (enable)	1
Modem Stop Bits (1)	0
RCV Char Parity (check)	1
Auto Ansbk (enable)	1

3. While you are still in Set-Up, use the <Next Screen> Key to get Modem. Using the arrow keys, set Modem as follows (using 1200 baud or 300 baud, depending on your modem):

```
Modem
8N = Data/BP
300/1200 = XMT Baud
300/1200 = RCV Baud
FDXA = Protocol
```

4. To save your settings, type <Shift S> while you are still in Set-Up mode. (If you don't want to save them, don't do anything. They will disappear automatically when you turn the machine off.)

5. Get out of Set-Up, and type in

```
DECMINI
```

6. Turn on your modem.
7. Dial a bulletin board number (you'll find a list of them below, near the end).
8. Press <Return>. You will be asked to enter your last name, first name, and password.
9. Be sure to write down the password you choose (don't choose

your name for a password).

10. On some boards, you will be asked to fill out a questionnaire. Do it. On your initial call, you may be "restricted" in your ability to use the bulletin board. You can explore, but you may not be able to download any software. If the FIDO you've chosen won't let you have full privileges immediately, your first "visit" is a good time to familiarize yourself with the FIDO system, which is very easy to use, and to read the message sections. You will get a short course (published in the last PC SIG Newsletter) on how FIDO works. The main thing to remember is that there are three areas, Main, Files, and Messages. Each has its own commands. The commands are single letters whose meanings are spelled out right above the prompt.

11. If the Sysop knows who you are, he may automatically upgrade you, or he may call you. Wait a day or two and call back. Follow the sequence above when you call back.

12. After calling back, you will be asked again for your password. Enter it.

13. After the FIDO logo and messages from the Sysop, you will receive the prompt for Main Commands. It will look something like this:

MAIN Commands:
M)sg-Section F)ile-Section G)oodbye
S)tatistics A)ns-Questionnaire B)ulletin Y)ell C)hange E)ditorial
V)ersion
Main M F G S A B Y C E U or ? for help

14. To find out what file sections are available, type

F <return>
A <return>

15. You will get a list of the File Sections Available which will be similar to this list (from Wash-A-RUG):

----- File Areas -----

1 ... E:FILES\USER\DOC\On-line user documentation
2 ... E:FILES\RAINBOW\PROGRAMS SPECIFIC TO THE DEC RAINBOW
3 ... E:FILES\LISTS\FIDO nodelists and BBS lists
5 ... E:FILES\FIDO\Rainbow-specific FIDO distribution
6 ... E:FILES\FIDONEWS\THE FIDONEWS
7 ... E:FILES\MS-DOS\MS-DOS Files
9 ... E:FILES\FIDO\UTS\FIDO MAINTENANCE UTILITIES
10 ... E:FILES\UNIX\MS-DOS Utilities with a UNIX(tm)
flavor.
11 ... E:FILES\COMM\Communications programs
12 ... E:FILES\USR\UPLO>User uploads (after they have been
checked)

16. You will be asked to choose the file area that you want. Type in the number and <return>.

17. You will get the following prompt:

File Area #2 E:FILES\RAINBOW\
A)rea-Change L)ocate F)iles T)ype G)oodbye U)pload D)ownload
S)tatistics M)ain-Menu R)aw-Display
File: A L F T G U D S M R or ? for help:

18. To get a list of the files available, type

F <return>

19. You will see something like this:

File Area #2: E:FILES\RAINBOW\

RAINBOW FILES

These files are specific to the DEC Rainbow. They may not work properly on other micros. They are marked as either CP/M or MS-DOS. Use of these on non-DEC hardware is not recommended.

MS-DOS

BUFFER.COM 1024 Increases Rainbow keyboard buffer size to 189 characters.
QIX.COM 28836 The arcade game. From the Gnomes at DEC.
MONSTA.EQE 29824 A full-screen, arcade-quality, video game. From the Gnomes at DEC.
RESETRB.COM 9771 Ever get your Rainbow stuck in graphics mode? This program will set all attributes back to normal.

AME86 is a DEC-written, copyrighted, public-domain package that allows a Rainbow to run CP/M-86 programs under MS-DOS.

AME86.DQC 24562 Documentation for the AME86 package.
AME86.EQE 8280 AME86 itself.
SEDT.LBR 84480 A version of the VAX/VMS EDT editor that runs under MS-DOS (using AME86) or under CP/M-86. Everything you need (and more) is in this library. Written and copyrighted by DEC, but public domain.

SEDTV12.LBR MISSING This is the latest (MS-DOS only) version of SEDT. It is currently located in the USR-UPLO file area.

DED.LBR 50048 Another full-screen editor for the Rainbow. The library contains full documentation. It is copyrighted, public domain, "freeware". It is really fast!

FV.EQE 5591 File viewing utility based on the DED editor.
FV.DQC 1519 Documentation for the above.

HISTORY.EQE 3712 Allows you to edit and re-issue commands you have used.
HISTORY.DQC 2610 Documentation for the history command

ALARM.LBR 23040 Allows display of messages at scheduled times.

FANCYEXE.LBR 61952 FPRINT and FEDIT along with documentation (MS-DOS)

FANCYFON.LBR 52352 Font files for use with FPRINT.EXE and FEDIT.EXE
FANCYSOU.LBR 52992 Source code for the above.

CP/M

DOSFLX.CQD 28032 This utility allows you to copy files from CP/M formatted disks to MS-DOS disks, as well as from MS-DOS formatted disks to CP/M disks. Much more powerful than RDCPM.

BOTH

BOOT.LBR 16000 DEC written & copyrighted (P/D). Allows PC-100A owners to get the winchester boot menu. Also allows booting to another partition w/o rebooting the Rainbow. Contains documentation and MS-DOS & CP/M executables.

20. Write down the names of the files you want to copy. The number in front of each file description is the size of the file in bytes. (The length of the file determines how long it will take to transfer it across the phone lines.)

21. You will get the Files prompt back at the end of the list. If you want to download a file, type

D <return>

22. You will get the following menu:

```
Download File(s)
Transfer Type
A ... Ascii      K ... Kermit
X ... Xmodem     XC ... Xmodem/CRC
M ... Modem7     MC ... Modem7/CRC
T ... Telink    TC ... Telink/CRC
? ... Help      Q ... Quit
```

A K X XC M MC T TC Q;

23. At the prompt, type in

TC <return>

(Note: you can type the last two commands together: D TC, if you don't need the menu.)

24. You will be prompted for a filename. Type in the drive and the name of the file:

```
B:AME86.DQC <return>
<escape>
R <return>
*
```

(Escape is function key 11; R is the letter R, for receive)

25. You may need to press return again once or twice in order to engage the transfer program. Once it engages, you will get the following notation at the top of your screen (it will be all on one line):

```
AME86.DQC ----- awaiting data_block
Block: 4 ----- Time Left: 5:38 ---- Tries Left: 10 --- (Telink)
(CRC) -----
```

26. The block and time figures will change as the file is downloaded. (If the transfer doesn't work the first time, and you run out of "Tries left", try again.)

27. When the transfer is successfully finished, you will get the notation:

1 files sent OK

28. You will get the prompt in 17 again. If you want to download another file, repeat the process. You can move to other file areas using A (for Area Change) and download from them using the same process. When you are finished, type either M for Main Menu, or G for Goodbye. Before you log off, you will be given a chance to leave a note for the Sysop.

29. After you log off, and hang up, you will still be in DECMINI. To get out, type

```
<escape>
Q
```

which will put you back in DOS.

You should find the file in the drive you specified.

[A Note on DECMINI: To get the menu, type

```
<escape>
```

You will get the following menu (the newest version of DECMINI has an extended menu, not printed here):

```
-----
File Transmission and Reception
```

```
R .... Receive file(s)
T .... Transmit file(s)
N .... Select file transfer mode
```

```
Character and Line Protocol
```

```
F .... Full Duplex
H .... Half Duplex
M .... Line Feeds Off
N .... Line Feeds On
V .... Set parity
```

```
System
```

```
? .... List TELINK Status
C .... Collect Console Text
B .... Set Baud Rate
Y .... List Disk Files
S .... Stop Collecting Text
Q .... Quit to DOS
-----]
```

That's it! There are two things which may slow you down: squeezed files and library files. If you see a "Q" in the file extension (last three letters, such as ".EQE"), the file has been squeezed. If the file extension is ".LBR", it is a library file. You will need to get USQ (an unsqueeze program) and LU (a library utility), either by downloading them or from the Public Domain Library, in order to use the files. USQ is on Volume 32, and LU is on Volume 31.

You can also use DECMINI to transfer files between two Rainbows, or between a Rainbow and any computer which has a version of Minitel on it.

(c) 1985 by Caroline M. Mack

=====

RAINBOW QUESTIONS AND ANSWERS

=====

Send your questions (along with the answer if you have one), to the editor. If there is no answer with the question, she will attempt to get answers. Be sure to explain the problem clearly and include complete information on how you can be contacted, including name, address, home and work telephone.

Q and A on DBASE II, V. 2.41 (not the DEC-Classified version), contributed by Mark Winston of the Washington Area Rainbow User's Group:

Q. When running the INSTALL program, it crashes repeatedly with an "ILLEGAL PARAMETER" message, and returns to the operating system. What could I be doing wrong?

A. You need to answer the INSTALL questions using UPPER CASE letters.

Q. A legal REPLACE. .NOUPDATE command keeps returning the DATA ITEM NOT FOUND error message. Why?

A. Despite the instructions in the manual, NOUPDATE must be specified before any FOR clause.

Q. The command FIND &x does not work right when x includes leading blanks (originally mentioned in the August-September Newsletter). What can I do?

A. FIND strips leading blanks before seeking a %% match when x is a variable, but preserves leading blanks when FINDing quoted literals. So use FIND '&x' instead of FIND &x if there is any chance x could contain leading blanks which must be preserved for matching purposes (e.g., when building a concatenated key.)

From Jay Bassin:

Q: When using MBASIC, where is the name of the program stored in memory?

A: In MBASIC, the name of the program in memory is stored between memory locations 418 to 428. To ask the name of a program (in case you forget), type the following command all on one line:

```
For I=418 to 428:PRINT CHR$(PEEK(I));:NEXT
```

From Tom Tugman:

Q. What is the difference between a "compiled" language and an "interpreted" language?

A. A computer language is the vehicle which allows you to conveniently give the computer a list of instructions to execute (usually called a program or code). The computer cannot understand the program until it is translated into binary code (sometimes called machine language). The compiler translates the program into binary instructions which can be understood directly by the computer. In a compiled language (for example, some versions of COBOL, Fortran, Pascal or C), the compiler creates an "object module", a group of binary instructions, which, if saved, can be reused without being recompiled. The object module can be run independently; the original language need not be present. An interpreted language (such as BASIC), also contains a compiler, but it translates the program code instruction by instruction, executing each instruction as it is translated. No object

module is created. Each time the program is run, it must be translated again, so the language must be available or the program cannot be run. Because the program must be recompiled each time it is executed, programs written in interpreted languages tend to be slower. They are also less portable.

From the St. Joseph's University Rainbow User's Group Newsletter:

Q. Can I use single-sided, double-density 48 TPI disks on my Rainbow even though it calls for 96 TPI disks?

A. Most (if not all) high quality 48 TPI disks will work. Some brands which have given good results are Verbatim and Max-ell. If the disks format without difficulty, they will probably work. Don't use them without having backup disks for all of your important programs or data.

From the San Francisco DEC PC LUG:

Q. Is there a clock/calendar board available for the Rainbow?

A. Don Brauns of Rainbow Data, Inc., (Culver City, CA) had developed an internal clock and calendar board for the Rainbow which will allow both CP/M-86/80 and MS-DOS to read the date and time automatically upon system start-up. The board is presently being tested by DEC so that Don can offer it as a third party add-in without endangering the warranty rights of Rainbow owners who install it. It was expected to be in production and available late in July or in August. (Will Roberts)

Q. How can I run the printer as a "Logging Device" while I scroll along in LC-TERM? The Print Screen key dumps the current screen, but does not enable the printer like Control P does at the operating system level.

A. Call up the main menu, using [Control-Main Screen], select option 4, "Raw Text File Transfer Menu," then select option 1 "Log terminal output to a file." At the file name prompt, type the printer's filename: "prn". Go back to terminal mode; you will be logging to the printer. To stop logging, select the "Close Log File" menu item. (Stuart Fuller)

Q. Is Unix, or a Unix-like operating system available for the Rainbow?

A. Rainbow VENIX is available from

VenturCom, Inc.
Unisource Software Corp.
71 Bent Street
Cambridge, MA 02141
(617) 491-1264

According to the description, you can run foreground and multiple background tasks, and it requires 256K and a 10 meg hard disk. There is also a program called, Co-Idris, which is apparently a UNIX look-alike, multi-tasking operating system that can co-exist with MS-DOS on the Rainbow's hard disk so users can run applications from both operating systems. Co-Idris requires 128K and a hard disk. It is available from

Whitesmith's, LTD.
97 Lowell Road
Concord, MA 01742
(617) 369-8499

(Michael Bowers)

=====

RAINBOW WISH LIST

This is a very short list. If you have items to add to it, send them either to the newsletter editor or the Rainbow Working Group Chairman, Lynn Jarrett.

1. IBM compatibility
2. Upgrade to PC compatibility
3. Ability to cheaply upgrade from a 100A to 100B
4. Ability to format a single sided IBM-PC disk under MS-DOS
5. VT2XX support
6. VT131 support
7. TECO
8. Portable or Lap Rainbow
9. Print Drivers for non-DEC printers
10. Full graphics capability for All-In-One
11. Full file transfer support (not third party)
12. Hard disk controller cards available as spare parts
13. Make lists of DEC PC users in specific geographic areas available to help form User Groups
14. Rainbow LAN with out the need for a VAX
15. Ethernet/DECnet communications
16. Ability to share the same hard disk between several Rainbows
17. Removable hard disk
18. 33 and 70 meg hard disks
19. Hard disk backup, both internal and standalone
20. Make larger hard disks available without having to remove 10 meg internal hard disk
21. Regis: make Rainbows with graphics cards and Poly-Regis return a 1 instead of a 0 when you inquire from All-in-1 with QA*terminal
22. Battery operated internal clock (see above)
23. Rainbow version of ThinkTank software
24. Sell Rainbows in national chain computer stores
25. Carry Rainbow software and hardware at DEC Business Stores
26. Rainbow "Draw and Paint"
27. Software for children
28. TPU
29. Implement print screen so that it works with all software programs
30. Genealogy software
31. Utility which will crack copy protection schemes so the programs can be put on hard disks and/or back up copies can be made
32. Concurrent MS-DOS
33. Compiler version of GW BASIC
34. Q-BUS for the Rainbow
35. Sidekick for the Rainbow
36. Product support for simultaneous dual-screen operation for applications programs such as LOTUS
37. Take copy protection off of DCS and DDS programs

=====

=====

DECIMATE

=====

RAINBOW TO DECIMATE AND BACK USING POLYCOM
By Alex Garbera

There may be occasions when you want to send a document from a DECmate to a Rainbow, or from a Rainbow to the DECmate using PolyCOM communications software. The DECmate has the capability to use the same communications package, but this is not necessary for sending and receiving text files.

Since WPS word processing has a character transmission package built into it, these transfers are easy. Here is just one example of how it works. To begin with, both the DECmate and the Rainbow must have the same communication parameters; that is, have the same Baud rate, data bits, stop bits, and parity. For example:

No parity
8 databits
1 stopbit
1200 Baud (can be any other rate from 300 to 9600 depending on the modem or the distance of the cable.)

These parameters can be set up in the Systems Option (SO,CC) menus of the DECmate and under the communications menu of PolyCOM.

DECmate to Rainbow

To send a text document from a DECmate to a Rainbow, the procedure is as follows:

1. Create a document such as this:

```
____ START CONTROL ____  
CX  
EOL CR LF  
____ END CONTROL ____
```

Note the number of the document and file it. This document will send a carriage return and a line feed at the end of every line.

2. Specify this as a control document by going into the system options menu (SO) and type "CD" with a space and the name or number that you just created.

3. On the Rainbow side, make sure that the "Communications" menu matches the DECmate parameters, and the "Console" menu is in ANSI mode with "crlf" as the new line mode. In the "Host" attributes menu, the EOL Sync must be set to None, and CR Action as "crlr" (no spaces).

4. With these settings in place, the user can go into terminal mode and presses "Select" "r" to receive a file and give it a file name.

5. Go into the "CX" menu of the DECmate and specify "DH" to send a document to the host. This can be any document that have previously created.

Remember that you are sending a document through the communications port, and not to the printer. This method will cause the loss of special features such as bolding, superscripts, special rulers with word-wrap tabs, etc. It's great for centered text and blocks of information.

Rainbow to DECmate

To send a document from a Rainbow to the DECmate:

1. The DECmate CX menu can be set to "HD" host to document. Give the document a name and press return.
2. go into the PolyCOM terminal mode and "Select" "S" to send a file and specify the name.

Sending text files is that easy. It may take a little coordination between the two users, but it's nice to know that DECmates and Rainbows can talk and share information.

To transfer more complicated files, such as CP/M programs and non-text files, the DECmate will need a CP/M or XPU board and the same PolyCOM software using the XFR and HST functions.

=====

DECIMATE II WPS Version 2.0 Fix
by Cheryl Johnson

One feature of WPS Version 2.0 that everyone has found annoying is the problem with editing a wide document. Instead of automatically going into the wider ruler with compressed characters, it stays at 80 columns and you shift between 'regions' by advancing past the 80th character. Most, if not all, of the time, it is more convenient to view all of the text at one time. It is a real bother to have to change something in a menu to do that every-time. So I have designed a user-defined key (UDK) to automate the sequence of strokes necessary to get to the wider ruler.

To define the UDK, from the Main Menu, type DK, press the space bar, type a number between 1 and 9 (which will be the number you use to activate the UDK). Press return, then type in the following key strokes:

```
Gold M
SM Space Wide
(Return)
(Return)
```

Press Gold Halt (the Halt key is just above the Tab key) to end definition of the UDK.

To use your UDK, edit the document you want to view in the wide mode. Press Gold and then the number you defined above. The stored sequence of keystrokes will be executed. When you file the document, the screen will automatically return to narrow rule. The UDK will remain stored on your system disk until you redefine the number.

(Cheryl Johnson, of the Grinnell College Office of Computer Services, is the DECUS PC SIG's DECmate Working Group Chairman.)

=====

DECIMATE NOTES FROM DECUS NEW ORLEANS
by Cheryl Johnson

--TypeEasy was announced, and is expected to be available in June or July. It is targeted for DECmate III users with LQP03's, but will work on the DECmate II with other DEC printers. TypeEasy gives the DECmate capabilities similar to those on a typewriter. For instance, it can be set so that the printer prints letter by letter, just as a typewriter does, or to print line by line, for used in processing forms.

--The DECmate does support the LN03, but it does not support special functions, such as font loading.

Minutes of the DECmate Working Group Meeting:

Members were introduced.
The following items were discussed:

--The DECmate wishlist

--Technical Documentation for OS278 (part number)

--Larger Personal Dictionary with XPU board or replacement of words in the main dictionary

--Identification of differences between CP/M and MS-DOS between the DECmate the Rainbow

--Identify specific software that will work on the DECmate

--Identify specific software that will work on the Rainbow

--Identify specific software which works on both Rainbow and the DECmate

--Foreign printer ability (DEC response: "thinking about it")

--VT102 emulation for the DECmate (DEC response: "priority for future versions of WPS")

--inability of the screen width wide feature to reset itself

--possible sessions in Anaheim to be sponsored by the PC SIG

- o DECmate Technical Q and A
- o DECmate as a PC
- o DECmate Office Workstation Script Processor

=====

DECIMATE QUESTIONS AND ANSWERS

=====

From Perspective:

- Q. How do I load CP/M on the DECmate hard disk with Master Menu?
- A. Boot the CP/M INSTALL disk (CP/M 2.2, Version 2.0 INSTL BIN RX50) in Drive A and do the following:
1. In response to the prompt for installing CP/M on the hard disk, choose H and press "DO".
 2. In response to the prompt for creating a new CP/M volume, choose C and press "DO".
 3. When prompted for a volume name, enter the name and press "DO".
 4. When prompted for a size, enter a size, keeping in mind that some CP/M files consist of 200K.
 5. After the system copies the CP/M utility programs, press "SET-UP" and "REMOVE".
6. Remove the INSTALL disk.
 7. Press "DO" to boot the hard disk.
 8. At the Master Menu, press "NEXT SCREEN".
 9. In response to the prompt to create a TAG, choose MM and press "DO".
 10. Position the cursor on End of List and press "INSERT HERE".
 11. Put in a three-letter TAG.
 12. Hit the down-arrow key for the next item. Type in LEGEND.

Q. While list processing with WPS-8, I receive an error message indicating I have too many characters in a record. How can I correct this problem?

A. The maximum number of characters in a record is 2,049. If your record does not exceed 2,049 characters and you receive this message, you probably omitted a closing diamond (<>) to end a record. If a record does have too many characters, you can use the following method to shorten the record by shortening the field names. Use the old list, create a spec document saying "process record", and create the following form document:

```
<<(new name)>(old name)>
<<(new name)>(old name)>
<<>>
```

Print to a document and use the resulting document as your new list. For example:

(old record)

```
<(employee name)>
<(employee address)>
<(employee salary)>
<(years with company)>
<>
```

(new form)

```
<<n>(employee name)>
<<a>(employee address)>
<<s>(employee salary)>
<<y>(years with company)>
<<>>
```

(new record)

```
<n>
<a>
<s>
<y>
<>
```

Q. Sometimes while using Multiplan, when I try to copy a formula, the wrong value is copied instead of new calculations being made as a result of copying the formula. What could be wrong?

A. Check your options command and check to see if recalc is off. Also make sure that the cells which will receive the new calculations are blank. Press the shift key followed by the ! to recalculate values based on the formula.

Q. I tried to convert a Multiplan file created under CP/M 80 to a WPS file and thought I followed the directions in the CP/M User's Guide under WPSCONVERT. However, when I went to edit my new document under WPS, there were only garbage characters on the screen. Why?

A. You probably did not run PRINT FILE from your Multiplan spreadsheet first to create an ASCII file or you did not use the correct name of the new ASCII file created. When you use PRINT FILE, name the ASCII file differently from the name which you gave the spreadsheet.

Q. Can I format a cell under Multiplan using the dollar sign (\$) without the two decimal places?

A. The two decimal places are a default in the software and cannot be modified.

Q. How can I print a CP/M file on the printer from the CP/M operating level?

A. Use the PIP command: PIP LST:=(filename)
(LST is the name assigned to the physical device for the printer.)

DECMATE WISH LIST

If you have items you want to include on the wish list, send them either to the editor of the newsletter, or the DECmate Working Group Chairman, Cheryl Johnson.

1. Global search and replace: the ability to do more than one search and replace string at a time.

2. Text wrap: text should automatically wrap when editing so that the user doesn't have to advance through it.

3. Additional paste buffers: the ability to save and rearrange multiple segments of text at the same time.

4. Spelling error detection: the ability to check spelling of text easily and accurately without using CP/M (to be included on DECmate word processing software, possibly as an optional feature), or rearranging document in any way. The current DECspell is very slow and doesn't seem to be very reliable.

5. Simple math logic: the ability to have five-function math (addition, subtraction, multiplication, division and percentages) on DECmate word processing software, possibly as an optional feature. Would like the ability to check columns of numbers and perform simple math operations with a minimal number of keystrokes. The current math software seems very complicated to use.

6. UDKs: the ability to actually do the UDK as the user is setting it up, not just displaying the typed keystrokes.

7. UDKs: the ability to edit existing UDKs which have already been filed.

8. ENTER key: would like it moved closer to the normal typing keyboard.

9. Print specific pages: the ability to print any specified page when reset pages are used; ie., print section II, page 8.

10. Graphics: the ability to easily draw vertical and horizontal lines around text that will print out to form boxes and charts, to be an inclusion in DECmate word processing software.

11. Background transmission: the ability to continue to use the system while transferring documents from the VAX, converting documents, using list processing and other foreground functions.

12. Working with a "COPY" document on disk, not the actual copy, so that if the user messes up, the original can be recalled to use again.

13. Multiple wraps: the ability to have multiple wraps so the user doesn't have to keep changing rulers; possibly by using the GOLD TAB to indicate the specific tab wrap that you desire at that time.

14. Sub- and superscripts displayed: show sub- and superscripts on the screen by actually moving the character up and down.

15. Printer stop: the printer should have the ability to detect when the ribbon breaks (not just runs out), and stop.

16. Printer detects paper out: the ability for the printer to detect when no paper is in the printer so it will not print on the platen.

17. Hard disk volumes: the ability to choose the size of segments on the hard disk, instead of having to break it up into 7 small volumes.

18. Software "q" problem: the ability to use GOLD SEARCH PAGE or GOLD PAGE BOTTOM without worry that a hidden "q" will be put in the document to cause various problems.

19. Screen scrolling: have screen scroll a single character at a time in wide documents, rather than jump.

20. Local software support: provide local support to help with special set-ups and problems. Sometimes you need someone to actually be there and share ideas with, not just a voice over the phone.

21. Wide screen setting to stay with the document: With version 2.0, giving users the ability to change the size of the screen. 2.0 does not allow the wide screen to be stored with the document. The chosen setting should stay with the editor menu (narrow or wide), to be stored with the document.

22. Cutting text of any size: to have the ability to cut any amount of text, possibly by making saving and cutting separate functions. That way, since the system wouldn't be saving it, the size of the text to be cut would not matter.

23. Replacement character on draft printer: the ability to print replacement characters on a draft printer going through communications, particularly double underscores.

24. Advance forward: to be able to advance directly to the end of a word, sentence, or segment of text.

25. Document view while printing: the ability to view or edit a document while other pages of it are being printed.

26. Multiple simultaneous print queue: the ability to queue several non-consecutive pages of a document to the printer at the same time: ie., queue pages 2, 6, and 8, or at least to be able to send page 8 of a document to the printer while page 2 of the same document is printing, so you can then work on something else.

27. Underlines recognized as part of a word: the ability for the DECmate to recognize underlines before text that are entered using select and the "UNDERLINE" key to be recognized as part of that word, i.e., 846. This would make the system back up to the beginning of the underlines when "BACKUP WORD" is pressed at the end of the word/number or delete the word/number with underlines consecutively when "DEL WORD" is pressed.

28. Larger capacity in DEC DX account: the ability to put more than 200 documents in a DEC DX account, up to 999 or so, if limited by three digits.

29. Recreate index: the ability to automatically recreate the index of a DEC dx account if it becomes corrupted.

30. Caps lock sound recognition: a different key sound when the LOCK key is on to signal the operator the lock key is on.

31. Replacement character view: the ability to see numbers/characters which have replacement characters, such as double underscores, onscreen.

32. Gold GRT DOCHT PAGE: use GOLD GET DOCHT to get only certain specified pages of a document.

33. Page markers: page markers not to be removed when rulers or text are changed.

34. Delete page key: the ability to delete an entire page at a time, allowing the operator to verify deletion of the page.

35. Double underscore with decimal tabs: the ability to use double underscoring with decimal tabs without having the text move over.

36. Simple column logic: the ability to set up multiple columns on a page by putting more than one set of margins on a page, also, allowing the use of the sheetfeeder during multi-column printing, since the page would not be going up and down.

37. Column centering: the ability to easily center headings over a column of text.

38. A place to get detailed information addressing software for wierd applications, such as PDB real time data acc replacement.

39. VT131 and VT2XX support.

40. A way to split out DX and compile it into OS278 (DECUS version).

41. Address comm port under COS 310.

42. Ability to use print screen with FMS forms that have line drawings.

43. Full compatibility with the LA210 and LN03.

44. Elimination of hard returns when transferring a WPS+ document from the VAX to the DECmate.

45. Capability to put escape sequences in a document, e.g., print a line at 5 characters per inch, then print the next one at 10 characters per inch, etc.

46. Support printer as a local printer, such as "VT100 printer port escape sequence."

47. Gold Q for Quit on DECmate and Rainbow WPS+.

48. The new 'search forward or backward' is great, but the user can't change direction after it is chosen. Go back to the old way.

49. Ability to use non-DEC printers.

50. The ability to call up the index in alphabetical order by document name.

51. The ability to use the system while doing list processing.

52. The ability to add information on to the end of a paste buffer before pasting, ie., cut a paragraph, then on the next page, add that to another paragraph, cut again, and past both paragraphs elsewhere, in one operation.

=====

PRO

=====

PDL: AN ONLINE PUBLIC DOMAIN LIBRARY FOR SCIENCE AND RESEARCH APPLICATIONS
by Linda Rosenthal

Digital's Laboratory Data Products Group in Marlboro, Massachusetts, has developed an on-line library of public domain research applications. The applications can be downloaded to users over the telephone lines with a 300 or 1200 baud modem and appropriate communications software.

The library is not affiliated with DECUS. It was established to serve as a focal point for applications and tools specifically intended to help the scientific researcher. In addition to providing a portfolio of FREE research applications and tools, the library has a public bulletin board facility which allows users from different laboratories to post and respond to a wide range of technical issues as well as to share information about software from other sources.

Laboratory Data Products (LDP) encourages contributions to the library. DECUS was very cooperative in getting us started almost a year ago by allowing us to place some of their research applications in our library. Since that time, contributions to the library have more than doubled, with the majority of the software aimed at the PRO 350. It is anticipated that the library's portfolio will be expanded to include applications and tools for other Digital processors used in the laboratory, including the Rainbow, MicroVAX, and VAX. In addition, pointers into other databases of public domain software, such as NASA's COSMIC library, Argonne's National Energy Software CENTER, and DECUS are also expected.

PDL is a dynamic database, and for that reason, does not publish or distribute a catalog. To find out what is in the library, you need to dial into the system. Following login to the PDL, there will be a short pause before the MAIN MENU appears on the screen. Options available on the MAIN MENU include:

*NEWS --a collection of short articles and information on hardware and software products for Digital computers

*NOTES --a public bulletin board facility for posting and responding to product issues

*SEND --a "private line" to Laboratory Data Products management

*DOCUMENTATION --listing of documentation available for the PRO 350

*APPLICATIONS --a choice that leads to a secondary menu which allows you to peruse application abstracts and/or invoke the file transfer utilities available on PDL. The APPLICATIONS MENU lists major categories of applications such as Graphics, Report Generations, Database Management, Instrument Control, and Compiler applications.

Transferring software from the Public Domain Library to your Digital computer is relatively easy. Any one of several file transfer protocols can be invoked from the library menu, including PFT (from Digital), Poly-Xfer (from Polygon), and KERMIT (from Columbia University). Kermit is one of the applications available from the library. It is a popular communications package of considerable power and elegance which allows users to copy both source (ASCII) and image (binary) files under an error-free protocol. Kermit is available for VMS, PRO/VENIX, P/OS, CP/M, MS-DOS, TOPS-20, RSX11M, RSTS, and RT-11 among others.

To use PDL, call (617) 467-4824. To log onto the system, respond to the system prompts as follows:

Username: PDL
Password: PDL

If you have any questions, comments or suggestions concerning the PDL, call Laboratory Data Products at (617) 467-6057.

Linda Rosenthal is a Senior Marketing Communications Specialist with DEC's Laboratory Data Products Group.

=====

REQUEST FOR PRO-300 SERIES ENHANCEMENTS

by Thomas R. Hintz
Chairman, PRO Working Group

Use of the PRO-300 series microcomputer continues to grow. The PRO finds its way into many diverse areas and has found favor with many users because of its ability to support several languages, and provide both the experienced and inexperienced user with a comfortable environment in which to work. Many existing applications can be easily ported to this machine.

Along with its use, however, comes an increasing number of requests for modification and upgrade. Since all PRO-users have a common interest in the future development of the PRO-300 series of micros, the PC SIG has started a wish list. Since a public list does not exist, I have been collecting wish list requests over the last month. During the New Orleans DECUS Symposium, the list has almost doubled in size. The variety of requests shows a desire for a broad range of future product enhancements. The broad spectrum of interest for the PRO was evidenced by the large number of SIGs which sponsored sessions related to the PRO. A total of eight SIGs gave more than 35 presentations.

The list below is presented in its entirety, but is NOT prioritized. Most of the requests are quite specific. Where the meaning was not clear, the wishes are listed as they were submitted. If you wish to clarify a particular item, please feel free to add to the description. We will continue to add to the list as new items are submitted. Any items that are important to you which are not on the list should be sent to me for inclusion. To continue this process we need to bring some order to this random list of suggestions. You, as a user of the PRO, should have an opportunity to determine their order of importance. Now is your chance. At the end of this newsletter is a form (ballot) to cast your votes. Each item is numbered. Number each wish to show what you consider to be the items of highest priority. Vote for as many items as you like, but number them from highest (1) to lowest priority. After the ballots have been returned, the results will be tallied, reported in the DECUS PC SIG Newsletter, and presented to DEC for their review. Now is your chance to be heard. VOTE TODAY before you forget.

WISH LIST ITEMS

1. RT-11 emulator executing under P/OS
2. BUS extension to provide more expansion slots
3. Streaming tape backup
4. Standalone backup
5. External disk(s)
6. Wild card specification for PFT
7. Warm restart for P/OS
8. Startup detect of battery backup status
9. Instructions for deleting unused application options to conserve disk space
10. Menu item to execute infrequently used applications from disk so that install/deinstall is not required
11. VAX server for cluster of PROs
12. Terminal emulation with full VT24X (e.g. downloadable characters, etc.)
13. DECNET command terminal should be compatible with VMS 4.x (e.g. should allow command line editing)
14. DECNET support for the communications port
15. Ability to spawn a BASIC compilation
16. Batch spooling facility on toolkit for compilation/link

17. Ability to connect two hard disks to to the bus, preferably utilizing the same controller.
18. Ability to dial into the TMS and the communications port from a remote terminal
19. PRO/FMS compatible with VMS/FMS
20. More compatibility between FMS screen control keys (function keys) and the P/OS menu control keys (e.g. P/OS uses the HELP key for help while FMS uses the PF2 key, etc.)
21. Disk formatting capability
22. Access to the DZ driver or the ability to use the other track/sector combinations supported by the RX50 controller
23. Disk compression utility
24. A method of finding out what files to move to recover contiguous disk space so that a complete reload is not required to unfragment the hard disk
25. A search command similar to what exists on VMS for finding text strings in a file or files
26. TPU (Text Processing Utility) for P/OS like the announced but not yet delivered text editor for VMS.
27. A more elaborate PRO/COMMUNICATIONS that makes full use of the TMS and voice box
28. Full window/multi-task support for user developed applications (i.e. SIDEKICK)
29. Enhanced CAD software (not Design Graphics Executive!)
30. DCL sources on Toolkit
31. Reliable PASCAL compiler
32. Complete documentation and listing of the VT102 and GIDIS (TFW) code
33. Graphics from VAX using PRO/COMMUNICATIONS V2.x
34. Maintain color setting when going between menus
35. A better DEC LANDER!
36. Asynchronous dial-up DECNET
37. Full VT125 emulation
38. Full DECGRAPH/DECSLIDE support
39. MENU sources
40. Output CORE graphics to GIDIS files (from PRO/BASIC)
41. PRINT SCREEN to a GIDIS file
42. Image backup of Winchester disk
43. Removable hard disk
44. Virtual terminal support
45. BATCH processing support
46. I D space support for 325 and 350
47. Supervisor mode support for 325 and 350
48. Larger buffer for drawing complex filled figures
49. Warm restart of P/OS
50. SIGHT-rotation of figures and text
51. GIDIS to/from NAPLPS conversion
52. GIDIS to REGIS conversion
53. HELP on floppy, less on hard disk
54. Master index to toolkit documentation
55. Complete documentation set, etc. on laser disk (with cross index)
56. Extending graphics for PRO/BASIC
57. Graphics support over DECNET
58. FPJ-11 floating Point for 380
59. Videotex creation
60. SIGHT-convert multiple objects into single one
61. Coexistence of multiple OS on a single hard disk for each to run in native mode
62. DECNET support under RT-11
63. Do not clear screen after logoff with PRO/COMM TMS

(Tom Hintz is an entomologist with the University of Florida, and long-time chairman of the Pro Working Group.)

=====

ROBIN (VT 180) NOTES

=====

by Richard D. Ferguson

--Jurgen Steiger of Zurich has written a program which will allow a Rainbow to write a floppy disk that is Robin-readable. For more information, contact John Comella, Robin Owner's Group, P. O. Box 492, Rollingsford, N. H. 03869.

--Exceptional Business Systems has issued version 1.3 of WPS-80, which provides bug fixes for earlier versions and new features, including printer support for the LQP03, LA210, and others; printer commands to change pitch, select sheet feeder trays, include escape sequences; ability to strip hard carriage returns in a CP/M document; improved super- and subscript handling; improved up- and down-cursor keys; and footnotes of up to 1000 characters. It also has list processing and mail merge. Although there is no math or sort, but it will allow the selection of certain records for merging selected fields. For Robins with RX50 floppies (80 track), it has a format converter which will convert from WPS-8 to WPS-80.

--Sources of VT-180 Software. Current concerns of Robin owners seem to focus on the availability of applications software--accounting, financial, utilities, etc., most of which is being sold at discount for CP/M. The original DEC CP/M Applications Software Referral Catalog for ROBIN lists most of the compatible packages that are available in VT-180 format, but many of the really useful ones (e.g. dBase II) are not listed. Here are some clues to help you find what you want:

1. All standard CP/M applications will usually run on the ROBIN; the best alternate disk format hat I have found is the Kaypro II format. It converts to the VT-180 disk format and causes no loader problems. It is a format that almost all dealers stock for most CP/M software packages. Many are now heavily discounted. With the declining number of VT-180 disk formats being carried, this is a good alternative.

2. Ken Heyda's MULTISYSTEM program is a path to non-ROBIN software; it enables you to copy/convert programs from other system's disk formats, and offers broader software choices. It is worth every penny; it costs (\$70), and will pay back the investment in applications software. there are other format conversion programs (MediaMaster, Multidisk) and with dealers no longer supporting VT-180 format, ROBIN owners are should not be without one of these essential tools.

3. Much free software is available through Bulletin Boards, other CP/M user groups, and the National Public Domain Software Center.

Richard D. Ferguson is editor of the Robin Owner's Group Newsletter. This was excerpted from that Newsletter.

=====

GENERAL INFORMATION

DEC-SPECIFIC COMPUTER MAGAZINES AND PUBLICATIONS: WHERE HAS THE RAINBOW GONE?

by Caroline Mack

Over the past several months, the Rainbow (not mention the PRO or DECmate) has virtually dropped out of sight in most publications. Two magazines which were originally dedicated to the Rainbow and Pro, Digital Review and Personal and Professional, have changed radically. Digital Review has drastically reduced Rainbow coverage, and mixes the small amount of coverage it does have in with coverage of VAXES and the complete range of DEC minicomputers. Personal and Professional has ceased publication, except as a small set in section in the DEC* Professional. Here is what is currently available:

Magazines:

The DEC Professional Plus Personal: (monthly) primarily covers VAX. The "Plus Personal" section takes the place of Personal and Professional magazine which is now defunct. Typically there are a couple of articles and reviews, a list of Rainbow User's Groups (changes each month), and a list of companies who supply software (they pay for the privilege of being listed). It does have a new product section. Few Rainbow software advertisements. Even though the Plus Personal section is small (24 pages in the next issue), because it appears monthly, there may be nearly as much information as there was in the bi-monthly, anemic Personal and Professional. The "Plus Personal" version of the magazine has about 10,000 subscribers. Subscription: \$21/year (12 issues). For more information, contact The DEC* Professional + Plus Personal, P. O. Box 362, Ambler, PA 19002-0362, (215)

Digital Review (monthly) also primarily covers VAX equipment, but it does still carry some Rainbow related articles. The April issue, which is fairly typical, had an article on adventure games which are available for both the Rainbow and the PDP-11. It did mention PCs--IBM PC's, that is. There is a new product section (Rainbow products are mixed in with all the others, so it takes a while to find them) and a section on DEC news. Oddly, the question and answer column (Tech Talk) usually has more Rainbow questions and answers than VAX. Few Rainbow software advertisements. Very slick looking. Subscriptions are free to those who qualify; otherwise they are \$29.97/year (12 issues). For more information, contact Digital Review, fourth floor, One Park Avenue, New York, NY 10016, (212) 503-5110.

HardCopy (monthly) also concentrates on Digital's larger systems, but it appears to be increasing coverage of the Rainbow (they estimate that there are about 30,000 readers with Rainbows). It has recently added a PC question and answer column. There is an ongoing list of user's groups. The February issue, which seems fairly typical, had an article on Rainbow tax programs. Subscriptions are \$16/year (12 issues). For more information, contact Hardcopy, Seldin Publishing Co., P. O. Box 759, Brea, CA 92621.

Newsletters:

Perspective is DEC's quarterly newsletter. The glossy 48 page publication is particularly notable for its extensive Question and Answer section on DCS software. Source of DEC Product Line Information. Free. For more information, contact Perspective, Digital Equipment Corporation, 40 Old Bolton Road, Stow, MA 01775.

DECUS PC SIG Newsletter (this newsletter) is the quarterly newsletter put out by the DECUS PC SIG. Starting September 1, all DECUS SIG Newsletters will be folded into one large monthly publi-

cation (estimated cost is \$35/year. For more information, call Carol Dunbar at DECUS, (617) 480-3418.

Several Local User Groups put out newsletters. Here are some:

The Washington Area Rainbow User's Group Newsletter (monthly, except August), 16-24 pages, features articles, question and answer, and software reviews. It is monthly except August. Currently free.

The Boston Computer Society DEC Personal Computer User's Group Newsletter has recently resumed publication. Edited by Anu Pareek, 3 pages. Notes from the group's chairman, meeting notes, reviews, and advertisements.

The Delaware Valley DEC Personal Computer User Group News (five times a year), 12-14 pages, edited by Tom Deahl, features articles, editorials, and software reports.

The Hartford (CT) Rainbow User's Group Newsletter (quarterly), 4-12 pages, edited by Jean Whitney. Features meeting notes, articles.

The Mid-Tennessee DEC PC LUG Newsletter (monthly), 3-4 pages, edited by Donald Goss. Features notes from the editor, meeting notes, articles.

The Northeastern Connecticut User's Group Newsletter (monthly), 6-18 pages, edited by Wilbur J. Widmer. Articles, how-tos.

San Francisco Bay Area DEC PC User Group News, 4-10 pages, edited by Dale Miller.

Santa Barbara, Ventura, San Luis Obispo County Area DEC-PC Local User's Group Newsletter (quarterly), 2 pages, edited by Rick Vincent. New.

Silicon Valley Digital PC User's Group Newsletter (monthly), edited by Bill Horton. Meeting notes, occasional articles.

St. Joseph's University Rainbow User's Group Newsletter (4-7 times a year), 4-6 pages, edited by Val Herzfeld. Information.

University of Pennsylvania Meeting Notice (monthly), 3-6 pages, edited by Bill Gavelis. Information, articles.

Robin Owners Group (ROG) Newsletter (?), 10 pages, edited by Richard D. Ferguson. Information, articles.

For ardent and wealthy DEC-watchers, Adolf (Sonny) Monosson puts out two informational DEC related publications in addition to the quarterly Monosson's DEC* Compatible Buyer's Guide:

Monosson on DEC*, (monthly) a newsletter which covers DEC related topics in depth. The DEC PC's are infrequent topics. The newsletter is available for a steep \$395/year from Monosson on DEC*, P. O. Box 71, Kenmore Station, Boston, MA 02215, (617) 267-2900.

The newest Monosson publication is Monosson's DEC* Market Weekly (50 issues a year) which covers current DEC topics. The issue I saw had very timely information which was not published elsewhere. The subscription fee is \$245/year. For more information, contact Monosson's DEC* Market Weekly at the address above.

Other sources:

Computers-R-Digital is another DEC related publication, which appears to be half-way between a newsletter and a magazine. Monthly. It consists of articles on DEC. Since I haven't seen a recent

issue, I don't know what the percentage of Rainbow related articles and advertising is. Subscriptions are free to those who qualify, otherwise are \$15/year (12 issues?). For information, contact Directory Database, Box J, Navesink, NJ 07752, (201) 291-1208. If you just want to look up articles on a specific topic, try the Microcomputer Index, put out by Microcomputer Index Company, P. O. Box 50545, Palo Alto, CA 94303, (415) 948-8304. The Index is available both in hardcopy and on-line through Dialog (File #233). The magazine version costs \$60/year for a subscription (it comes out 6 times/year). It indexes articles from about 75 sources, 40 of which are popular computer magazines, and abstracts some articles. The Index has recently been taken over by a new company, and is still trying to catch up. The on-line Dialog version has articles through October, 1984 indexed, but should soon have November-December. Once 1985 is up to date, the on-line version will be updated monthly.

(This article originally appeared in the WARUG Newsletter.)
(c) 1985 by Caroline M. Mack

=====

WHAT WOULD YOU TELL DEC IF YOU HAD THE CHANCE? WOULD YOU BUY ANOTHER RAINBOW?

Would Rainbow owners/users buy another Rainbow? This was one of the questions asked on a questionnaire in the July issue of the Washington Area Rainbow User's Group Newsletter, which went to WARUG members all over the United States. Here are the results on the questionnaires received as of July 28, 1985:

If I had it to over again, I:

Would buy another DEC Rainbow again.	43%
Might buy a Rainbow if it were a bargain.	46%
Would not buy another Rainbow for any reason.	11%

Another of the questions asked in the current Washington Area Rainbow User's Group Membership Questionnaire was, "What would you tell DEC if you had the chance?" Here is a sampling of the answers from members:

--Extend the hotline for technical support (dial-up advice and "hand-holding") to those with no warranty support. Every software package says "call DEC at 800-DEC-8000 for answers to questions, but you can't get in without an access number.

--Provide IBM compatibility.

--Get more third party involvement in the PC area.

--Create a fixed/removable hard disk, forthwith.

--Try to get a little more compatibility with IBM. . .there are a few programs we miss--although not many. I'm quite pleased with support.

--That they're blowing it by not having a more aggressive Rainbow marketing policy in academe, where the strongest, most loyal users are.

--I am writing Ken Olson about my problems.

--BZ-zz-zz-z-z!!!

--Make a user's guide that has examples--an index to find things (the present one is not complete enough).

--Keep up with current technology in your micro PC products, at a reasonable price.

--1. Stop ignoring the stand-alone Rainbows

2. Lower prices.

3. Cooperate with 3rd party software companies to encourage more software.

4. Provide "Perspective" [quarterly publication] on a monthly basis with more user-oriented material on how to use particular features of the Rainbow.

--Support the Rainbow as a stand alone unit. Get sales of Rainbows back to the local level so that Rainbow users can get what they need.

--Support the Rainbow and encourage third party development. Support small business needs as well as VAX users.

--Firm up your PC strategy, from the DECmate up through the VAX.

--Masterful example of how not to market PCs. Very short-sighted to restrict choice of printers and to not open up the market for 3rd party manufacturers.

--Please recognize the importance of your customers to your own survival as a company. Get your sales and marketing act together. (From a company with over 400 Rainbows.)

--Lower your prices and support what you sell.

--DEC blew it! Their marketing strategy for the Rainbow stinks! A more liberal and helpful marketing strategy for the retail stores could have made the Rainbow the best selling computer on the market. You have to get the computer to the people as well as get the people to the computer. As for their support, I highly commend them for the Atlanta hotline. It is a good idea and it works! Their local support and software purchasing policies have and anti-customer attitude. How does a company expect to sell products if they will not demonstrate them in their own stores! I think if they want to continue being a factor in the micro-computer market, they have to do some rethinking and improve their marketing for this line of products.

--The company is too difficult. We need a lot of telephone numbers to call for parts, etc. The hot line was worthless when I was using it.

--They need to publish (or make available) a book containing phone numbers to call for information on various subjects. In the past, I haven't been able to find out anything because I don't have the number of anyone who knows what I want to know.

--I've had, and I've done. I'm hoarse.

--Drop dead! Overpriced hardware, insufficient and overpriced software.

--User's don't desire closed systems. PC's won't go away even when networked to hosts. Third party hardware and software is needed. DEC should fix design problems, such as floppies without hubs.

--Continue to support the Rainbow.

--Lower your prices on add-ons and software.

--Reconsider your policy regarding sale of Rainbow as a PC through retail outlets.

--Improve the hotline or provide alternate methods of technical support. Make third party software for the Rainbow available more quickly. (DESQ was about 18 months late.)

--They started a PC venture. .now they can't simply walk away from it, like a reluctant parent. They've got the resources to provide better support (better = more responsive + communications oriented) and if they don't they'll lose thousands of potential future customers.

--Make software/hardware available that will:
1. allow full IBM compatibility
2. allow full DEC PRO compatibility and
3. documentation/software to address/operate ports without going to assembly language.

--Increase user knowledge of the machine and make hardware changes easier.

--Don't kill the Rainbow!!

--They have really orphaned the machine! Why was the machine not made easily expandible, why was the A developed with such short sightedness, why stop the TRUMP (A to B) upgrades?

--It's an excellent machine, better than IBM PC. If it were only mass-marketed, there would be a greater variety of third party software and hardware available! I think this is the Rainbow's only disadvantage, but it's a BIG one.

--You should offer a PC100A to PC100B upgrade at a very low cost. This will keep users who were loyal to the Rainbow all along current with the latest hardware. It will also give them incentive to buy new hardware that works better on the B unit, such as the hard disk or the graphics option.

--Continue to support the 100; lower prices to compete with IBM; provide compatibility with the IBM PC.

--Make the Rainbow PC compatible, advertise it, and sell it in retail stores again. Give hotline support to all users, regardless of warranty or service contract.

--The Rainbow is an outstanding machine--don't let it die for lack of new software, add-ons, upgrades, and user-support.

--Support Rainbow users.

--Make the Rainbow IBM PC compatible.

--Hey guys, your documentation needs improvement, we need to know more about the internals and how to get the machine to do what we want.

--You're all cranked-up (production-wise), so why quit now? Better does prevail! Make an expansion chassis with IBM PC slots!

--Do not abandon the Rainbow! Develop/distribute/fix the utility that is supposed to read and write IBM PC diskettes, especially for MS-DOS.

--You have to be kidding! Where should we start? Have the DEC marketing staff take a level 100 Marketing course, and get the word out about the Rainbow. Make a firm commitment to the PC or drop it and sell third party manufacturers the rights to it.

--IBM compatibility. Either get into the PC market and support it, or get out. Support of Rainbows and PROs has been shameful and left everyone with a bitter taste toward DEC. The owner's only salvation is the LUG groups.

--Need much, much more software support and support for the Rainbow as a PC. Also, I don't understand why Wordstar, which is one of DEC's DCS programs, doesn't support DEC's sheet feeder.

--Would like to more buy more lower priced software. Want to see a better commitment to the Rainbow. Want more information on Rainbow technical manuals.

--Why is even simple stuff so hard to do?

--You've got a good machine, learn how to market it! Get on the ball before it is too late.

--Their marketing strategy is deplorable. They have the best PC on the market--why not go for it! Why don't they encourage software producers to write for their family of machines?

--It will be a cold day in hell before I recommend DEC to anyone.

--More software, especially educational.

--I would tell Mr. Olsen that PC's started in the home and will have their greatest impact on society because they let us work at home. PC's are viable!

--It's a real down having them get out of the PC business.

--Admit IBM exists!

--Support Rainbows and UNIX.

(c) 1985 by Washington Area Rainbow User's Group

=====

POINTS OF ETIQUETTE
Ms. Motherboard

Dear Ms. Motherboard,

I started a new job last month with a "Beltway Bandit" firm in Maryland. I was hired as a secretary for a group of programmers and analysts. Everything was going along really well, until they gave me a microcomputer last week. They told me I should learn how to do word processing on the machine to handle all their proposals. They gave me Word Star and a box of floppy disks and said to try the tutorial programs before anything else. Well, I started by taking a floppy disk out of the box and trying to figure out how to use it in the microcomputer. I took it out of its envelope and removed the hard wrapper around it (they really pack those disks well, the plastic was almost impossible to remove). When I finally got it out, the flimsy round plastic thing didn't fit into the disk drive. When I called my boss to ask for help, he was furious. He swore at me and asked what I thought I was doing tearing up floppy disks. He fired me on the spot with no explanation of what I did wrong. I feel awful, because I still don't know what got him so angry. Can you help me?

Yours hopefully,
Susie Ex-Sec

Dear Susie,

Ms. Motherboard deplores acts of anger and (gasp!) coarse language. Your boss was probably upset because you inadvertently destroyed a WordStar system disk. Floppy disks are self-contained. You aren't supposed to cut the plastic off from around them. In fact, you should be very cautious when you handle them. They should not be bent or written on, and you should never touch the interior plastic section. A finger print, hair, cigarette ash, or drop of a soft drink can destroy them.

Your boss should have given you another chance. . .however, since he didn't, Ms. Motherboard suggests you take a course in word processing at a local community college, adult ed center, or some other inexpensive place. You'll not only gain new skills and confidence in dealing with a computer, you'll learn enough about computers to avoid another disaster. Good luck!

Dear Ms. Motherboard,

There's this guy in my math class who thrills me to the max. I mean this is the real thing. But he keeps asking me over to his house to see what tricks he can do with pixels. I thought I was with it, but what in the universe is he talking about?

Love ya,
Purple Paula

Dear PP,

Until recently, Ms. Motherboard herself was under the impression that a pixel was a female pixie. However, she has discovered that a pixel is actually a dot. The computer's screen is made up of a matrix of little dots which are either on (lighted) or off (dark). These dots are arranged in patterns (depending on which are on or off) that form the characters seen on the screen. When the text is moved (or scrolls) these patterns are shifted by turning off the previous dots and turning on the adjacent dots (in the proper direction). Pixels can be used to produce pictures and other graphics as well as characters. However, Ms. Motherboard hopes that this offer is not just a techno-adolescent variation of the old line, "Come see my etchings." Just in case, make sure his parents are home when you visit.

Dear Ms. Motherboard,

I have been shopping for an inexpensive daisy-wheel printer for my Rainbow 100A and the TTX 1014 seems to fit the bill because it is compact and produces electronic typewriter quality. Furthermore, it has a built-in pin feed similar to my LA-50, RS-232 serial and Centronics-type parallel interfaces, and a price that's hard to beat (as low as \$365.00 in magazine ads).

Like so many beginners, I am not a technical wiz when it comes to computers and it is my understanding that DEC uses some odd-ball serial port signals in the Rainbow, and is somewhat closed mouth about pin assignments. So I wrote to TTX and asked them if the TTX 1014 is compatible with my Rainbow. They said, "Yes, the TTX 1014 is compatible with the DEC Rainbow 100's printer port, BUT, you must also have the XON/XOFF firmware with it."

My question is, What is XON/XOFF firmware, and does my Rainbow 100A have it? If it doesn't how do I get it? I just don't want my Rainbow to turn into charcoal grill when I plug in a non-compatible printer.

Firmly yours,
Beginner Buff Howie

Dear BBH,

Good news! Ms. Motherboard's technical experts tell her that the XON/XOFF protocol is built into the Rainbow, and can be found in the Set-Up mode. When you get into Set-Up, use the the Next Screen function key to move to the second screen, "PARAM SET", then use the right arrow to move to "AUTO XON/XOFF". If the number in that column is "1", it is already on; if it is "0", it is off; change it to "1" by using the up or down arrow. To retain the setting, type "Shift S". You will have then set the XON/XOFF setting to "on".

XON/XOFF is a communications protocol in which the printer and the computer establish a dialog. Because the computer can send characters to the printer much faster than the printer can print

them, the XON/XOFF protocol allows the printer to stop the computer from transmitting more characters until it is ready to print them.

Dear Ms. Motherboard,

My eyes are killing me. I am a programmer analyst, so I spend most of the day staring at a video screen. Is there anything I can do to avoid eye strain at a video terminal? I can barely see the movies on HBO after a hard day at the terminal.

-Blinded by the Light

Dear BBTL,

You may need to change the lighting in the room where you work. To minimize eyestrain, you need to prevent glare on the screen and reduce and even out the room's lighting level. First check the placement of the screen itself. The screen should not be in a position where sunlight falls directly on it. Place it where you won't have to directly face a window when you look at it. When you work with a VDT, the overall light level in the room should be lower than is typical for offices. You will need more light over your work area. Invest in an "architect's lamp" which can be adjusted at the head and arm. Place the lamp so that it illuminates your paperwork and keyboard. Experts recommend florescent lighting since it causes fewer shadows.

Finding perplexing parity errors? Questions related to computer etiquette and the effect of computers in your life should be sent to Ms. Motherboard care of the Newsletter Editor. Naturally, letters should be sent on tasteful buff micro-perf bond with matching envelopes, but dot matrix printing is acceptable.

=====

THE GRAPEVINE

. . .Borland may put out a Modula-2 compiler soon. . .Mark William's C in a Rainbow MS-DOS version should be out by the end of June (\$. . .Used printers are typically discounted as much as 60% over the original cost. . .if you can find a used HP Laserjet, it could be as low as \$. . .R Base: 5000 will not be available in a Rainbow version unless DEC comes to them and specifically requests it, according to MicroRim. . .dBase III for the Rainbow--it IS available--mentions that you need MS-DOS version 2.11 to run it. . .fortunately, not true. . .it runs fine on MS-DOS 2.05. . .Think twice before starting to change config.sys, especially if you have any files on the system already. . .you may get some very strange errors. . .If you were planning to switch your 100A motherboard for a 100B, you may have to forget it. . .apparently DEC has released an internal memo banning the TRUMP, or trade-up program, but check with your local DEC Service office--a few haven't gotten the message. . .Turbo-Pascal version 3.01A is being sent out. . .it apparently corrects problems in version 3.0. . .Framework will not be available for the Rainbow. . .nor will ThinkTank. . .DEC is threatening employees so they won't talk about a new computer expected to be announced before Christmas 1985. . .the new personal computer will not be compatible with the Rainbow. . .A lot of KayPro CP/M public domain software works on the Rainbow. . .WordStar 2000 commands are not the same as good old WordStar. . .AME-86 does not allow file transfers using Cross-Talk. . .it also does not work with any CP/M-80 programs. . .works well with WordStar. .

(c) 1985 by Caroline M. Mack

=====

=====

SHORT NOTES

--Was DEC right about office automation? Future Computing, a marketing research company, forecasts that in 1985, about 75% of corporate micros will be used as standalones. In 1984, nearly 80% were used as standalones. Why not make Rainbows attractive to both the standalone and the office automation market, DEC?

--Verbatim Disk Replacement Offer: Verbatim will replace any 3M, Maxell, or Dysan floppy disk which fails with a Verbatim disk for free. To take advantage of the offer, mail a failed 8", 5.25", or 3.5" disk, in its original envelope, along with your name, address, and brand and model of your computer system to: Verbatim Offer, P. O. Box 7306, Clinton, IA 52736. There is a limit of one refund per name or address. The offer is limited to the first 100,000 failed disks received, and expires October 1, 1985.

--The 4000 member Turbo User's Group (TUG) is a specialized group for users of Borland's Turbo Pascal. Membership in the national group entitles you to receive TUG Lines, their bi-monthly newsletter. The December 84/January 85 issue contained a number of articles related to Turbo Pascal, such as "TYPER: A Typewriter Emulator", "Four Lifesaving Utilities for CP/M", "UCSD Program Logic Conversion", and "8087 Support: That Much Faster?". The 36 page newsletter also has a New Products Section. Information on membership (\$20/year) is available from the Turbo User Group, P. O. Box 1510, Poughkeepsie, NY 12550.

--The Directory of Information Age Newsletters is a directory of more than 380 newsletters which specialize in personal computers, electronics, telecommunications, investments, office automation, and related subjects. Available for \$95 from Frank Communications Group, P. O. Box 144, Mount Vernon, NH 03057.

--Barry Folsom, manager of DEC's Rainbow group, has resigned. He has taken a position as vice-president of East Coast engineering with Sun Microsystems, of Mountain View, California. Folsom headed the product team which developed the Rainbow 100, 100+, 190, and the VT180. Folsom's resignation closely follows the resignation of Neil Rich, Manager of Tactical Marketing for the Rainbow. Rich left that post in January to take another position within DEC.

--Interactive Technology, Inc., which produces RDM for the PRO 300 series, has changed its address to 10700 SW Beaverton-Hillsdale Hwy., Suite 406, Beaverton, OR 97005, (503) 644-0111.

--Data Sources, Inc., plans to offer the Computer Industry Digest. Data Sources claims the semi-annual publication will contain abstracts of new product of all new product reviews as well as articles from 700 computer-related publications. Current mailings advertise the price for one year at \$112. More information is available from Data Source's Computer Industry Digest, 4th Floor, One Park Avenue, NY, NY 10016.

--Rainbow software is available by mail at substantial discounts from MEC Software, 1713 St. Mary's Avenue, Parkersburg, WV 26101, (304) 485-9126. MEC discounts range from 4% to 33% on an extensive list of Rainbow software. They also carry Univation hardware and other peripherals. Technical information is available during business hours. MEC accepts checks, credit cards, and purchase orders. Call or write for a catalog between the hours of 9 a.m. and 7 p.m., Monday through Friday, and 10 a.m. to 3 p.m. on Saturdays.

--Samna Corporation has removed copy protection from Word II and Word III, making it possible to copy the program onto hard disks and to make backups. Congratulations, Samna, for a sensible decision. . .

--Summa Technologies, which offers Freestyle (a text outliner similar to ThinkTank, and a word processor), is now offering site licenses. The cost of a basic license for Freestyle is \$7000. Summa will also, for an extra fee, allow employees to make copies of Freestyle for personal use. Summa intends to make site-licensing affordable even for small companies.

--Bulletin Board Users should watch out for a program called VDIR.COM. It is designed to trash your system when it runs, including your hard disk. It sends little messages telling you what it is doing. Once it is run, you can't be certain when or if it will reappear. It works like a "tape worm."

--DEC has made the specs of the All-In-One Office System available to encourage third party developers to link applications programs to All-In-One.

--Hard times for high tech in the stock market: Like many other high tech companies, including IBM, DEC's stock price has been dropping, and probably has not hit the bottom yet. DEC insiders (corporate officials, directors, and so on) have been selling their DEC stock. Insiders typically buy and sell well in advance of major changes in of companies' stock, giving an indication of the direction the stock will take. The stock took another dive recently after analyst Marc Schulman, of Hambrecht and Quist, Inc., estimated that DEC stock would have lowered earnings during fiscal 1985 and 1986.

--Despite expects drops in DEC's stock earnings, DEC continues to give Rainbows to colleges (the only way to get rid of them since they won't sell them in retail stores. . .?) It has now given micros to 50 colleges which received funds from the Department of Education's Fund for the Improvement of Postsecondary Education project, as well as to several universities who are apparently emphasizing networking. Recent donations include \$11 million (of a \$30 million networking project at Stevens Institute of Technology, and several million for a network at Syracuse University. Both projects are based on DEC's powerful new VAX 8600 computer.

=====

SOFTWARE AND HARDWARE

--EMUDEC is an interface which is designed to be lodged in an RS-232 cable between a Qume, Diablo, or Fujitsu daisy wheel printer and the computer. It intercepts LPT02 commands and translates them into Qume Sprint or compatible commands. No other software or hardware modifications are necessary. Using a Z80 processor, it provides a 2KB buffer, and sends return signals which emulate a DEC printer. Baud rates from 150 to 9600 are supported, as are choices in parity, stop bit, and word length. For more information, contact Techsolv Products, Ltd., 20 Queens Road, Reading, Berkshire, RG1 4AL, England.

--Softwright Systems offers Chesswright, a menu driven Chess game with seven skill levels from beginner to expert. Chesswright allows the user to play white or black with the computer, to play with another player, or to watch the computer play itself. It takes advantage of a graphics board and color monitor if available, but has an alternate screen display if they are not available. Chesswright requires MS-DOS and 128K of RAM, runs faster with more memory. Available for \$ from Softwright Systems, P.O. Box 3208, Durham, NC 27705; (919) 383-4441.

--Real Estate Microcomputer Systems, Inc. (REMS), offers Real Estate Investor II for the DEC Rainbow. Real Estate Investor II uses modern real estate analysis techniques to aid the user in making sound real estate investment decisions. Features include, 5 or 10 year analysis, partial year analysis upon acquisition, up to 10 different loans using 9 different types of loans, refinancing

capabilities, 100% or greater financing, Internal Rate of Return, Financial Management Rate of Return, variable parameters from year to year, automatic mathematic calculations, and full screen editing. Requires 192K RAM for MS-DOS version, 64K for CP/M. Available for \$ (with a 30 day money-back guarantee) from Real Estate Microcomputer Systems, Inc., 526 NW Second Street, Corvallis, OR 97330, (503) 757-8887.

--JEB Systems, Inc., has announced Marketing Management System, a comprehensive sales tracking and follow-up system. The heart of MMS is a database with pre-defined screens which store relevant information and provide space for memo information. A simple text editor allows the user to create, merge and print one page letters for quick response to sales prospects. The on-line Master File can be used with an automatic dialing function to facilitate automatic dialing. MMS' forecasting allows the user to summarize lead and sales activities to determine productivity, summarize cost and exposure data to determine cost effectiveness, summarize lead and sales activity by product, determine the average closing time for each product, quantify each individuals sales activity, and make a monthly forecast based on leads on file by product or salesperson. The program keeps records on sales follow-up activities. Using the database, labels, envelopes and cards can be produced in a number of formats. Also included are data import and export utilities, global change, help screens, record locking, interface with Lotus 1-2-3 and AthenaGraph for graphics, and interface with databases such as Datatrieve and DBaseII. The Marketing Management System is available on a number of different DEC systems. The cost for the DEC Rainbow and the DECmate II is \$ 3: for the PRO/350, \$ for the MicroVax, \$ for the Vax, \$ The form letter/Lotus 1-2-3 Interface option is available for \$, and the Lotus 1-2-3 Interface is \$ The MMS Prospector is available for the DEC Rainbow at \$ For more information contact JEB Systems, Inc., P. O. Box 7U, 37 Main Street, Franconia, NH 03580, (800) 821-1006.

--Soft Gold, Inc. has recently announced the Business BASIC Native-Code Compiler, which augments Applications BASIC, an extended BASIC interpreter designed for programming business applications. The Native-Code Compiler uses the same source code as Applications BASIC, but provides a run time system, which protects the code. Both are compatible with MAL/Basic Four's Business BASIC Level III. The interpreter is \$; the compiler/run time module is \$. Requires MS-DOS, but there is a CP/M version which offers fewer features. For information, contact Soft Gold, Inc., P. O. Box 2718, Newport Beach, CA 92663, (714) 476-3004.

--For those who prefer not to do their own customization of WordStar or who have LA-100 Printers (which need a special patch), MSRAINBD offers a patch which sets function, arrow, and numeric keypad keys to common WordStar commands. CP/M, WordStar, and 64K required. Available for \$ from Computec, PO Box 4445, San Luis Obispo, CA 93043, (805) 544-5685.

--Graphic M*IXS, the makers of FINGRAPH for the Pro 350, are now offering Speech*Maker for the Rainbow. Speech*Maker is a graphics program designed to make visual aids, such as transparencies and handouts, inexpensively. The program provides over 100 "pre-defined" layouts to simplify design, and allows design of custom layouts. It enables the user to draw lines, boxes, and circles, use 3 character fonts (regular or italicized) in four different sizes, save layouts with or without the information within. Requires MS-DOS with 256K RAM, does not require a plotter. Available for \$ plus \$ shipping from Graphic M*IXS, Inc., 960 Clocktower Drive, Springfield, IL 62704, (217) 793-0884.

--Sales Professional Database, designed for individuals and sales organizations, has been released by User Friendly Programmers. Using a menu system, it offers headings for customers, manufacturers, products, salesmen, expenses, mailing list, quotations,

orders, shipments, back orders, commissions, call reports, notes, to-do lists, and back-up prompts. Requires CP/M 86. Available for \$1095 from Jan Krieger, User Friendly Programmers, Inc., 501 Valley Brook Road, McMurray, PA 15317.

--The Bridge, a Virtual Microsystems add-on board, allows MS-DOS programs to be run on the Pro 350 and Pro 380. The bridge is software and a board which includes an 8086 processor and 256K bytes of RAM. It fully emulates IBM PC bit-mapped graphics. VM also offers products which enable the PDP 11/23, Micro-11, and Micro-VAX to run MS-DOS applications. Available from Virtual Microsystems, 2150 Shattuck Avenue, Suite 720, Berkeley, CA 94704, (415) 841-9594.

--A demo version of ATHENA/graph, which includes complete documentation, is now available for \$. ATHENA/graph is decision support graphics program available for the PRO 350 and PRO 380. Full package is \$, available from Ship Analytics, Inc., North Stonington Professional Center, P. O. Box 410, North Stonington, CT 06359, (203) 535-3092.

--C-CALC is available for the PRO 350 on the P/OS operating system. It is currently in beta testing for the Rainbow. Written in the C language, it offers menus, English-like commands, on-line help, an export/import routine, and ability to work with up to four parts of a spreadsheet or four spreadsheets simultaneously. C-CALC offers a dialup demonstration at 300/1200 baud. The demo number is (206) 828-3913; Username: DEMO; Current Password: DSD. The lines automatically detect baud rate. If the computer does not respond immediately, press the <return> every few seconds until it does. To get help, type the letter "H" at the prompt. The demo allows you to print a portion of the manual so that you will have the information you need to actually try C-CALC. C-CALC for the PRO is \$: for the Rainbow, \$. For information, contact DSD Corporation, Suite A, 10420 N.E. 37th Circle, P. O. Box 2669, Kirkland, WA 98033-0712, (206) 822-2252.

--PRO/TSX-Plus, an operating system recently announced by S&H Computing Systems, Inc., allows the single user Pro 350 to function as a multiuser, multitasking computer. PRO/TSX-Plus is a new version of the company's TSX-Plus operating system. The new system supports up to three users, and will support most RT-11 applications designed for the PDP-11 without modification. System features include spooled printing, concurrent program control, a facility to lock files, communications which utilize dial in support, logon security, a virtual debugger, interprogram messages, and improved I/O speed. The system is supposed to reduce by 20-40% the time the older system needed to compile DEC Fortran. Available for \$ from S&H Computing Systems, 1027 17th Avenue S., Nashville, TN 37212.

BOOKS

How to Buy Software. Alfred Glossbrenner. St. Martins Press, 1984. Paper, \$14.95. 648 pages. Excellent source of information, not only on how to buy software, but how to evaluate it.

American University Programs in Computer Science: Their Resources, Facilities, and Course Offerings. Edited by William W. Lau. 1984. GGL Educational Press. 210 pages, \$20, postpaid. Describes computer science programs at more than 150 United States colleges and universities. Available from GGL Educational Press, 2555 East Chapman, Suite 300, Fullerton, CA 92631.

LETTERS TO THE EDITOR

Dear PC SIG Steering Committee:

The June issue of the QA/PC/Graphics Newsletter just arrived today. I am completely appalled by the butchering job done on the newsletter by the DECUS staff in the name of commercialism. When my father hired a cabinet maker to build a black walnut serving cart, he drew up the plans and gave them to the cabinet maker to have the piece made. Until the cart arrived, he didn't know how much it would cost. He felt that if he had to ask, he could not afford it. Most of us cannot afford to order software or hardware in such a manner.

Several of the authors and I were at a Washington Area Rainbow User's Group Steering Committee meeting and agreed that it was very inappropriate for DECUS to change a copyrighted article before it is published. Since I have submitted articles in the past and have several that I am writing at the present time, I feel that it is only fair to the authors that DECUS publish the exact guidelines that we are to write under. To white out the prices makes for very silly reading. I do not like to be made a fool. I also remember a case where the names of DEC competitors were not allowed in an article. Most of the prices involved are available in the DECdirect catalog.

At the present time, most Rainbow owners feel that they have bought an "Osborne." DEC has abandoned them. The public magazines have abandoned them. If you call a third party software vendor, they will deny they have the product. With extreme effort [on the user's part] they will give the [necessary] information. Once you call back and say, "I read in ----- that I can buy ----- from you for \$----- and it is part number -----, you can generally get the product. For instance, Ashton Tate denies that there is a Rainbow version of dBase III if you call them. If you tell them it is part number x, for \$y, they will sell it. If DECUS abandons the Rainbow, Rainbow user's will have no where to turn.

I think that the commercialism policy needs to be revised. It is clearly a different issue when you are dealing with PCs than with VAXes.

At a minimum, an author's kit is needed to list the rules.

Art McClinton
Mitre Corporation

Editor's note: The letter above, and the two replies below, were taken from DCS, the DECUS Leadership Communication System.

From Don Golden, SIG Communications Committee:

Art, I do understand your comments. I was involved in the decision to white out, rather than reject, the entire issue. This decision was made in compliance with the DECUS commercialism policy. I hope the board will consider the plight of the PC users and PC article authors and find a way to exempt PC software vendors who are in the DEC stable (ie., you can buy their product off the shelf at a DEC computer store) from this highly restrictive policy. Somehow seeing a review of ChessWright's Chess floppy at \$XY.YY is greatly different from seeing some VAX Software DEM's offering at \$X,YYY.YY

From Barbara Maaskant, Chairman of the PC SIG

Thank you for your comments regarding the SIG newsletter and pricing. I have a copy of the present policy (Caroline Mack recently forwarded it), and I will be happy to send it. I agree with you that it greatly takes away from the submission and that the PC world is very different from that of the mainframes. From my own experience, price is often the second question asked about a piece of software, and a major means of evaluating the necessity of many features. With the volume of software PC owners have to choose from, it is an unnecessary hassle to have to go to another

source for price information. The majority of PC users don't do not get tied up in hardware bundled pricing, academic pricing, special deals, etc. I can understand why quoting a price for mainframe products can not easily apply to the majority of readers. . . . I do not feel the logic behind this policy carries over to the PC world.

I believe a lot of attention is being paid to new editor training and think the idea of an author's packet is excellent. I also think the issue of commercialism and what might be archaic policies should be reviewed by the [SIG] Communications Committee. I have forwarded your comments and mine to our newsletter editor and Communications Committee representative. My apologies for what I know had to be a very upsetting surprise.

7 24 Jul 85 23:14:03 (RECV'D) #0.25

From: Ken Kaplan on Fido #22 in Net #100 PCLUG, St_Louis.....MO
To: Caroline Mack on Fido #483 in Net #109

Subject: June DECUS SIG Newsletter
Congratulations on a job well done. I hope the word spreads fast now that we now have a central source for Rainbow information. If DECUS suggests you leave out the prices next time, suggest they find another editor. Who do they think they are hiding pricing information from?

Thanks for all the hard work,
Ken Kaplan
Chairman St Louis DECUS PCLUG
& Fido International Coordinator on Fido 1/0

* Via Node 100/10
* Via Fido 109/115

[Editor's Note: The letter above shows what a FIDO Bulletin Board Message looks like.]

=====

USER GROUP COLUMN

 Editor's Note: I will try to update this column in each issue. If you know of a DEC PC user's group that is not listed, please send me the name, address, telephone number and a contact person. Also, please let me know of any changes in the information here. Thanks. --ED

CALIFORNIA

 INLAND EMPIRE PC LUG

Chairman:
 Jan Snyder
 General Dynamics
 380 Veronica Street
 Upland, CT 91786
 (714) 620-7511, x 1118/1101

 SACRAMENTO VALLEY LUG
 RAINBOW SIG

Chairman:
 Robert Walraven
 1309 Notre Dame Drive
 Davis, CA 95616

 PACIFIC RAINBOW LOCAL USER'S
 GROUP

Co-chairman:
 Steve Jack
 1623 Granville, Apt. 10
 West Los Angeles, CA 90025
 (213) 207-8310

 SAN DIEGO AREA RAINBOW LOCAL
 USER'S GROUP

Chairman:
 Rick Eliopoulos
 Advanced Software Applications
 5258 Vickie Drive
 San Diego, CA 92109
 (619) 488-2116/5258

 SAN FRANCISCO BAY AREA DEC PC
 USER GROUP

P. O. Box 12561
 Northgate Station
 San Rafael, CA 94913-2561

Chairman:
 Dale W. Miller
 (415) 472-6531

 SANTA BARBARA AREA DEC PC LUG

Chairman:
 Rick Vincent
 253 Aspen Way
 Santa Barbara, CA 93111
 (805) 964-9744

 SILICON VALLEY DEC PC USER'S
 GROUP

Chairman:
 Seth Goldberg
 Stanford University
 P. O. Box 4349
 Stanford, CA 94305-4349
 (415) 854-3300, x 2874

 SOUTHERN CALIFORNIA PC LUG

Chairman:
 Bill C. Davis
 GoodGames
 1457 1/2 West 219th Street
 Torrance, CA 90501
 (213) 618-1083

 CONNECTICUT

HARTFORD RAINBOW USER'S GROUP
 P. O. Box 10387
 West Hartford, CT 06110

Chairman:
 Reginald Dionne
 (203) 725-6000, x 5248
 (203) 583-4816

 NEW HAVEN RAINBOW LUG

Chairman:
 William B. Leng
 Southern Connecticut State
 University
 501 Crescent Street
 New Haven, CT 06515
 (203) 397-4625

 NECRUG (Northeastern Connec-
 ticut Rainbow User's Group)

Chairman:
 Howard Roberts
 67 Route 6
 Andover, CT 06232
 (203) 486-3920 (days)

DELAWARE

 DUPONT RAINBOW USER'S GROUP

Laura A. Lindsay
 E. I. Dupont de Nemours & Co.,
 Inc.
 Information Systems Department
 6-904-2
 Wilmington, DE 19898
 (302) 774-4853

 DISTRICT OF COLUMBIA, MARYLAND
 AND VIRGINIA

 WASHINGTON AREA RAINBOW USER'S
 GROUP

Chairman:
 N. Jay Bassin
 9514 Midwood Road
 Silver Spring, MD 20910
 (301) 589-5318

For information, contact:
 Newsletter Editor:
 Caroline M. Mack
 6415 Adelphi Road
 University Park, MD 20782
 (301) 927-0108

 WASHINGTON AREA PRO USER'S
 GROUP

Sharon Weiss
 255 Congressional Lane
 Rockville, MD 20852
 (703) 692-1380

 FLORIDA

 TRI-COUNTY RAINBOW LUG

Bill Tabor
 Computer Products, Inc.
 2900 Gateway Drive
 Pompano Beach, FL 33069
 (305) 974-5500, x7258

 HAWAII

 HAWAII RAINBOW USER'S GROUP

(Apparently defunct)
 For information, contact:
 Professor Russell Yost
 U. of Hawaii
 Tropical Agriculture Department
 3190 Maile Way
 (St. John's 017)
 Honolulu, HI 96822

ILLINOIS

 CHICAGOLAND AREA PC/WPS LUG

For information, contact:
 Cheryl A. Celeste
 Monsanto
 9701 West Higgins Road
 Suite 500
 Rosemont, IL 60018
 (312) 823-9050

 MASSACHUSETTS

 Boston Computer Society
 DEC PERSONAL COMPUTER USER'S
 GROUP

Boston Computer Society
 One Center Plaza
 Boston, MA 02108
 BCS (617) 367-8080

Chairman:
 Karl Rosenberger
 Newsletter Editor:
 Framingham, MA 01701
 (617) 879-8307

 MINNESOTA

 MINNEAPOLIS-ST. PAUL PC LUG

Being formed. Contact
 John Vilandre
 University of Minnesota
 Dept. of Epidemiology
 Stadium Gate 27
 Minneapolis, MN 55455
 (612) 376-4066

 MISSOURI

 ST. LOUIS DEC PC LUG

Chairman:
 Ken Kaplan
 Data Research Associates
 9270 Olive Boulevard
 St. Louis, MO 63132-3276
 (314) 432-1100

 NEBRASKA

 OMAHA RAINBOW USER'S GROUP

Shirley Bohaty
 1343 Bel Aire Blvd.
 Wahoo, NE 68066
 (402) 433-4766

NEW HAMPSHIRE

SEACOAST PC LUG

Kevin Sperl
ACS, Inc.
P. O. Box 76
Center Strafford, NH 03815
(603) 664-9717

NEW MEXICO

DEC PC LOCAL USER'S GROUP OF
ALBUQUERQUE, NEW MEXICO

Chairman:
Stan Outler
7204 Oralee NE
Albuquerque, NM 87109

NEW YORK

LONG ISLAND LUG
PC SIG

Chairman:
Al (Fred) Scholldorf
Reuters, Ltd.
90 Davids Road
Hauppauge, NY 11788
(516) 435-7116

NEW YORK METRO DEC PC LUG

Co-Chairman:
Bob Bennett
DEC User's Group of Greater New
York
697 West End Avenue, #9B
New York, NY 10025

GREATER ROCHESTER AREA LUG
(PC SIG?)

Gary Griswold
B&G Associates
P. O. Box 81
Webster, NY 14580
(w) (716) 722-1723
(o) (716) 477-5664

NORTH CAROLINA

RESEARCH TRIANGLE LUG
PC SIG

Jack Brickley
P. O. Box 2713
Chapel Hill, NC 27515
(919) 929-7791

OHIO

CENTRAL OHIO PC LUG

Chairman:
Dana Klebes
Battelle Memorial Institute
505 King Avenue
Columbus, OH 43201
(614) 424-4947

Robert L. Lindsay
Lancaster City Schools
345 East Mulberry Street
Lancaster, OH 43130

PENNSYLVANIA

DELAWARE VALLEY DEC-PC USER
GROUP

Chairman:
Roland Spressart
RSP Engineers
89 Signal Hill Road
Holland, PA 18966
(215) 968-3494

ST. JOSEPH'S UNIVERSITY RAINBOW
USER'S GROUP

Chairman:
Dr. Val Herzfeld
5600 City Avenue
Philadelphia, PA 19131
(215) 899-7665

UNIVERSITY OF PENNSYLVANIA DEC
RAINBOW USER'S GROUP

Bill Gavelis
906 South 46th Street
Philadelphia, PA 19143

TENNESSEE

MIDDLE TENNESSEE DEC PC LUG

Chairman:
Dennis D. Knowles
Cumberland Associates
108 Brookhollow Drive
Old Hickory, TN 37138
(615) 754-9151

TEXAS

DALLAS RAINBOW USER'S GROUP

For information, contact:
Chairman:
Ken Kattner
Kadex Corporation
11311 North Central Expressway
Suite 300
Dallas, TX 75243
(214) 692-6080

DIGITAL SMALL COMPUTER USER'S
SOCIETY (DISCUS)

Dennis Forcier
P. O. Box 470521
Dallas, TX 75247

HOUSTON DEC PC LOCAL USER'S
GROUP

Chairman:
Allen Bartram
Houston Micros
12502 Millbanks
Houston, TX 77031
(713) 495-3168

or
9119 South Gressner
Suite 101
Houston, TX 77074
(713) 981-5107

UTAH

SALT LAKE'S RAINBOW USER'S
GROUP
(a SIG of the Salt Lake LUG)

J. R. Westmoreland
6748 Acom Road
Midvale, UT 84047
(801) 262-5251

VIRGINIA

RICHMOND USERS GROUP

Chairman:
Bill Myers
Department of Chemistry
University of Richmond
Richmond, VA 23173
(home) (804) 320-1500
(office) (804) 285-6321

WISCONSIN

NORTHEAST WISCONSIN LUG

Steve Peschke
Network System Design
300 North Main Street
3rd Floor
Oshkosh, WI 54901

GERMANY

GERMAN PC SIG

Dr. Otto Titze
Institut fur Kernphysik THD
TH Darmstadt
Schlossgartenstrasse 9
6100 Darmstadt
Germany
(Telephone) (0 6151) 16 33 23

EUROPE

EUROPEAN PC SIG

Chairman:
Paul Sawyer
School of Chemical Engineering
University of Bath
Bath, Avon, BA27AY
England

AUSTRALIA

DECUS AUSTRALIA PC SIG

Chapter Office
P. O. Box 384
Chatswood
New South Wales, 2067
Australia
Phone 412-5252

ISRAEL

Herzliya, Israel

COMPUSERVE

The VAXSIG has an active group
of DEC PC users. Compuserve
subscribers can type GO PCS-16
at the ! prompt and select
sub-area 6. A real time con-
ference is held every Wednesday
at 9:30 PM EST. Also has a
BB and software area.

ROG--ROBIN OWNER'S GROUP

 Chairman:
 John Comella
 2 Mockingbird Lane
 Maynard, MA 01754

AMERICAN BAR ASSOCIATION

 DEC User Group
 Lawrence Eisenberg
 (818) 788-0354

Kelly Frey
 Harwell Barr Martin and Stegall
 P. O. Box 2960
 Nashville, TN 37219-0960

David Sykes
 Duane, Morris and Heckcher
 One Franklin Plaza
 Philadelphia, PA 19102
 ABA CONFER #170

NATIONAL DECUS ADDRESS

 DECUS (Digital Equipment User's
 Society)
 Ann Foley
 249 Northboro Road (BP02)
 Marlboro, MA 01752
 (617) 480-3259

DEC RAINBOW ORIENTED BULLETIN BOARDS

Bulletin Boards are believed to be open 24 hours unless noted. Unless noted, all accept 300 and 1200 baud rates; some accept 2400 baud. This list of DEC-oriented bulletin boards may not be complete. If you know of other bulletin boards with a DEC PC orientation, contact the editor.

(201) 750-3748 DEC-WARE II 107/317 Iselin, NJ David Horowitz
 (203) 453-1089 Rainbow's End 110/210 Guilford, CT
 Matt Gertner (10pm - 6am)
 (212) 535-8924 DEC-WARE 107/79 New York, NY Matt Kanter
 (213) 204-2996 Rainbow Data 102/36 Culver City, CA
 Don Brauns (CHARGES: \$2.50 a month, minimum, 6 months)
 (213) 618-8454 Southbay RB 102/383 Torrance, CA Glenn Bowes
 (213) 644-1963 Rainbow Brite 102/3603 Hawthorne, Ca
 Bruce Headly
 (214) 392-1121 Big D Fido 19/ Dallas, TX Dennis Forcier
 (301) 484-2831 CLP-BBS 13/2 Pikesville, MD John Madill
 (301) 776-2300 The Beauty Board 109/601 Laurel, MD
 John Raum
 (305) 321-2369 Joe's Rainbow 18/46 Orlando, FL Joe Clayton
 (312) 759-5402 Illini Data 115/393 Bolingbrook, IL Rob Elliot
 (312) 331-2396 Microcomputing Forum 115/666 Markham, IL
 Andrew McElroy (by REQUEST)
 (314) 432-4129 DECUS Central 100/51 St. Louis, MO Ken Kaplan
 (314) 726-3448 Mike's Board 100/16 St. Louis, MO
 Mike Mellinger
 (314) 576-2743 PC LUG 100/22 St. Louis, MO Ken Kaplan
 (314) 234-1462 MDC KDC St. Louis, MO Ben Baker (5pm-8am)
 (314) 962-0395 DCA BBS 100/17 St. Louis, MO Jon Hitchman
 (315) 589-3981 Hitchhiker's Guide 107/23 Williamson, NY
 Fritz Howard (12am-6pm)
 (408) 274-1427 Vern's FIDO 10/13 San Jose, CA Vern Crawford
 (415) 864-1418 Fido's Board 10/1 San Francisco, CA
 Tom Jennings
 (415) 431-8088 Puppy's Board San Francisco, CA Tom Jennings
 (502) 425-9941 Scherer's Dublin, OH David Orr
 (602) 277-3530 Rainbow BBS 114/333 Phoenix, AZ Jim Kashner
 (10pm-4pm M-F; 12am-8am, weekends)
 (608) 233-8449 MidNet 11/90 Madison, WI Mike Mansfield
 (609) 429-6630 DEC-House 107/82 Cherry Hill, NJ
 Brian Sietz (12am-5:30pm)
 (616) 343-0996 Thieve's World Kalamazoo, MI Ian Schirado
 (617) 481-7147 WayStar 101/14 Marlboro, MA Kevin Porter
 (617) 787-3033 Midnight DEC 101/45 Boston, MA
 David Strickler (12am-3pm)
 (617) 721-1688 DEC-Line 101/202 Winchester, MA Bill MacNeil
 (Closed until 9/1/85)
 (617) 632-1861 Dave's FIDO 101/27 Gardner, MA David Rene
 (617) 874-4325 Dave's Annex 101/310 Westminster, MA
 David Rene
 (617) 646-3610 NECS 101/44 Arlington, MA Dave Mitton
 (617) 486-2285 RBG-ENG HootNet 101/367 Littleton, MA
 Bruce Gibson (Run by the Rainbow Engineering Group
 at DEC)
 (619) 488-2116 San Diego Rainbow LUG 102/350 San Diego, CA
 Rick Eliopoulos (7am-7pm, DECUS members by re-
 quest)
 (703) 671-0598 The Bear's Den 109/74 Falls Church, VA
 Kurt Reisler
 (703) 359-6549 Wash-A-RUG 109/483 Fairfax, VA
 Washington Area Rainbow User's Group--Kurt Reisler
 (714) 537-7355 Mike's Rainbow 102/313 Garden Grove, CA
 Mike Hamilton (RESTRICTED)
 (714) 964-0454 Medic 102/410 Huntington Beach, CA
 Phillip St. Erne, MD
 (717) 794-5268 Catt House Fido 13/489 Blue Ridge Summit, PA
 Bob Catt

(801) 262-5251 IMALUG RUG 15/1003 Midvale, UT
 J. R. Westmoreland
 (805) 643-0982 Oberon Systems 102/430 Ventura, CA
 Scott Johnson
 (806) 795-0102 DEC Domain 19/67 Lubbock, TX Alan Minchew
 (806) 742-5328 DEC Bronson 19/918 Lubbock, TX Bronson Johnson
 46-541-33-170 Day Rainbow 101/348 Karlstad, Sweden
 Conny Jonsson (300 baud, CCITT except during FIDO-
 NET)
 44-6-354-6480 FIDO UK1 101/4401 Newbury, England
 Ron Smallwood (300 baud, European modem proto-
 col: 9pm-6am; Bell 212A modem protocol: 9am-3pm)

Non-DEC FIDO:

(603) 924-9820 BYTEnetLISTINGS 101/104 Peterborough, NH George
 Bond

(non-Fido bulletin boards)

(201) 249-0691 CP/M-Net (tm) East Piscataway, NJ
 (505) 831-0205 ROBIN REBS Albuquerque, NM Elroy Gonzales
 (617) 467-4824 PDL (Scientific Applications Public Domain Soft-
 ware; to log onto system respond to prompt for
 "Username" and "Password" with "PDL")
 (617) 467-7437 DEC MARKET: Login: Type <Return> to start. At the
 @ prompt, type "LOG LGK.KERMIT KERMIT". The last
 word will not show on the screen.

TYPE/PITCH SAMPLES USING WORDSTAR PRINT CONTROLS

WORDSTAR VERSION 3.33 (CP/M-86) OR 3.31 (MS-DOS)
 INSTALLED FOR DEC RAINBOW 100+; LA-50 PRINTER

(CPI = CHARACTERS PER INCH)

^PQ	<pre> abcdefghijklmnopqrstuvwxyz (16.5 cpi) ABCDEFGHIJKLMNOPQRSTUVWXYZ 1234567890 </pre>
^PQ^PD	<pre> abcdefghijklmnopqrstuvwxyz (16.5 cpi double) ABCDEFGHIJKLMNOPQRSTUVWXYZ 1234567890 </pre>
^PA	<pre> abcdefghijklmnopqrstuvwxyz (elite [12 cpi]) ABCDEFGHIJKLMNOPQRSTUVWXYZ 1234567890 </pre>
^PA^PB	<pre> abcdefghijklmnopqrstuvwxyz (elite bold) ABCDEFGHIJKLMNOPQRSTUVWXYZ 1234567890 </pre>
^PA^PD	<pre> abcdefghijklmnopqrstuvwxyz (elite double) ABCDEFGHIJKLMNOPQRSTUVWXYZ 1234567890 </pre>
^PA^PY	<pre> abcdefghijklmnopqrstuvwxyz (elite enhanced) ABCDEFGHIJKLMNOPQRSTUVWXYZ 1234567890 </pre>
^PA^PY^PD	<pre> abcdefghijklmnopqrstuvwxyz (elite enhanced double) ABCDEFGHIJKLMNOPQRSTUVWXYZ 1234567890 </pre>
^PN	<pre> abcdefghijklmnopqrstuvwxyz (pica [10 cpi]) ABCDEFGHIJKLMNOPQRSTUVWXYZ [DEFAULT PITCH] 12334567890 </pre>
^PN^PB	<pre> abcdefghijklmnopqrstuvwxyz (pica bold) ABCDEFGHIJKLMNOPQRSTUVWXYZ 12334567890 </pre>
^PN^PD	<pre> abcdefghijklmnopqrstuvwxyz (pica double) ABCDEFGHIJKLMNOPQRSTUVWXYZ 12334567890 </pre>
^PN^PY	<pre> abcdefghijklmnopqrstuvwxyz (pica enhanced) ABCDEFGHIJKLMNOPQRSTUVWXYZ 1234567890 </pre>
^PN^PY^PD	<pre> abcdefghijklmnopqrstuvwxyz (pica enhanced double) ABCDEFGHIJKLMNOPQRSTUVWXYZ 1234567890 </pre>

```

^PR      abcdefghijklmnopqrstuvwxyz (8.25 cpi)
         ABCDEFGHIJKLMNOPQRSTUVWXYZ
         1234567890

^PR^PD   abcdefghijklmnopqrstuvwxyz (8.25 cpi double)
         ABCDEFGHIJKLMNOPQRSTUVWXYZ
         1234567890

^PE      abcdefghijklmnopqrstuvwxyz
         ABCDEFGHIJKLMNOPQRSTUVWXYZ
         1234567890
         (6 cpi)

^PE^PD   abcdefghijklmnopqrstuvwxyz
         ABCDEFGHIJKLMNOPQRSTUVWXYZ
         1234567890
         (6 cpi bold)

^PE^PB   abcdefghijklmnopqrstuvwxyz
         ABCDEFGHIJKLMNOPQRSTUVWXYZ
         1234567890
         (6 cpi double)

^PE^PY   abcdefghijklmnopqrstuvwxyz
         ABCDEFGHIJKLMNOPQRSTUVWXYZ
         1234567890
         (6 cpi enhanced)

^PE^PY^PD abcdefghijklmnopqrstuvwxyz
         ABCDEFGHIJKLMNOPQRSTUVWXYZ
         1234567890
         (6 cpi enhanced double)

^PW      abcdefghijklmnopqrstuvwxyz
         ABCDEFGHIJKLMNOPQRSTUVWXYZ
         1234567890
         (5 cpi)

^PW^PB   abcdefghijklmnopqrstuvwxyz
         ABCDEFGHIJKLMNOPQRSTUVWXYZ
         1234567890
         (5 cpi bold)

^PW^PD   abcdefghijklmnopqrstuvwxyz
         ABCDEFGHIJKLMNOPQRSTUVWXYZ
         1234567890
         (5 cpi double)

^PW^PY   abcdefghijklmnopqrstuvwxyz
         ABCDEFGHIJKLMNOPQRSTUVWXYZ
         1234567890
         (5 cpi enhanced)

^PW^PY^PD abcdefghijklmnopqrstuvwxyz
         ABCDEFGHIJKLMNOPQRSTUVWXYZ
         1234567890
         (5 cpi enhanced double)

```

PRINT CONTROL SUMMARY:

PRINT CONTROL	^PQ	^PA	^PN	^PR	^PE	^PW
HORIZONTAL PITCH (CPI)	16.5	12	10	8.25	6	5
LINE LENGTH (COLUMNS)	132	96	80	66	48	40

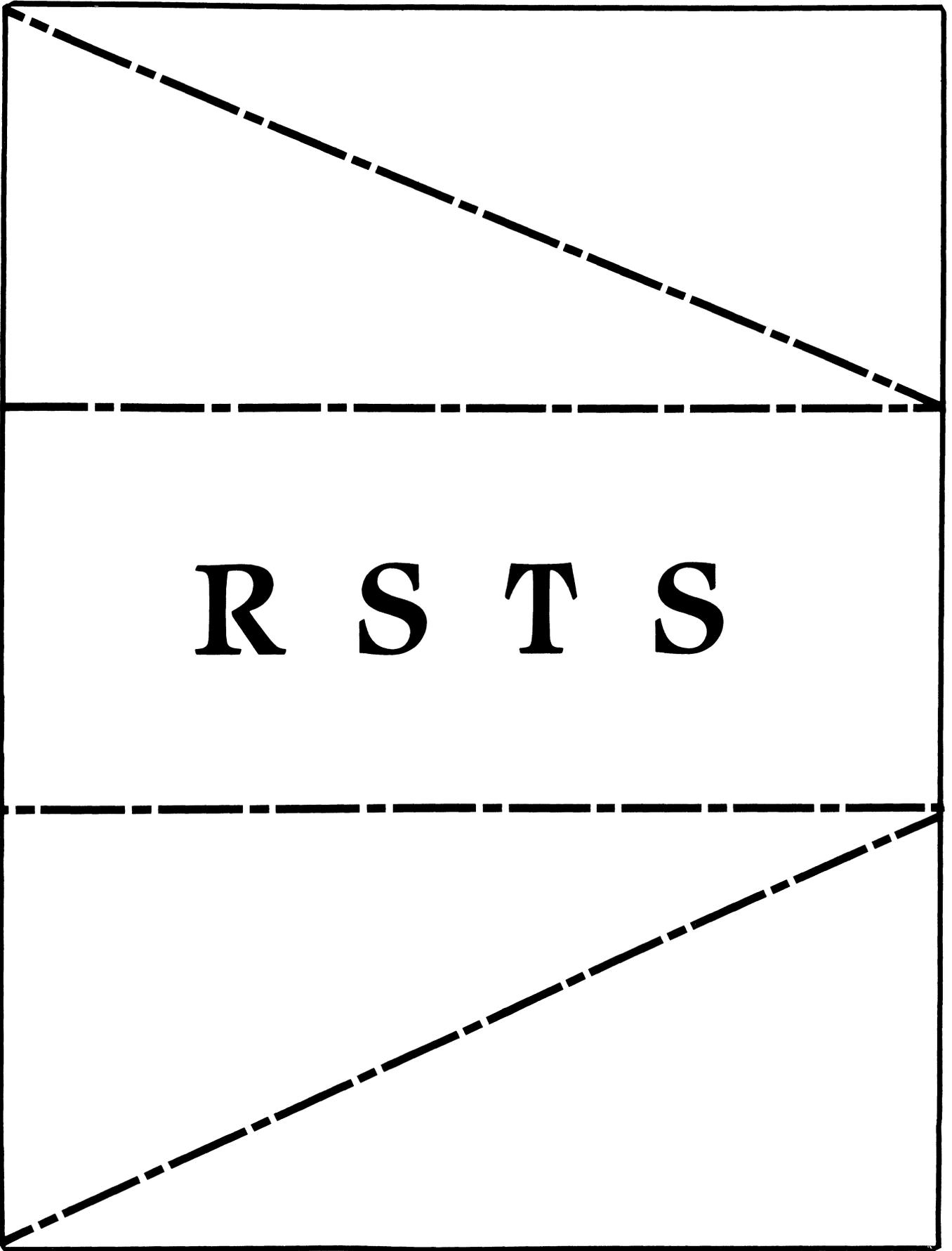
NOTES:

1. ^PY ACTS AS A TOGGLE TO START & STOP ENHANCED PRINTING. (ENHANCED PRINTING DOES NOT WORK WITH 16.5 OR 8.25 PITCHES).
2. ^PB ACTS AS A TOGGLE TO START AND STOP BOLD DENSITY. (BOLD DENSITY DOES NOT WORK WITH 16.5 OR 8.25 PITCHES).
3. BOLD AND ENHANCED DENSITY CANNOT BE USED AT THE SAME TIME.
4. ^PD ACTS AS A TOGGLE TO START & STOP DOUBLE STRIKING. (DOUBLE STRIKING WORKS WITH ALL PITCHES.)
5. TO RETURN TO DEFAULT PITCH FROM ^PQ, ^PW, ^PE OR ^PR, USE ^PA^PN; FROM ^PA, USE ^PN ONLY.
6. THESE PRINT CONTROLS AFFECT ONLY HORIZONTAL PITCH (CPI). DEFAULT VERTICAL PITCH IS 6 LINES PER INCH.

LA100 PRINTER SPECIAL FUNCTION SETTINGS

Function	Sequence 8-bit	7-bit	Function	Sequence 8-bit	7-bit
index	IND (132)	ESC D	SET PRINT QUALITY	CSI qual	* z
next line	NEL (133)	ESC E	draft	CSI 0	* z or
Horiz. Tab Set	HTS (136)	ESC H	letter	CSI 1	* z
Vert. Tab Set	VTS (138)	ESC Z		CSI 2	* z
partial line down	PLD (139)	ESC K	UNDERLINE		
partial line up	PLU (140)	ESC L	enable underline	CSI 4	m
reverse index	RI (141)	ESC M	disable underline	CSI 0	m
single shift 2	SS2 (142)	ESC N	SELECT FONT (0 - 4)	CSI 1	font m
single shift 3	SS3 (143)	ESC O	SELECT CHARACTER SET	CSI gset	cset
device control string	DCS (144)	ESC P	gset: (= G0 primary	UK = A	
control string introducer	CSI (155)	ESC [) = G1 primary	USASCII = B	
string terminator	ST (156)	ESC \	* = G2 primary	Finnish = 5	
operating system command	GSC (157)	ESC ^	+ = G3 primary	Norweg/Danish = 6	
private message	PM (158)	ESC _	, = G0 alternate	Swedish = 7	
application program command	APC (159)	ESC -	- = G1 alternate	German = K	
			. = G2 alternate	French Can. = 9	
			/ = G3 alternate	French = R	
				VT100 drawing = 0	
				Italian = Y	
				Spanish = Z	
				Multinational = <	
MODE SETTING COMMANDS			HORIZONTAL TAB CONTROL		
enable C1 transmit		ESC SP 6	Horiz. Tab Set	HTS (136)	ESC H
disable C1 transmit		ESC SP F	tab clear at active position		CSI 0 g
enable C1 receive		ESC SP 7	clear all tabs		CSI 2 g or
disable C1 receive		ESC SP 6			CSI 3 g or
set line feed new line mode		CSI 2 0 h	Set tabs		ESC 2
clear line feed new line mode		CSI 2 0 l			CSI t1 ; t2 ;
set auto wraparound mode		CSI ? 7 h			... ; tn u
clear auto wraparound mode		CSI ? 7 l	VERTICAL TAB CONTROL		
set horizontal pitch mode to font pitches		CSI ? 2 9 h	Vert. Tab Set	VTS (138)	ESC Z
set pitch mode to all pitches		CSI ? 2 9 l	set vertical tabs		CSI t1 ; t2 ;
POSITIONING COMMANDS			clear vertical tab		... ; tn u
vertical position absolute		CSI line d	clear all vertical tabs		CSI 1 g
horizontal position absolute		CSI col			CSI 4 g or
horizontal position relative		CSI delta-col a			ESC 4
cursor up		CSI delta-line A			
vertical position relative		CSI delta-line e			
index	IND (132)	ESC D			
next line	NEL (133)	ESC E			
partial line down	PLD (139)	ESC K			
partial line up	PLU (140)	ESC L			
reverse index	RI (141)	ESC M			
FORMS DEFINITION					
set page width alignment		CSI left ; width * s			
set margins		CSI left ; right s			
set lines per physical page		CSI lines t			
set top and bottom margins		CSI top ; bottom r			
FONT AND SIZE SELECTION					
Set horizontal pitch		CSI pitch w			
5		CSI 5 w			
6		CSI 6 w			
6.6		CSI 7 w			
8.25		CSI 8 w			
10		CSI 0 w or			
		CSI 1 w			
12		CSI 2 w			
13.2		CSI 3 w			
16.5		CSI 4 w			
Set vertical pitch		CSI pitch z			
2 lines per inch		CSI 4 z			
3 lines per inch		CSI 5 z			
4 lines per inch		CSI 6 z			
6 lines per inch		CSI 0 z or CSI 1 z			
8 lines per inch		CSI 2 z			
12 lines per inch		CSI 3 z			

	TITLE: DECUS U.S. CHAPTER PUBLICATIONS COMMERCIALISM POLICY	SECTION: I
	POLICIES & PROCEDURES	ISSUE DATE: 3/81
APPROVED BY U.S. EXEC. BOARD		REVISION:
 <p>POLICY FOR DECUS U.S. PUBLICATIONS</p> <p>In accordance with the U.S. Chapter's commercialism policy the following policy applies to the area of Publications. There are three specific areas which this Publications Policy will address.</p> <ol style="list-style-type: none"> Chapter-wide publications - (i.e. DECUSCOPE, Symposia Proceedings, etc.) will not include product descriptions of either Hardware or Software. SIGs, LJGs and Special Working Groups' Publications - may provide information relative to commercial products which area available for use on DEC systems provided the information is technical in nature. This definition can be further defined by remembering that the information should tell; <u>HOW</u> the product works; not <u>WHAT</u> it does. <p>A number of restrictions are placed on the above areas:</p> <ol style="list-style-type: none"> No job offerings may be published. No hardware product which directly conflicts with a DEC piece of hardware may be published (i.e. Disks, CTPPT, etc.) Digital products are exempt from the chapter-wide SIG publications restriction with the exception of pricing. The controlling group (i.e. Board, SIG Steering Committee, etc.) has the right to institute stricter restrictions for its publications. 		
PC-89		



R S T S



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

William Hobbs
ComManD. Inc.
Indianapolis, IN

Library Representative

Susan Abercrombie
Ventrex Laboratories Inc.
Portland, ME

DEC Counterpart

Joel Arker
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
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

Terry Medlin
Survey Sampling, Inc.
Westport, CT

Vice-Chairperson

Elizabeth Bailey
Tennessee Valley Authority
Muscle Shoals, AL

Symposium Coordinator

R.J. Hopp
Carlton Financial Comp.
South Bend, IN

Pre-Symposium Seminar Coordinator

Hans Jung
Associated Press
New York, NY

Communications Committee Representative

Kerry Wyckoff
LDS Church
Salt Lake City, UT

Newsletter Editor

Dominic J. DiNollo
Loral Electronics
Yonkers, NY

Store Coordinator

Bob Freeborn
Savin Corporation
Binghamton, NY

Session Note Editor

Dan L. Eisner
Perkin-Elmer Corp.
Garden Grove, CA

Librarian

Glenn Everhart
Mt. Holly, NJ

Campground Coordinator

Jerry Ethington
Prolifix Inc.
Frankfort, KY

DEC Counterparts

John Bennett
Digital Equipment Corporation
Maynard, MA

Dick Day
Digital Equipment Corporation
Nashua, NH

Working Group Coordinator

Ed Cetron
Center for Biomedical Design
Salt Lake City, VT

Working Group Chair

Evan Kudlajev
Philadelphia Electric Co.
Philadelphia, PA

RSX Group Chair Software Clinic Coordinator

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

Muti-Processors Working Group Coordinator

Bruce Mitchell
Machine Intelligence & Indus. Magic
Hudson, WI

Networks Working Group Coordinator

Mark Podany
Case Western Reserve University
Cleveland, OH

SRD Working Group Coordinator

Bob Turkelson
Goddard Space Flight Center
Greenbelt, MD

Accounting & Performance

Working Group Coordinator

Denny Walthers
American McGaw
Irvine, CA

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

From the Editor

Table of Contents

From the Editor	1
Letters to the Multitasker	2
The Bag of Tricks: MACRO-11	4
Sort V3.0 - Part I	5
Enhancements to RSX Phase IV Decnet	9
Decnet Changes for RSX-11M Systems Modified for Loadable Device Driver Data Bases	11
Professional 300 Series Communications - Part I	18
The QIO	35

To all reading the RSX Multitasker for the first time: Welcome. The Multitasker is the communication organ of the RSX SIG. It is used to convey technical as well as SIG business news to the RSX Community. Starting this month, the Multitasker will no longer be published as a separate newsletter. We have been combined with all other SIG newsletters into a single monthly volume. As long as submissions to the Multitasker continue, the monthly publishing schedule will be maintained. I encourage all users of RSX systems to contribute to the newsletter. Most everyone is doing something which would be beneficial to others. Share your knowledge with us.

In this issue, Ralph Stamerjohn starts a series of articles on the RSX Executive. The QIO is examined in the first article. The first of a three part article on SORT V3 for RSX appears in this issue. Bart Lederman will share his knowledge and experiences using and improving SORT. We will here from Bart in the near future on instruction execution timing. The Bag of Tricks: MACRO-11 continues. This month, part seven of Bruce Mitchell's column on MACRO-11 coding is presented. The remainder of the issue deals with communication and networking subjects. Carl Mickelson and Michael Mazzone tell us how they modified Decnet to suit their needs. If you own our use a PRO you will find the dissertation by Thomas Turano and Roman Pinsky very useful. The intricacies of PRO communication capabilities are explained.

On last word on contributions to the Multitasker. I can accept many types of media. A list of formats appears below. Please do not let this be the reason for not contributing. Special arrangements can be made and all media is returned.

Magnetic Tape:	1600/6250 BPI	BRU, PIP, FLX, VMS BACKUP
Floppy Disk:	8 Inch	RX01/RX02 ODS-1 or ODS-2
	5 Inch	RX50

TU58 Cart

Dominic DiNollo
Loral Electronic Systems
Engineering Computer Center
Ridge Hill
Yonkers, New York 10710
(914) 968-2500 ext 2210

RSX MULTITASKER

Letters to the Multitasker

Dept of Physics
University of Southampton
Southampton SO9 5NH

The Editor, Multitasker
C/O DECUS
1 Iron Way
Marlboro, MA 01752 USA

Dear Sir:

I followed last years letters on how to place a double quote inside a character string while using IND with interest. Recently I have had the problem of putting a single quote (i.e. the character used for substitution control) inside a string. My application, a F77 program generator requires assignment to character variables. FORTRAN is not happy with double quotes and insists on single ones. The solution to this problem turns out to be very simple. If a .SETS is executed before one enables substitution then the instruction will work if the right hand side contains a '.

For example,

```
.sets sq "!"           ;Puts a single quote in sq
.enable substitution
;
; lots of code
;
.data 2 character*20 prompt
.data 2 prompt='sq'This is the prompt'sq'
;
;
```

results in the following FORTRAN

```
character*20 prompt
prompt='This is the prompt'
```

It was not obvious to me from any of the documentation that this would work and I was quite surprised when it did. I hope that other people will find this useful.

Dr. J.N Carter

RSX-2

RSX MULTITASKER

Allied/Bendix
Aerospace Division
P.O. Box 1159
Kansas City, MO 64141

The Editor, Multitasker
C/O DECUS
1 Iron Way
Marlboro, MA 01752 USA

Dear Sir:

Does anyone know the new features of RSX-11M-Plus V3.0 ? Would it be possible to have a brief article describing these new features? Rumor has it that Virtual Disks will be supported ... What about a Memory Disk? Named directory support and logical names ... will these new features be supported by Exec calls? How much like POS will V3.0 be ... What are the differences?

As you can see, I have many questions. The Spring '85 DECUS Symposium alluded to some of these items, but most were not directly addressed. V3.0 test sites ... open up!

Sincerely,

Dave O'Larte

RSX-3

The Bag of Tricks: MACRO-11

Bruce R. Mitchell
Machine Intelligence and Industrial Magic
PO Box 601
Hudson, WI 54016

This column covers MACRO-11 bag-of-tricks routines, as stated in previous issues of the Multi-Tasker. It appears as space permits. All MACRO programmers are encouraged to submit their favorite routines to the Multi-Tasker so that these useful, interesting, or just plain bizarre tricks can be put out before the SIG in general for the admiration and edification of all.

In this month's column, we again have a bit of code which is at quite trivial and yet very useful.

When a task is installed, it can default to either the Taskbuilt name (TASK= directive) or a name given at install time (Install /TASK= switch). In many cases - the majority of cases, one may reasonably assume - the task's running taskname will not matter.

However, there are cases where the task's running taskname DOES matter. A few of these cases are where the task will be receiving send data packets, acting as a DECnet task, or where multiple copies are impermissible for some reason.

The GTSK\$ directive allows a task to capture information about itself and the system on which it is running. To a task which is running name critical, the offset G.TSTN in the GTSK\$ buffer are interesting; they return the task's running name in Radix-50. A task can check these for a correct value, then error out or perform other error actions if the expected values are not found. As we saw last month, it could just as well take other action - it could send it to cooperating tasks, to inform them of what level of interchange protocol it can understand, etc. No doubt other uses are possible.

An example of the entry point of a task which captures this information follows. Note that the information supplied in the buffer is Radix-50 and may have to be converted to ASCII to be useful.

```
; Obtain information about self from the system
GTSK$ GTSK$ TSKBUF ; Get information on self
```

```
TSKBUF: .BLKW 16. ; Storage area for task parameters
```

```
.PAGE
.SBTTL NTRYPT Entry Point
```

```
; There is no point (at this time) in more than one BARFOO per system,
; since tasks address each other by installed name, so this ensures
; that only one copy, with the name BARFOO, will run on the system.
```

```
NTRYPT: DIR$ #GSINFO ; Get information about self
CMP TSKBUF+G.TSTN, #RBAR ; Is first half of name "BAR"?
BNE 10$ ; If not, prepare to exit

CMP TSKBUF+G.TSTN+2, #RFOO ; Is second half of name "FOO"?
BEQ 20$ ; If so, jump into time checking

; This task's name is not BARFOO; do something about it
10$: ... ; task takes appropriate error action
```

```
; This task's name is indeed BARFOO; proceed with initialization
```

```
20$: ... ; task continues ...
```

```
.END NTRYPT
```

SORT V3.0 - Part I

B.Z. Lederman
Bankers Trust Co.
P.O. Box 318, Church Street Station
New York, N.Y. 10015

My work is in an area where correctly operating software is very important, so I did considerable testing of V3.0 before passing it on to the applications people. I found that for sorting very short files, V3.0 was slower (probably due to the more complicated command parsing code), but for long files (hundreds or thousands of blocks long) it was slightly faster. A very big improvement is in

the sort specification language. When I first heard of V3.0 at the spring 1984 symposium (Cincinnati), the DEC speaker polled the audience and I believe there was at most one other person besides myself who had ever used the V2.0 specification file, and I've never met anyone else before or since who has used it; but with V3.0 use of the specification files has been simplified by orders of magnitude, and this, plus the fact that you can now merge multiple input files, means that V3.0 has a wider range of applications than V2.0 had. V3.0 also has at least one V2.0 problem fixed: it will correctly assign input and output files to your local SY: when it is not the system SY:, whereas with V2.0 you always had to put in explicit devices on the input and output file names. A further convenience is that you do not have to put in a /FO switch: the output file will default to the same file organization as the input file.

This is not to say that V3.0 is perfect. I will describe the few problems I have seen.

If you do have V2.0 specification files and use the supplied utility TRN to translate them to V3.0 specification files, explicit comparisons to numeric literals might not be translated properly. In my case, comparisons with an octal constant were always translated into a comparison with a hexadecimal constant of zero. This is easy to fix by hand, but you should watch out for it. The default to hexadecimal is also a bit annoying.

A more serious error is that comparisons between a binary data field and a numeric constant fail unless the constant is expressed in hexadecimal. The failure is accompanied by an error message which states that there is an invalid specification, but does not give the line number. This will occur with a command file such as: (only part of the file shown)

```
/PROCESS=RECORD
/FIELD=(NAME:IDENT, POSITION:1, SIZE:2, BINARY)
/CONDITION=(NAME:TASK_REC, TEST:(IDENT EQ %X0003))
/INCLUDE=(CONDITION:TASK_REC,
KEY= . . . .
```

This specification file works correctly, but changing the constant from %X0003 to %D0003 (a decimal constant) will cause the sort to fail. I have not yet received an answer from DEC from my SPR on this.

The manual shows fields in specification files separated with equals signs ("="), but the translation utility uses colons (":"): apparently, both are acceptable, but only one is shown in the documentation.

The manual shows comparisons between fields, but not between a field and a numeric literal. It turns out that the format to enter numeric literals is the same as for the /PAD command, or as shown

in the example given above.

A minor point is that Sort does not truncate output files. If you normally sort an entire file, you will never see this problem. If you do use specification files to reject unwanted records, or reduce the size or re-format the output record, then the output file may be smaller than the input file, but it will have the same number of blocks allocated to it as were in the input file. Pre-allocating blocks in a file makes writing to the file faster, so that is O.K., and you can always use PIP to truncate the output file to its actual size. It does mean that if your output file is very much smaller than your input file, you will want to truncate the file to avoid wasting disk space.

A more interesting set of circumstances was uncovered when I tried to improve Sort performance through the use of resident libraries. I have found that other tasks were improved when linked to the RMS resident libraries (rather than use disk overlays), but Sort-11, as distributed, does not use any of the resident library options. If nothing else is gained, at least there are fewer disk overlay reads, and this is useful on moderate to heavily loaded systems. In some cases, use of the resident libraries will increase the amount of task space that Sort can use as its work area, and that also will improve performance. I found that it was very easy to link Sort to the RMS resident library, and more particularly to the Supervisor mode resident library, with a considerable reduction in system load. The actual sorts did not run significantly faster when tested on an unloaded system (PDP-11/70 with RP06 disks), but when I placed a constant and fairly heavy disk and CPU load on the system, the improvement was considerable. In some cases, the increased internal working space allowed sorts to be done entirely in memory, where the distributed version of Sort had to use work files (Sort V3.0 also has the advantage of optionally printing out a considerable amount of information on its internal state during the sort, and the amount of resources were used). An incidental benefit is that the task image itself is smaller when it does not have all of the RMS code in it, so it occupies less space on disk (nice if you have small disks).

An even further increase in performance was obtained by building Sort as an I-and-D space, non-overlaid task: all of the sorts worked, but we found that if the task image went over a certain size, evidence was obtained that it might not always work, as it was apparently using signed arithmetic in calculating its data area, and it could not cope with a work space greater than 32Kwords. I do not think it appropriate to repeat all of the SPR response to these tests, except that DEC might re-consider linking to the RMS resident libraries. The version of Sort-11 V3.0 that we have been using in production for several months is linked to the Supervisor mode RMS resident library. We would prefer to go to the non-overlaid I-and-D space version, but a bug in VMR will not allow it to be installed (it thinks too many APR's are being used), and we are still a little worried about its performance, even

though it passed all of our tests.

Incidentally, if you are still using Sort-11 V2.0 and have RMS V2.0, you can link to the RMS resident libraries, especially the Supervisor mode library, and obtain similar improvements, and it can also be built non-overlaid. V2.0 cannot be built as an I-and-D space task due to poor code management: there is data mixed into instruction PSECTs, and other problems. I do not know if V2.0 can be linked to the old RMS V1.8 resident library.

Sort-11 V3.0 comes with provisions for building a Sort resident library. This might be useful if you want to call Sort routines from within a program, and it looks as if it will be easier to use than was V2.0, but we haven't tried it. The library itself appears to be very poorly organized. Sort-11 is actually two products, Sort and Merge. Although they appear to share most of their code (which would be reasonable), nearly every module in the resident library is duplicated: once for Sort and once for Merge, which seems absurd. The resulting library is a memory resident overlaid monster, and I cannot believe that both products which are so similar are so totally different internally. The library might pay off if you could link Sort itself to it, and there is a lot of the Sort-11 task code in the library; but DEC does not supply a build file to link the Sort-11 task to the Sort-11 resident library, and I was not successful in building a library or a task where this could be done. This would have been useful for situations where many sorts are being done at the same time, as physical memory would be saved, but fortunately (for us) we don't have a shortage of memory.

We also use PRO-350s here, and I found that there is no real sort utility for P/OS. (There is sort of a callable utility, but nothing you can get from DCL or in the tool kit.) Both versions of Sort-11 (V2.0 and V3.0) will build and run well on the PRO with the RMS resident library, but I don't know how you would go about obtaining a license for this setup.

Editor's Note

This is the first part of a three part article on SORT V3.0 by Mr. Lederman

Enhancements to RSX Phase IV DECNET

Michael E. Mazzoni
Process Control Systems
1300 S. Calhoun Road
Brookfield, WI 53005

DEC distributes several undocumented enhancements to DECNET PHASE IV for RSX-11M/M+ with the standard kits. These enhancements are distinct from the unsupported software distributed in UFD [200,200] of the network kits. Three enhancements that I have succeeded in using are:

- o Snapshot copy of an NTD display to a file
- o Network command terminal and network command terminal host support
- o Area routing

NTD

The DECNET Node Task Display (NTD), as built by the DEC standard NETGEN procedure, will give a permanent copy of the output only if it is run on a hard copy terminal.

DEC has provided a more elegant solution to the permanent copy problem. In the command file [x,24]NTDBLD.COMD (created by NETGEN) there are instructions for enabling a snapshot dump of an NTD display on a video terminal to the file NTD.SCR. The modification involves uncommenting one line, and commenting out two lines in the command file. The instructions in the command file are quite clear, so they need not be repeated here. Total time to edit and re-taskbuild should be about 5 minutes.

The new NTD is significantly larger than the standard NTD, so when it is installed RSX warns that this task which overmaps the I/O page. This warning can be ignored.

Network Command Terminal

The Network Command Terminal, or NCT, seems to be an as yet unimplemented feature of DECNET Phase IV. Several references are made to this animal in the RSX DECNET manuals, but there is no specific list of its capabilities or how to use it.

RSX MULTITASKER

To make a NETGEN create NCT, edit the file [137,10]DECPRM.COMD and set the symbol \$DNCT to true.

If NCT is included in the NETGEN, a device RT: and DECNET processes RTH and NCT are built.

RTH is the remote command terminal host support process. It resides on the user's local node and allows users on other nodes to logically connect their terminals to the local system through the network command terminal facility.

NCT is the network command terminal server. It resides on the user's local node, and allows the user's terminal to be logically connected to any other node in the network which has the network command terminal facility.

An RSX task to use NCT is not included with DECNET and is presumably not yet available. VMS, though, seems to have more hooks in place.

When a user on a Phase IV VMS node logs into an RSX node with these added features, that user appears on device RT: instead of device HT:. Unfortunately an RT: terminal device just doesn't comprehend what to do about escape sequences, and so video terminal applications from VMS to RSX are hopelessly mutilated. If you thought that HT: was bad, try an RT:. Some notes I have from an M+ v3.0 preview seminar say that RT: problems won't be fixed in that release.

So, the NCT for RSX is going to need a lot more work before it is useful.

Area Routing

Area routing is the way DEC chose to get around the 1023 node limit for earlier DECNET versions. A PHASE IV network can consist of up to 63 areas, each with 1023 nodes. Special nodes, area routers, connect the different areas together.

A discussion of why and where to use area routing is beyond the scope of this writeup. Of all the reasons, having to add a 1024th node to an existing network is probably at the bottom of the list. I must leave it up to the interested reader to investigate if multiple areas are for him. The point is if you want it for RSX, it's there.

Although multiple areas was only announced for VMS v4.0 nodes, it was included in the 1983 RSX release of DECNET PHASE IV but not documented. And it DOES work.

RSX MULTITASKER

To make a NETGEN support area routing, edit the file [137,10]DECPRM.COMD and set the symbol \$DXMAS to true. During NETGEN you will be asked if you want to make the node an area router. If you answer yes, the area routing task RCP2.. is built (RCP1.. is the normal routing task).

There is only one caution. RSX nodes are traditionally identified by a number from 1 to 1023. For area routing, the numbering scheme is AA.NNNN where A is the area number, NNNN is the node number in that area. All RSX nodes, routing or non-routing, connecting to a network with an RSX area router, must be re-NETGEN'ed to incorporate support for the new numbering scheme.

DECnet Changes for RSX11M Systems Modified for Loadable Device Driver Data Bases

Carl T. Mickelson
Goodyear Aerospace Corporation
Akron, Ohio 44315

This article presents the revisions needed to install a DECnet V4.0 End-Node distribution kit on an RSX11M system operating with the changes described in the paper "Loadable Device Driver Data Bases in RSX11M SYSGEN", presented at the Spring 1985 DECUS U.S. Chapter Symposium at New Orleans, May 1985. This information complements the change files submitted to the RSX SIG tape at the symposium.

Since the changes presented at New Orleans remove the terminal data base from the Executive task image, and load them into pool only when the terminal driver is loaded by VMR, some DECnet programs will no longer be able to find the console terminal UCB (.TTO) or SCB (\$TTO) at link time. As a result, changes have to be installed into the programs to dynamically locate the required data structures at run time, and to reference them with PC relative addressing rather than by immediate addresses.

The command files that follow are designed to apply the required patches to the DECnet programs that access the console terminal (TTO) device data base in the RSX11M Executive.

File: DECUSDNET.COMD

```
.ENABLE SUBSTITUTION  
.SETS SAVUIC "'<UIC>'"
```

```

;
; This command procedure applies a set of patches to DECnet V4.0,
; necessary to operate the network software on systems that have been
; modified in accordance with the modifications documented in the
; paper "Loadable Device Driver Data Bases in RSX11M Sysgen",
; presented at the Spring 1985 DECUS Symposium at New Orleans.
;
; It is assumed by this command procedure that the DECnet distribution
; has already been copied to the NETGEN volume by the PREGEN procedure
; as described in the DECnet installation manual, and that the DECnet
; distribution kit has been updated with the DECnet saveset from the
; RSX11M V4.1 Update E distribution.
;
.ASKS [0:5:"'<SYSDEV>':" ] KIT What device contains the NETGEN disc
.IF KIT EQ "" .SETS KIT "'<SYSDEV>':"
ASN 'KIT'=KIT0:
.ASKS [0:5:"'<SYSDEV>':" ] MOD What dev contains the DECnet mod kit
.IF MOD EQ "" .SETS MOD "'<SYSDEV>':"
ASN 'MOD'=MDO:
.ASKS [0:9:"'SAVUIC'" ] UIC What [UIC] contains the DECnet mod kit
.IF UIC EQ "" .SETS UIC "'SAVUIC'"
SET /UIC='UIC'
.SETF DOTTO
;
; The TT0REFMOD.CMD file is used to apply some changes to the
; DECnet programs NTINIT and NTL to remove link time references
; to the console terminal UCB and SCB symbols .TTO and $TTO,
; since these are no longer present in the RSX11M symbol table.
; This procedure creates a new module FN$TTO and adds it to
; NETLIB.OLB to allow the DECnet programs to find and define
; the addresses of the required console terminal data structures.
; The starting and referencing modules of both NTINIT and NTL
; are patched to include a call to FN$TTO and to eliminate
; immediate references to the two symbols (.TTO and $TTO).
;
.ASK DOTTO Do the TT0REFMOD patches now
;
.IFT DOTTO @MDO:TT0REFMOD
SET /UIC='SAVUIC'
ASN =KIT0:
ASN =MDO:

```

File: TT0REFMOD.CMD

```

.ENABLE SUBSTITUTION
.SETS MODUIC "'<UIC>'"
.;
.; Install required utilities
.;
.SETF INSLBR
.IFNINS ...LBR .SETT INSLBR
.IFT INSLBR PRV INS $LBR

```

```

.SETF INSMAC
.IFNINS ...MAC .SETT INSMAC
.IFT INSMAC PRV INS $MAC
.SETF INSPAT
.IFNINS ...PAT .SETT INSPAT
.IFT INSPAT PRV INS $PAT
.SETF INSPIP
.IFNINS ...PIP .SETT INSPIP
.IFT INSPIP PRV INS $PIP
.SETF INSSLP
.IFNINS ...SLP .SETT INSSLP
.IFT INSSLP PRV INS $SLP
.;
.; Delete files that will be rebuilt
.;
PIP MDO:TTODEL.CMD;*/DE/NM
;
.ENABLE DATA #0
MDO:FNDTTO.*;*/DE/NM
MDO:NTINIT.*;*/DE/NM
MDO:NTLROO.*;*/DE/NM
MDO:NTLGET.*;*/DE/NM
MDO:TTOASM.CMD;*/DE/NM
MDO:TTOALBR.CMD;*/DE/NM
MDO:TTOPAT.CMD;*/DE/NM
MDO:NETCOM.CMD;*/DE/NM
MDO:NETCOM.COR;*/DE/NM
.DISABLE DATA #0
.CLOSE #0
PIP @MDO:TTODEL
.;
.; Create source file for console data structure locator
.;
.OPEN #0 MDO:FNDTTO.MAC
.ENABLE DATA #0
.TITLE FNDTTO - Find Console Terminal Data Structures
.IDENT /VA4.00/
.PSECT TTOADR
.TTO:: .WORD 0 ; Console Terminal UCB Address
$TTO:: .WORD 0 ; Console Terminal SCB Address
.PSECT
FN$TTO::
MOV RO,-(SP) ; Save RO
MOV # $DEVHD,RO ; Point at device data structures
10$:
MOV (RO),RO ; Link to next device DCB
CMP # "TT,D.NAM(RO) ; Is it the TT DCB?
BNE 10$ ; If not, get next DCB address
MOV D.UCB(RO),RO ; Get TTO UCB address
MOV RO,.TTO ; Save UCB address in global
MOV U.SCB(RO),$TTO ; Save SCB address in global
MOV (SP)+,RO ; Restore RO
RTS PC ; Return

```

```

        .END
        .DISABLE DATA #0
        .CLOSE #0
        .;
        .; Create patch source file for NTINIT
        .;
        .OPEN #0 MDO:NTINIT.PAT
        .ENABLE DATA #0
        .TITLE NTINIT - Eliminate Link Time Refs to .TTO and $TTO
        .IDENT /VA4.00/
        .PSECT
$$$=.
        ; Set reference symbol

.=$$$+0
        JSR PC,PAT001 ; Call first patch
        ; Point to starting instruction

.=$$$+70
        MOV .TTO,-(SP) ; Change to PC relative reference
        ; Point to .TTO reference

.=$$$+100
        CMP $TTO,@(SP)+ ; Change to PC relative reference
        ; Point to $TTO reference

        .PSECT PATCHS ; Patch code PSECT
PAT001:
        JSR PC, FN$TTO ; Locate the console data structures
        MOV (SP), -(SP) ; Copy return address
        MOV #17, 2(SP) ; Simulate original starting instruction
        RTS PC
        .END
        .DISABLE DATA #0
        .CLOSE #0
        .;
        .; Create patch source file for NTLROO root module
        .;
        .OPEN #0 MDO:NTLROO.PAT
        .ENABLE DATA #0
        .TITLE NTLROO - Eliminate Link Time Refs to .TTO and $TTO
        .IDENT /VA4.00/
        .PSECT
$$$=.
        ; Set reference symbol

.=$$$+0
        JSR PC,PAT001 ; Call first patch
        ; Point to starting instruction

        .PSECT PATCHS ; Patch code PSECT
PAT001:
        JSR PC, FN$TTO ; Locate the console data structures
        JSR PC, $NLINI ; Replace original starting instruction
        RTS PC
        .END
        .DISABLE DATA #0
        .CLOSE #0
        .;

```

```

        .; Create patch source file for NTLGET module
        .;
        .OPEN #0 MDO:NTLGET.PAT
        .ENABLE DATA #0
        .TITLE NTLGET - Eliminate Link Time Refs to .TTO and $TTO
        .IDENT /VA4.00/
        .PSECT
$$$=.
        ; Set reference symbol

.=$$$+10
        MOV .TTO,-(SP) ; Change to PC relative reference
        ; Point to .TTO reference

.=$$$+20
        CMP $TTO,@(SP)+ ; Change to PC relative reference
        ; Point to $TTO reference

        .END
        .DISABLE DATA #0
        .CLOSE #0
        .;
        .; Create assembly command file for patch sources
        .;
        .OPEN #0 MDO:TTOASM.CMD
        .ENABLE DATA #0
        MDO:FNDTTO,FNDTTO=MDO:FNDTTO
        MDO:NTINIT.POB,NTINIT=MDO:NTINIT.PAT
        MDO:NTLROO.POB,NTLROO=MDO:NTLROO.PAT
        MDO:NTLGET.POB,NTLGET=MDO:NTLGET.PAT
        .DISABLE DATA #0
        .CLOSE #0
        .;
        .; Assemble patch sources
        .;
        MAC @MDO:TTOASM
        .;
        .; Preserve original object libraries
        .;
        SET /UIC=[132,24]
        .TESTFILE KTO:NTI11MOLB.VGN
        .IF <FILERR> EQ 1 .GOTO CNT1
        PIP KTO:NTI11MOLB.VGN;*/RE=KTO:NTI11M.OLB;*
        PIP KTO:NTI11M.OLB=KTO:NTI11MOLB.VGN
        .CNT1:
        .TESTFILE KTO:NTLOLB.VGN
        .IF <FILERR> EQ 1 .GOTO CNT2
        PIP KTO:NTLOLB.VGN;*/RE=KTO:NTL.OLB;*
        PIP KTO:NTL.OLB=KTO:NTLOLB.VGN
        .CNT2:
        SET /UIC=[130,24]
        .TESTFILE KTO:NETLIBOLB.VGN
        .IF <FILERR> EQ 1 .GOTO CNT3
        PIP KTO:NETLIBOLB.VGN;*/RE=KTO:NETLIB.OLB;*
        PIP KTO:NETLIB.OLB=KTO:NETLIBOLB.VGN
        .CNT3:

```

RSX MULTITASKER

```

SET /UIC=[137,10]
.TESTFILE KTO:NETGENCLB.VGN
.IF <FILERR> EQ 1 .GOTO CNT4
PIP KTO:NETGENCLB.VGN;*/RE=KTO:NETGEN.CLB;*
PIP KTO:NETGEN.CLB=KTO:NETGENCLB.VGN
.CNT4:
SET /UIC='MODUIC'
.;
.; Create librarian module extraction command file
.;
.;
.OPEN #0 MDO:TTOLBR.CMD
.ENABLE DATA #0
MDO:NTINIT.OBJ=KTO:[132,24]NTI11MOLB.VGN/EX:NTINIT
MDO:NTLROO.OBJ=KTO:[132,24]NTL0LB.VGN/EX:NTLROO
MDO:NTLGET.OBJ=KTO:[132,24]NTL0LB.VGN/EX:NTLGET
.DISABLE DATA #0
.CLOSE #0
.;
.; Extract modules to be patched
.;
LBR @MDO:TTOLBR
.;
.; Create patch utility command file
.;
.OPEN #0 MDO:TTOPAT.CMD
.ENABLE DATA #0
PAT MDO:NTINIT.OBJ;2=MDO:NTINIT.OBJ;1/CS:125037,NTINIT.POB;1/CS:020422
PAT MDO:NTLROO.OBJ;2=MDO:NTLROO.OBJ;1/CS:132607,NTLROO.POB;1/CS:013527
PAT MDO:NTLGET.OBJ;2=MDO:NTLGET.OBJ;1/CS:014063,NTLGET.POB;1/CS:012326
.DISABLE DATA #0
.CLOSE #0
.;
.; Patch object modules for changes
.;
@MDO:TTOPAT
PIP MDO:*.OBJ;*/PU
.;
.; Replace changed modules in libraries
.;
SET /UIC=[132,24]
LBR KTO:NTI11M.OLB/RP/-EP=MDO:'MODUIC'NTINIT.OBJ;2
LBR KTO:NTL.OLB/RP/-EP=MDO:'MODUIC'NTLROO.OBJ;2
LBR KTO:NTL.OLB/RP/-EP=MDO:'MODUIC'NTLGET.OBJ;2
SET /UIC=[130,24]
LBR KTO:NETLIB.OLB/RP/-EP=MDO:'MODUIC'FNDTTO.OBJ
SET /UIC='MODUIC'
.;
.; Create NETCOM.CMD correction source file
.;
.OPEN #0 MDO:NETCOM.COR
.ENABLE DATA #0
MDO:NETCOM.CMD;2=MDO:NETCOM.CMD;1/-AU

```

©

RSX MULTITASKER

```

-/.OPEN...NTINITBLD.ODL/,/,/;TTOMOD/
-/.ROOT/,/,/;TTOMOD/
.ROOT ROT-LIB-RSX-EXS-OVR
-/ROT:/,/,/;TTOMOD/
LIB: .FCTR IN:[130,24]NETLIB/LB:FNDTTO
-/.OPEN...NTLBLD.ODL/,/,/;TTOMOD/
-/LIB:/,/,/;TTOMOD/
LIB: .FCTR IN:[130,24]NETLIB/LB:CBTA2:GCL:TPARS:FNDTTO
/
.DISABLE DATA #0
.CLOSE #0
.;
.; Extract NETCOM.CMD from NETGENCLB.VGN
.;
LBR MDO:NETCOM.CMD;1=KTO:[137,10]NETGENCLB.VGN/EX:NETCOM
.;
.; Apply NETCOM.COR source correction file
.;
SLP @MDO:NETCOM.COR
PIP MDO:NETCOM.CMD/PU
.;
.; Reconstruct NETGEN.CLB
.;
SET /UIC=[137,10]
LBR KTO:NETGEN.CLB/DE:NETCOM
LBR KTO:NETGEN.CLB/CO=KTO:NETGEN.CLB
PIP KTO:NETGEN.CLB/PU
LBR KTO:NETGEN.CLB/IN=MDO:'MODUIC'NETCOM.CMD
SET /UIC='MODUIC'
.;
.; Remove any utilities installed at start
.;
.IFT INSLBR PRV REM ...LBR
.IFT INSMAC PRV REM ...MAC
.IFT INSPAT PRV REM ...PAT
.IFT INSPIP PRV REM ...PIP
.IFT INSSLP PRV REM ...SLP
;
;This completes application of the console data structures changes
;

```

These files are applied after PREGEN has produced a copy of the DECnet distribution kit, and the changes in the DECnet save set contained on RSX11M V4.1 Update E have been applied. These files are applied with the command @DECUSDNET. When the modification procedure finishes, the DECnet generation procedure is invoked by @[137,10]NETGEN.

Note that these changes may not apply to Full-Function DECnet kits, or to DECnet-PSI software, as these distribution kits are not available at the author's site.

Professional 300 Series Communications - Part I

Thomas Turano
 Roman Pinsky
 Laboratory Data Products
 Digital Equipment Corporation
 Marlborough, Massachusetts

Introduction

Recently we considered the various ways a program running on a PRO 350/380 could exchange data with a program running on another computer. In attempting to use the PRO's communications ports to transfer data, we encountered several problems which we think would be beneficial to discuss with the readers of the MULTITASKER.

A PRO 350/380 computer has three hardware ports with which it communicates with other computers. Two of these ports are RS-232/RS-423 serial lines and one is an optional ETHERNET connection. The two RS-232/RS-423 ports are the printer port (TT2:) and communication port (XK0:). Both ports are standard on the PRO. Although they are both serial line ports they have different characteristics. The third port (NX0:) is used for ETHERNET communications when the PRO is configured with an optional DECNA ETHERNET board. The different ways in which these ports operate provide flexibility but with added complexity.

This paper modifies one example to describe the various ways each port can be used to send and receive data. Our particular application did not allow us to use DECNET nor the Pro's communications routines. We do not discuss their use. Instead, we concentrate on the use of P/OS system calls to accomplish our objectives.

It is important to remember that when NOT using established protocols to deal with communications between computers, the applications programs themselves become responsible for maintaining data integrity. Protocols, in general, are complicated schemes for maintaining a link between computers and detecting when that link has been damaged. This means that if messages are lost (for instance when someone unplugs the serial line of one of the computers) it is up to the application programs to discover the fault and take corrective action. Communications protocols are

beyond the scope of this article although most applications can be made fairly secure with a minimum of protocol overhead.

Example Application

The purpose of this article is to discuss the various ways the communications ports can be accessed, regardless of the type of data being transmitted. We assume a simple protocol consisting of a message which is delimited by two characters: a percent (%) to start the message and a carriage return (<CR>) to terminate it. The percent allows the receiving machine to determine that data follows and to distinguish the data from line noise. The carriage return was chosen as a terminator so that the receiving computer has the option of using a FORTRAN READ to receive the data. How the carriage return is placed in and removed from the data is discussed in the example programs. An additional constraint of the example protocol is that the PRO program begins the conversation and receives a reply each time it sends data to the other machine. The examples we use to demonstrate QIO requests also demonstrate the use of a timeout which allows the PRO to detect the loss of a message or a reply. The FORTRAN READ/WRITE example program does not use the timeout feature. The FORTRAN example to demonstrate a timeout is beyond the scope of this article.

Our example program runs on the PRO and initiates data transfer with a remote computer. The remote computer runs a program which is waiting for the data. In our example, the PRO sends a series of commands (ASCII strings) to the remote system and the program on the remote system interprets the commands and takes the appropriate actions. As each action is completed, the remote computer sends a reply to the PRO. These reply strings are also ASCII strings. They vary in length depending upon which command is sent by the PRO.

The PRO can send the following commands:

1. %INIT<CR> - Initialize the other program and begin taking data
2. %INPUT<CR> - Return data to the PRO
3. %CLOSE<CR> - Terminate the experiment

The remote computer replies to the INIT and CLOSE commands by returning the string %000<CR> for success or %500<CR> for failure. The reply to the INPUT command consists of a three character success indicator ("000" if the command is successful) followed by a one character data count, a two character count of the number of characters to follow, followed by the data. So if data were

RSX MULTITASKER

```

100      FORMAT(lx,'Hit return to begin.', $)
      READ(5,200) start      ! OK to begin to transmit
200      FORMAT(lx,lA1)
C
C Output the command to BEGIN
C
      WRITE(1,300)          ! Transmit command to initialize
300      FORMAT(x,'%INIT')      ! remote system
C
C Read the reply from the remote system
C
      READ(1,400) percent,status      ! Read the reply put 1st
      FORMAT(lA1,lI3)                ! character into %
400      IF ( percent .NE. '%' ) stop ' Protocol violation error '
      IF (status .NE. succes) GOTO 999 ! If the remote system reports
      failure report error
C
C Request data to be input
C
      WRITE(1,500)
500      FORMAT(x,'%INPUT')

      READ(1,600) percent,status,valnum,chrnum,(data(i),i=1,valnum)
      IF ( percent .NE. '%' ) stop ' Protocol violation error '
      IF (status .NE. succes) GOTO 999 ! If the remote system reports
600      FORMAT(lA1,lI3,lI1,lI2,<VALNUM>I3) ! failure report error

      WRITE(1,700)          ! Transmit command to stop
700      FORMAT(x,'%CLOSE')      ! remote system
      READ(1,400) percent,status
      IF ( percent .NE. '%' ) stop ' Protocol violation error '
      IF (status .NE. succes) GOTO 999 ! If the remote system reports
      failure report error
C
C Successful completion
C
      WRITE(5,800)          ! Success received
800      FORMAT(x,'The program has terminated normally. ')
C
C Close the logical unit
C
      CLOSE (unit=1)      ! Close the port
      CALL EXIT
C
C Error returned
C
999      WRITE(5,850)          ! Failure received from
      remote system, so stop
850      FORMAT(x,'The remote computer has returned failure. ')
C
C Close the logical unit
C

```

RSX-22

RSX MULTITASKER

```

      CLOSE (unit=1)
1100     WRITE(5,900)
900     FORMAT(x,'The PRO program has abnormally terminated. ')
      END

```

Example Program: Printer Port Using QIO Requests

An alternate method of communicating with another computer through the PRO printer port is by using QIO requests. QIO requests allow greater flexibility than the FORTRAN READ/WRITE statements. For example, you can not read a buffer of 500 characters using a FORTRAN READ statement. This number of characters exceeds the system default input buffer size, therefore, a QIO must be used to read a large number of characters into a single buffer. While the use of a QIO requires more sophistication than a simple FORTRAN READ/WRITE, it is still rather straight forward.

Specifically, the QIO statement is a subroutine with the following form:

```
CALL QIO (function,lun,event_flag,priority,io_status_block,
         device_dependent_parmameters,directive_status)
```

Individually these parameters have the following meanings:

1. Function - describes what the QIO is to do. In the example below the QIOs are used to read and write. The value of the parameter for READ LOGICAL BLOCK is 10400 octal, READ VIRTUAL BLOCK 1200 octal and WRITE is 11000 octal.
2. Lun - is the logical unit to which the data is read or written.
3. event_flag - is a flag which is set when the QIO completes. It is generally used for synchronization and is not explicitly used in the program example.
4. priority - although this variable must be in the string it is ignored.
5. io_status_block - is a 2 word array which returns the final I/O status.
6. device_dependent_parameters - is an array of data which informs the device about the impending QIO. In our example we are only concerned with the first three parameters. The QIO is to the

RSX-23

serial line has a possible parameter list of:

- o `buffer_address` - the address of the buffer to be transmitted or into which the data is to be received
- o `len` - the length of the data to be transmitted or received (in bytes)
- o `timeout` - the length of time that a QIO may remain uncompleted before terminating without completing

The QIO to the printer port terminates when either the number of characters requested is received OR a carriage return is received. Therefore, the QIO is posted for the largest number of characters expected. A carriage return in the reply string will terminate buffer whenever it is received, whether or not the number of characters requested has been received.

7. `directive_status` - is the status returned by the QIO. The program checks this status to determine if the QIO was successful.

In our example the specific form of the QIO used is `WAIT QIO (WTQIO)`. `WTQIO` has the same argument string, but waits for the QIO to complete before returning to the program. The `WAIT` form is used because our protocol is synchronous; that is, for every command there is a reply. There is no reason to go on unless that reply is received. It is here that a timeout is required so that if the reply is lost, the program does not wait forever.

The system call that is required when using QIO's is the `GET ADDRESS` call (`GETADD`). `GETADD` gets the address of the buffer which is to be transmitted, or into which data is to be received. This call is documented in the example program itself. For a more extensive discussion of the QIO mechanism, refer to the *P/OS System Reference Manual*.

The example given below is the same program written with QIO requests rather than FORTRAN `READ/WRITE` statements. Since the QIO to do a read and write requires more coding than does the FORTRAN `READ/WRITE`, the read and write QIOs have been broken out into subroutines which are called each time the transfer of data is to take place. This is done simply to reduce the amount of code.

Unlike the `FORMAT` statement of FORTRAN `READ`, which puts the data into the form requested, the QIO returns into the input buffer the actual characters received. Remember that the data is sent across the serial line in ASCII form and if the program requires numbers instead of characters, the program must convert these characters into the numbers they represent. In the example below, the status returned within the remote system's reply is compared to the value 0 (success). Although the program could have compared the three character string returned in the reply to the character string "000" we preferred to convert the reply string to the value the characters represent, and then to compare this value to 0. Since this is done for every reply string, the conversion is done within a subroutine.

```

PROGRAM demo
IMPLICIT integer (A-Z)

INTEGER      status  ! Status returned by the remote system
INTEGER      succes  ! Constant value for success
INTEGER      unit    ! Logical unit number for TT2:

C
C ASCII values
C
CHARACTER*1  start   ! Command to begin transmitting data
CHARACTER*80 cmd     ! Array to hold the transmitted data
CHARACTER*80 reply   ! Array to hold the received data

C
C Define the constants
C
unit = 2                ! Unit number for TT2:
succes = 0              ! Data collected successfully
lun = 4                 ! Logical unit number
replen = 80            ! Maximum length of the reply string

C
C assign lun line to (TT2:)
C
C          lun = 4
C          device = tt2:
C
CALL ASNLUN(lun,'TT',unit,idsw)
if (idsw .lt. 0) stop 'error assign lun' ! If no lun assigned stop

C
C Read input from the PRO terminal to start the program once the
C experiment is ready.
C
WRITE(5,*) 'Hit return to begin.' ! OK to begin to transmit
READ(5,100) start
100      FORMAT(1x,1A1)
C
C Output the command to INITIALIZE the remote system
C
cmd = '%INIT'
```

RSX MULTITASKER

```

      replen = 7          ! Exact reply length to be read
C
C Assign lun line to (XK0:)
C
C      lun = 4
C
CALL ASNLUN(lun,'XK',unit,idsw)
if (idsw .lt. 0) stop 'error assign lun' ! On error stop
C
C Call subroutine to clear the type-ahead buffer
C
CALL clear(lun)
C
C Read input from the PRO terminal to start the program once the
C experiment is ready.
C
WRITE(5,*) 'Hit return to begin.'
READ(5,100) START
100      FORMAT(1x,1A1)
C
C Output the command to INITIALIZE the remote system
C
      cmd = '%INIT'
      cmdlen = 5
      CALL wrlnk(cmd,cmdlen,lun)
C
C Read the reply from the remote system
C
CALL rdlnk(reply,replen,lun)
status = conv(reply,replen,2,4) ! Convert the characters to
      numbers
IF (status .NE. succes) GOTO 999! If failure report error
C
C Transmit a command to INPUT data
C
      cmd = '%INPUT'
      cmdlen = 9
      CALL wrlnk(cmd,cmdlen,lun)
C
C Read first part of the reply from the remote system
C
CALL rdlnk(reply,replen,lun)
status = conv(reply,replen,2,4)
IF (status .NE. succes) GOTO 999
C
C Determine the rest of the data left in a buffer, if any
C if so read it
C
      chrnum = conv(reply,replen,6,7)
      IF (chrnum .ne 0) CALL rdlnk(datrep,chrnum,lun) ! read the rest
C
C Transmit command to stop the remote system
C

```

RSX-30

RSX MULTITASKER

```

      cmd = '%CLOSE'
      cmdlen = 6
      CALL wrlnk(cmd,cmdlen,lun)
C
C Read the reply from the remote system
C
CALL rdlnk(reply,replen,lun)
status = conv(reply,replen,2,4)
IF (status .NE. succes) GOTO 999
C
C Successful completion
C
WRITE(5,200)
200      FORMAT(x,'The program has terminated normally.')
      CALL EXIT
C
C Error returned
C
999      WRITE(5,300)
300      FORMAT(x,'The remote computer has returned failure.')
      END
C
C *****
C
      SUBROUTINE clear (lun)
      IMPLICIT INTEGER (A-Z)
C
C This routine will clear type-ahead buffer
C
      INTEGER          lun          ! LUN for the device
      CHARACTER*80     clrbuf       ! Temp buffer to read into
      INTEGER          ipari(6)! Parameter buffer
      INTEGER          iosbi(2)! IO status block
      INTEGER          idsw         ! Device status word
      DATA            iorlb /"1200"! Read logical block with purge
C
C Parameter block values
C
      ipari(2) = 80          ! read and purge 80 bytes of data
      ipari(3) = 0          ! time out 0
C
C Do the read to clear the buffer
C
      CALL GETADR(ipari(1),clrbuf) ! Get address of buffer
      CALL QIO(iorlb,lun,,iosbi,ipari,idsw) !
      IF (idsw .LT. 0 ) STOP ' device error '
      IF (iosbi(1) .LT. 0) STOP ' error in read and purge '
      RETURN
      END
C
C alternative (second) way to clear type-ahead buffer

```

RSX-31

RSX MULTITASKER

```

C
C      SUBROUTINE clear (lun)
C      IMPLICIT INTEGER (A-Z)
C
C      INTEGER          lun
C      CHARACTER*80     clrbuf
C      INTEGER          ipari(6)
C      INTEGER          iosbi(2)
C      INTEGER          idsw
C
C      DATA            iovlb /"10400/
C      DATA            iokill /"12/
C
C      Parameter block values
C
C      ipari(2) = 80      ! Number of characters to read
C
C      CALL GETADR(ipari(1),clrbuf) ! Get the buffer address
C      CALL QIO(iorvb,lun,,iosbi,ipari,idsw) ! Read buffer
C      IF (idsw .LT. 0 ) STOP ' device error '
C      CALL QIO(iokill,lun) ! Kill the READ
C
C      RETURN
C      END
C
C *****
C      SUBROUTINE wrlnk (CMD, MAXLEN, LUN)
C      IMPLICIT INTEGER (A-Z)
C
C      This routine will write to device using QIO
C
C      INTEGER          maxlen      ! Length of string passed
C      INTEGER          lun          ! LUN for transfer
C
C      QIO parameters
C
C      INTEGER          iparo(6)     ! I/O parameter block
C      INTEGER          iosbo(2)     ! I/O status block
C      INTEGER          idsw         ! Device status word
C      INTEGER          efn          ! Event flag
C      CHARACTER*80     cmd          ! Buffer of data to transmit
C      DATA            iowvb /"11000/ ! Write Virtual Block
C
C      QIO parameter values
C
C      efn = 1              ! Event flag
C      iparo(2) = maxlen   ! Number of char requested
C
C      Get the address of the I/O parameter block and then
C      transmit the data (wait for the completion)
C

```

RSX MULTITASKER

```

      CALL GETADR(iparo(1),cmd)      ! get buffer address
      CALL WTQIO(iowvb,lun,efn,,iosbo,iparo,idsw)
      IF (idsw .LT. 0 ) STOP ' device error '
      IF (iosbo(1) .LT. 0) STOP ' error write to device '
      RETURN
      END
C *****
C
C      SUBROUTINE rdlnk (REPLY, MAXLEN, LUN)
C      IMPLICIT INTEGER (A-Z)
C
C      This routine will READ from device using QIO
C
C      INTEGER          maxlen      ! Length of string passed
C      INTEGER          lun          ! Logical unit number for transfer
C
C      QIO parameters
C
C      INTEGER          ipari(6)     ! I/O parameter block
C      INTEGER          iosbi(2)     ! I/O status block
C      INTEGER          idsw         ! Device status word
C      INTEGER          efn          ! Event flag
C      CHARACTER*80     reply       ! Buffer of data to receive
C      DATA            iorlb /"1200/ ! Read logical Block
C                                     with timeout
C
C      QIO argument values
C
C      efn = 1
C      ipari(2) = maxlen
C      ipari(3) = 2              ! 2 x 10 sec timeout period
C
C      Get the address of the I/O parameter block and then
C      receive the data (wait for the completion)
C
C      CALL GETADR(ipari(1),reply)
C      CALL WTQIO(iorlb,lun,efn,,iosbi,ipari,idsw)
C
C      Check to determine if the QIO request failed or timedout
C
C      If ((idsw .EQ. 2) .OR. (idsw .LT. 0)) stop ' failure or TMO'
C
C      If QIO OK then check for run time failure
C
C      IF (iosbi(1) .LT. 0) STOP ' error read from device '
C      IF (reply(1:1) .NE. '%') STOP ' protocol error '
C
C      RETURN
C      END
C *****
C

```

```

INTEGER FUNCTION conv (srting, maxlen, stpos, endpos)
C
C This routine will translate bytes into integer value
C
C INTEGER maxlen,stpos,endpos,result,base
C CHARACTER*80 string
C Initialize the result
C
C result = 0
C base = 1
C
C Do conversion from ASC decimal to integer decimal
C
C Do 10 I = endpos,stpos,-1
C   result = result + base * (ICHR((string(i:i))) - 48)
C   base = base * 10
10 CONTINUE
conv = result

RETURN
END

```

Editor's Note

The is the first of a two part article by Thomas Turano and Roman Pinsky. Next month ETHERNET running on the PRO will be investigated.

The QIO

Ralph Stamerjohn
Meridian Technology Corp.
St. Louis, Missouri

A fully configured RSX system has 80 system directives, but one stands alone as the most important. The QIO directive and its synchronous form, QIOW, is the only means of doing I/O on a RSX system. Unlike operating systems such as RSTS/E, RT-11, and UNIX which use different system calls for I/O, RSX has only the QIO.

The QIO was a key component in the original RSX design. It has lasted, without any major changes from RSX-11D V1 to the latest release of VMS. You still use the same QIO that was used back in 1971 to open a file, read data, rewind a magtape, or write a message to the terminal.

Most programmers equate QIO's with real-time devices such as A/D converters or LPAll's. Many RSX programmers have never needed to program a QIO directly. Instead the two I/O run-time system supplied with RSX, FCS and RMS, hide the actual QIO's. It is simple to issue a QIO from even high-level languages and thereby achieve greater throughput and device control than normally available.

The simple Fortran-77 program CKSUM1 in figure 1 reads a fixed-length (512 byte), unformatted file and caculates the summation and exclusive-or checksum. On the 2048 block TEST.DAT file, the program takes 104 seconds on a PDP-11/24 with RL02's.

Figure 2 shows how the program was converted to use double-buffered logic to read 16 disk blocks at a time. The new program, CKSUM2, now takes only 21 seconds, a 5-fold increase in performance. Note that performing just the caculation loop 2048*512 times takes 15.5 seconds. In effect, an I/O bound program has been made almost compute bound.

To use QIO's directly, one must start by mastering its basic components and learning how to properly issue a QIO and test for success and errors.

A QIO or QIOW consists of 6 standard parameters and 0-6 I/O specific optional parameters. The standard parameters are function code, logical unit, event flag, priority, I/O status address, and AST address. All but the first two are optional. An easy way to keep the parameters and their order straight is the memory aid 'FLEPIA'.

The function code identifies the type of I/O operation. While supplied as a word, there are not 65535 separate RSX I/O functions. Instead the function code is really treated as a two byte field. The high byte is the actual I/O function code and can range from 0-31. The low byte is the I/O function subcode and contains values specific to the actual device.

Most of the 32 I/O function codes have standard assignments. For instance, the value 21 is a read virtual and a 22 is a write virtual. Note that the actual word values in octal are 10400 and 11000 respectively. In the sample program, notice how a PARAMETER statement is used to define the read virtual function.

The low byte is used by specific devices to differentiate a particular function. For instance, a read logical request (code 2) to a terminal can be set not to echo input by setting the subcode to 20 octal. This makes 1020 the octal word equivalent of the I/O function.

Macro-11 programmers should use the symbol names for the I/O function codes, i.e. IO.RVB for read virtual block. A complete list of I/O function code symbols and values is found in Appendix B of the RSX-11M/M-PLUS I/O Drivers Reference Manual. The manual also discusses what functions each device driver uses.

One caution about I/O functions. Whenever possible the read/write virtual functions (IO.RVB/IO.WVB) should be used instead of the read/write logical (IO.RLB/IO.WLB). A common error made by a privilege program is to write some message intended for a terminal over the boot block on a disk. This is because RSX allows a privilege program to use logical I/O function to address absolute block numbers on the disk. If the logical unit was not properly assigned to a terminal, disaster can happen.

Using the virtual form of the read/write I/O codes keeps all protection mechanisms in force at all times. However, if the function uses any subcode fields, the read/write logical must be used. The RSX executive zeros this byte when processing virtual requests.

The next parameter in a QIO is the logical unit number. The logical unit is assigned to the hardware device. Logical units range from 1 to the limit determined when the task was linked (TKB UNITS option).

It is important to remember that RSX has four types of device names: real, psuedo, logical, and TI:. A real device is exactly that, a hardware device on the system. For instance the name DL3: refers to RL01/02 drive 3. A psuedo device is a device that always points to a real device. Common psuedo devices include LB0: and SY0: for the system disk, SP0: for the spooling disk, and CL0: for the console listing.

A logical device is a device name that maps to a real or psuedo device. What sometimes confuses people is that a logical name can be the same as a real device, but point to some completely different device. If I issue the command 'ASN DB1:=DL3:' and then assign DL3:, I am really doing I/O to DB1:. Finally, the name 'TI0:' always refers to the terminal at which the program is running.

Logical unit assignments take place in a variety of ways. If you do nothing when building a program, the task is given 6 logical units. Numbers 1-4 default to device 'SY0:', logical unit 5 to 'TI0:', and logical unit 6 to 'CL0:'. Any higher logical units are left unassigned. TKB can make specific device assignments if you use the "ASG=ddn:unit" option.

Note that all the devices named above are not real, instead they are psuedo devices and potential logical names. When you install a task, the INS program gets the device name set at task build time and first searches for a matching logical name. If match is found, the corresponding real device is assigned to the logical unit. Next the real and psuedo devices are searched. If no matching name is found, a warning message is output and the logical unit left unassigned.

Once a task is installed, the logical unit assignments can be changed externally using the REAssign command. A running program can change its assignment by using the assign logical unit (ALUN\$) directive. Note that RSX always assigns logical units to real devices, not logical device names. Once RSX makes an assignment, changing or deleting the logical device name has no effect.

When the LUN command or RMDemo task page is used to view a task's logical units, you sometimes see more units than expected. This is because features such as overlays and ODT add logical units beyond what would normally be supplied. One debugging technique is to use the LUN command to examine a task linked with ODT and then the REAssign command to assign the last two logical units to a different terminal. Now you keep the program's and ODT's I/O on separate terminals.

The event flag is the first optional QIO parameter. I/O takes time and the event flag is one means of determining when it is finished. If you specify an event flag, RSX clears it immediately and then sets the flag when the I/O completes. As shown in figure 2, the normal method of testing for I/O completion is to wait on the event flag (CALL WAITFR).

When doing synchronous I/O, the QIOW directive is used to issue the I/O request and wait on completion. RSX does not require the event flag for the QIOW form, however if one is not named, you have simply issued a QIO and control is returned immediately to the program. A easy means of exhausting system pool is a loop which includes a QIOW with no event flag.

The fourth parameter, priority, is left over from RSX-11D and remains valid for only IAS. These systems allowed a program to specify a priority of the QIO different from the issuing task priority. In RSX-11M, RSX-11M-Plus, and Micro/RSX this parameter is always ignored and is almost always left blank.

RSX devices always return I/O status to a double-word block. The address of this block, if any, is named in the fifth parameter. Just like an event flag, the I/O status block is zeroed when the QIO is issued and set to the final status when the I/O completes. One simple way to check for I/O completion is to test the I/O status for a non-zero value.

It is critical that the I/O status information be processed in pieces. You first test the low byte of the first word for success or error. Success is always greater than zero, an error is negative. Device specific status is sometimes returned in the high byte of the first word.

The second word is usually a transfer count. For example, the number of bytes successfully read in a read logical/virtual request is stored in the second word. Note that an I/O can transfer data and still get an error return in the first byte. A IO.RVB for five disk blocks to a four block file will return an end-of-file error (IE.EOF) and a transfer count of 4000 octal.

In the example, the transfer count is used to test for I/O success or failure. If no data is transferred (IOST(2) = 0), the program assumes it has reached the end of the file.

The last standard QIO parameter is the AST address. AST stands for asynchronous system trap. It is a powerful mechanism in RSX that allows a task execution to be interrupted and control passed to some routine to service an event. The QIO AST is triggered when the I/O completes so your service code processes the results of the I/O. You cannot set the AST address from a high-level language, so the parameter is not even present. Further discussion of AST's and I/O programming will come in a future issue.

After the six standard parameters, come up to six I/O function specific parameters. There is an almost infinite variety to format and uses of these parameters.

One common set of parameters are those used for output to terminals, printers, and such unit-record equipment. Only three parameters are used, a buffer address, buffer length, and carriage control. The latter is a character that specifies how many blank lines follow the text. The standard Fortran formatted control characters are used, space for one line, l for page eject, and so forth. A zero value means no carriage control should be supplied.

Disk reads and writes take a slightly different format. The first two parameters are still a buffer address and buffer length. But the third parameter is ignored and the next two specify the starting disk block as a 24-bit number. The low part is in parameter 5 and the high byte in parameter 4. The final I/O parameter is not used.

If you use the read/write virtual functions (IO.RVB/IO.WVB), the disk block number is within the file which you previously have opened. Disk files start with block 1.

But if you are using the read/write logical functions (IO.RLB/IO.RLB), you are addressing absolute disk blocks, starting with block 0. Your program must be privileged or the disk mounted foreign to do such I/O.

Looking at each part of CKSUM2, you can see how simple it is to use QIO's for disk I/O. The two parameter statements are used simply to define the number of disk blocks per I/O and the read virtual block function.

```
PARAMETER      IOBLK=16
PARAMETER      IORVB="21*"400
```

The next section declares the QIO I/O status blocks, I/O parameters, and input buffers for both QIO's. A good practice is to always initialize the parameter blocks to zero.

```
INTEGER*2      IOST1(2),IOPRM1(6),BUF1(256*IOBLK)
INTEGER*2      IOST2(2),IOPRM2(6),BUF2(256*IOBLK)
DATA           IOPRM1/6*0/,IOPRM2/6*0/
```

Fortran has a special feature that allows a program to declare it will be performing its own I/O to a file. A minus one value to the BUFFERCOUNT keyword disables normal Fortran I/O.

```
OPEN           (UNIT=1,...,BUFFERCOUNT=-1)
```

Before a QIO is issued, the I/O parameter array must be initialized. As mentioned previously, the first parameter in a disk I/O is the address of the buffer. Fortran has no syntax for setting a variable to the address of another, so the system library routine GETADR is used. It simply stores the address of the second argument into the first. The second array element is set to the size of the buffer in bytes.

```
CALL GETADR(IOPRM1(1),BUF1(1))
IOPRM1(2) = 512*IOBLK
```

```
CALL GETADR(IOPRM2(1),BUF2(1))
IOPRM2(2) = 512*IOBLK
```

The two initial QIO's are now issued, using the system library routine QIO. The first reads from the beginning of the file (block 1) and the second is offset by the blocking factor. Note the QIO's use the same logical unit as the OPEN statement and separate event flags so the program can tell when each QIO finishes.

```
IOPRM1(5) = 1
CALL QIO(IORVB,1,1,,IOST1,IOPRM1)
IOPRM2(5) = IOPRM1(5) + IOBLK
CALL QIO(IORVB,1,2,,IOST2,IOPRM2)
```

The program now enters its main loop. It waits for the first I/O to finish. Success or error is tested by checking the transfer count. In this simple example, it is assumed that a zero transfer means the end-of-file has been reached. BUF1 is checksummed and the QIO reissued. We know the starting block number will be whatever the last block used in the second QIO plus the blocking factor constant IOBLK.

```
100 CALL WAITFR(1)
   IF (IOST1(2) .EQ. 0) GOTO 999
   DO 110 I = 1,IOST1(2)/2
       SUMADD = SUMADD + BUF1(I)
       SUMEOR = IEOR(SUMEOR,BUF1(I))
110   CONTINUE
   IOPRM1(5) = IOPRM2(5) + IOBLK
   CALL QIO(IORVB,1,1,,IOST1,IOPRM1)
```

When finished with BUF1, the program waits for the second QIO to finish and repeats the steps above.

There are a few caveats to QIO's and disk I/O. A QIO is a directive and returns directive status. If the QIO fails as a directive, waiting on an event flag is meaningless because the I/O has never been issued. In a more rigorous program example, the CALL QIO statements would add the directive return and test for error.

```
CALL QIO(IORVB,1,1,,IOST1,IOPRM1,IDS)
IF (IDS .NE. 1) THEN
    ...error processing code...
```

RSX also uses two end-of-file definitions. When using FCS or RMS, they keep an EOF position to a byte within a block. The file may actually have blocks allocated beyond this position. The file

system considers its end-of-file to be the last block actually allocated in the file. When using QIO's to access disk blocks, you are using the file system's EOF.

I have only touched the surface on QIO's. In future articles I will examine AST's, high-performance terminal I/O, and more details on the file system. Read Chapter 1 of the RSX-11M/RSX-11M-Plus I/O Driver Reference Manual for an introduction to RSX input/output and Chapter 5 for details on disk QIO's.

```
PROGRAM          CKSUM1

REAL*4           TIME
INTEGER*2        SUMADD,SUMEOR
INTEGER*2        BUF1(256)

OPEN             (UNIT=1,FILE='TEST.DAT',STATUS='OLD',RECORDTYPE='FIXED',
1               ACCESS='SEQUENTIAL',FORM='UNFORMATTED')

SUMADD = 0
SUMEOR = 0
TIME = SECNDS(0.0)

100  READ (1,ERR=999) BUF1
DO 110 I = 1,256
    SUMADD = SUMADD + BUF1(I)
    SUMEOR = IEOR(SUMEOR,BUF1(I))
110  GOTO 100

999  TIME = SECNDS(TIME)
WRITE (5,1000) TIME,SUMADD,SUMEOR
1000 FORMAT (' Elapse seconds = ',F6.2,' Checksums = ',06,1X,06)
STOP
END
```

Figure 1

```
PROGRAM          CKSUM2

PARAMETER        IOBLK=16
PARAMETER        IORVB="21*"400

REAL*4           TIME
INTEGER*2        SUMADD,SUMEOR
INTEGER*2        IOST1(2),IOPRM1(6),BUF1(256*IOBLK)
INTEGER*2        IOST2(2),IOPRM2(6),BUF2(256*IOBLK)

DATA IOPRM1/6*0/,IOPRM2/6*0/
```

RSX MULTITASKER

```

OPEN      (UNIT=1,FILE='TEST.DAT',STATUS='OLD',BUFFERCOUNT--1)

SUMADD = 0
SUMEOR = 0
CALL GETADR(IOPRM1(1),BUF1(1))
IOPRM1(2) = 512*IOBLK
CALL GETADR(IOPRM2(1),BUF2(1))
IOPRM2(2) = 512*IOBLK
TIME     = SECNDS(0.0)

IOPRM1(5) = 1
CALL QIO(IORVB,1,1,,IOST1,IOPRM1)
IOPRM2(5) = IOPRM1(5) + IOBLK
CALL QIO(IORVB,1,2,,IOST2,IOPRM2)

100      CALL WAITFR(1)
IF (IOST1(2) .EQ. 0) GOTO 999
DO 110 I = 1,IOST1(2)/2
        SUMADD = SUMADD + BUF1(I)
        SUMEOR = IEOR(SUMEOR,BUF1(I))
110      CONTINUE
IOPRM1(5) = IOPRM2(5) + IOBLK
CALL QIO(IORVB,1,1,,IOST1,IOPRM1)

CALL WAITFR(2)
IF (IOST2(2) .EQ. 0) GOTO 999
DO 120 I = 1,IOST2(2)/2
        SUMADD = SUMADD + BUF2(I)
        SUMEOR = IEOR(SUMEOR,BUF2(I))
120      CONTINUE
IOPRM2(5) = IOPRM1(5) + IOBLK
CALL QIO(IORVB,1,2,,IOST2,IOPRM2)
GOTO 100

999      TIME = SECNDS(TIME)
WRITE   (5,1000) TIME,SUMADD,SUMEOR
1000    FORMAT (' Elapse seconds = ',F6.2,' Checksums = ',O6,1X,O6)
STOP
END

```

Figure 2

THE mini. tasker

DECUS

RT-11 SIG NEWSLETTER

SJ

RTMON

RMON

FILEX

CSI

ODT

PIP

LD

SYSMAC

FB

DIR

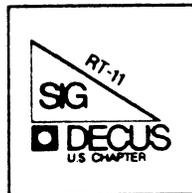
QUEUE

QUEMAN

KMON

TECO

PAT



DUP

VM

K52

LIBR

BINCOM

KED

DUMP

BUP

SRCCOM

LINK

SIPP

FORMAT

SLP

RESORC

TTYSET

IND

MACRO

XM

JSW

HELP

RT SIG Steering Committee

Chairman

John T. Rasted
JTR Associates
Meriden, CT

MACRO Contact

Nick Bourgeois
NAB Software Services, Inc.
Albuquerque, NM

Personal Computer Contact

LUG Contact

William K. Walker
Monsanto Research Corp.
Miamisburg, OH

FORTRAN Contact

RUNOFF Contact
Robert Walraven
Multiware, Inc.
Davis, CA

Symposium Coordinator

Ned Rhodes
Software Systems Group
Rockville, MD

TECO Contact

Product Planning Contact

John Crowell
Crowell Ltd.
Los Alamos, NM

Communications Committee Representative

FMS Contact

Susan Rasted
Software Dynamics, INC.
Wallingford, CT

DECNET Contact

Ken Demers
Adaptive Automation
New Haven, CT

Newsletter Editor

COBOL Contact

Bill Leroy
The Software House, Inc.
Atlanta, GA

UNIX Contact

Wish List Contact

Bradford Lubell
L.A. Heart Lab
Los Angeles, CA

Pre-Symposium Seminar Coordinator

Suite Manager

Bruce D. Sidlinger
Sidlinger Computer Corp.
San Antonio, TX

RT-11 Hardware Contact

Carl Lowenstein
University of Calif., San Diego
San Diego, CA

Standards Coordinator

Kenneth L. Aydlott
Teledyne Hastings-Raydist
Hampton, VA

TSX Contact

C Contact

Jack Peterson
Horizon Data Systems
Richmond, VA

Digital Counterpart

Diana L. Miller
Digital Equipment Corporation
Maynard, MA

Library Contact

Tape Copy Distribution

Tom Shinal
General Scientific Corp.
Rockville, MD

Tape Copy Generation Contact

Ralston Barnard
Sandia Laboratories
Albuquerque, NM

BASIC Contact

Ed Stevens
EMDA Inc.
Pasadena, CA

APL Contact

Doug Bohrer
Bohrer & Company
Wilmette, IL

CAMAC Contact

J.W. Tippee
Kinetic Systems Inc.
Lockport, IL

RT-11 SIG Announcement

After many years of service to the RT-11 SIG as the RT-11 newsletter editor, Ken Demers has had to give up his position due to business pressures. We would like to thank Ken for the time he has devoted to the SIG and wish him well in his new business venture.

I am happy to announce that Bill Leroy will be our new newsletter editor beginning August 1, 1985. All RT-11 articles, bug fixes, hints, kinks etc. should be sent to him at the following address for inclusion in our section of the new DECUS newsletter.

Contributions to the RT-11 newsletter should be sent to:

Bill Leroy
The Software House
P.O. Box 52661
Atlanta, GA 30355
(404) 231-1484

Bill Leroy
The Software House, Inc.
P.O. Box 52661
Atlanta, GA 30355

Other communications should be sent to either:

John T. Rasted	or	RT-11 SIG
JTR Associates		C/O DECUS
58 Rasted Lane		249 Northboro Road, BP02
Meriden, CT 06450		Marlboro, MA 01752
(203) 634-1632		

His phone number is (404) 231-1484 if you should need to call him regarding an article which you have submitted and then discovered a typo. (One and all, that's an invite to publish!)

John and I are putting together this copy since the change-over from Ken to Bill would have left Bill short on time to get it put together. Please don't blame him for any boobos that we pull!

Since DECUS has now eliminated the individual SIG newsletters in favor of a single combined monthly publication, we will have to conform to the concrete deadlines which are now in force. All SIG contributions are to be in the DECUS office ready for publication by the last day of each month. Please give Bill ample lead time when you submit. He has to format our section and mail it to the office in plenty of time to beat the deadline. If we miss the deadline we miss the next month's issue.

We have in the past published quarterly and it will be up to Bill to decide, based on material available, how often we will submit to the new newsletter. If you don't see us as often as you would like, the reason may be that you haven't taken the time to submit your favorite hint or way of solving a problem or maybe your request for help with a problem you haven't solved. The solution to a problem at your site might just solve the problem for someone else. It might even help them from getting into trouble in the first place.

Sue Rasted
DECUS Communications Committee Representative

Table of Contents

RT-11 SIG Announcement	2
RT-11 VAX/RT Position Paper	3
RUNOFF Fix	10
New Use for Virtual Memory (VM) Handler	11
Annotated Directory RT-11 SIG Tape Spring 1985	25

RT-11 SIG
VAX/RT POSITION PAPER

FORWARD

This is a position paper intended to represent the RT-11 SIG's views on a proposed real-time operating system for the VAX family of computers, most particularly the MicroVAX processors. It is a distillation of requests, opinions, and fights from RT-11 and VAX users, with further input from DEC. This draft is the outcome of the so called "Murphy Meeting" of the RT-32 Working Group held in Albuquerque on March 22, 1985. (Unfortunately most of the Working Group finally couldn't attend, so that there were more DEC people than customers in attendance. We, nevertheless, held our own.)

Comments or questions about this paper may be addressed to the RT-11 SIG Product Planning Contact, John Crowell, Crowell, Ltd., 145 Andanada, Los Alamos, NM 87544.

PREFACE

0.0 There has for the last several years been a growing interest in a Real-Time operating system for the VAX family of computers. Informally dubbed RT-32, such an operating system would provide capabilities not found in present offerings from DEC or third parties. This document represents a condensation of requests and opinions from the RT-11 user community. It describes those features considered to be essential to such an operating system, and may be thought of as representing the position of the RT-11 Special Interest Group (SIG) of the Digital Equipment Computer Users Society (DECUS).

0.1 Inasmuch as the term RT-32 is already in use by another manufacturer, the name VAX/RT will be used hereafter in this document. The features described herein are intended to be functional descriptions rather than requests for specific details. (e.g. A request for search, comparison, and patching utilities need not involve separate utility programs, but might all be incorporated into the requested symbolic debugger.)

1.0 General

The single-user/multi-job environment of RT-11 has served the user well for over ten years, and has shown a remarkable ability to expand its functionality to take advantage of hardware developments within the PDP-11 architecture. With the inevitable dominance of 32-bit hardware in the near future, it is clear that

RT-3

the RT environment should be migrated into that space. In particular with the introduction of the MicroVAX, the need for a responsive single-user/multi-job real-time operating environment has become apparent.

1.1 In the typical RT-11 situation (if anything about RT-11 can be called typical), the software user and the software developer are the same person. In whatever VAX/RT situation is forthcoming, one may well assume that the user will want to do software development, data acquisition and control, and data analysis on the same computer under the same operating environment. One might expect that the user will expect to be able to do some word processing, data base management, and "busy work" as well.

1.2 The Real-Time aspects of such a VAX/RT apply principally to the scientific and/or industrial user who will be interfacing his computer to devices requiring rapid response. In such situations, the allowable frequency of events (interrupts) should be determined primarily by the interrupt latency of the hardware, and not by overhead in the operating system. With existing MicroVAX hardware, an interrupt rate of 10^4 events/sec should be accommodated. For near-future products, a rate of 10^5 events/sec is not an unreasonable request.

1.3 The multi-job capability of VAX/RT should be based upon a priority-only structure. No system overhead should be wasted on job shuffling or round-robin scheduling. There should be minimal overhead devoted to job protection. Real-time jobs have, of course, absolutely highest priority.

1.4 The executive/kernel of VAX/RT should include the "basic" file server. (See Section 2.1.) Directory services for the basic file system should also be in the executive (e.g. wild-card lookups). Thus, for instance, the redundant code now found in RT-11's PIF, DUF, and DIR could be avoided, and user-written programs would have easy access to directory services. Indirect command files (i.e. files consisting of basic monitor command strings), referred to in RT-11 as @FILE processing, should be integral to the executive. (See Section 3.2 for mention of indirect control files.)

1.5 Although the virtual memory aspects of the VAX architecture are popular and valuable, at first-release only non-paging jobs need be supported. At some future date, an optional RUN/VIRTUAL feature will, no doubt, be requested, but is not thought to be a necessity.

1.6 The operating system is to be "open" in the sense that the user/programmer has direct access to all hardware and memory on his system without having to go through the executive. (i.e., The operating system will "get out of the way" when asked.)

RT-4

2.0 File Structure

Not surprisingly, the point of greatest disagreement centers around the file structure of VAX/RT. The file system described here is a proposal based upon compromises between desired functionality and realistic possibility.

2.1 Basic File Structure

The "basic" file structure is supported in the executive. It has the following features:

2.1.1 All files are contiguous on the volume. If an attempt is made to write past the end-of-file, a contiguous extension file is opened (blind to the user). Fragment linkage information is maintained in the directory. Fragments are concatenated when the volume is next "squeezed."

2.1.2 File names are stored in the directory in ASCII (RAD50 may be forever discarded). Both upper and lower case characters are allowed, but without case sensitivity. Dollar sign and underscore characters are legal characters in file names so long as they are not the first character. File names may be one to nine characters. File types may be zero to three characters.

2.1.3 There are no User Identification Codes (UIC). In short, there is no multi-user file protection.

2.1.4 There are no version numbers. As usual, editors and patching utilities are expected to rename source files to type .BAK, but no further maintenance of previous versions is required.

2.2 Subdirectories

Nested directories will be handled as they now are in RT-11, by logical devices LD00: - LDFF: which are bootable. Such subdevices may be assigned logical names which are not restricted to four character device names. (e.g. ASSIGN LD22 MYSTUFF)

2.3 Other File Structures

Other file structures will be supported by means of ancillary control processors (ACP) which run as independent jobs. Programmed requests to be directed to an ACP differ from those to the basic file server in the executive only by flag bit(s) in the parameter block. File structures thus supported might include RT-11, ANSI magtape, RMS, UNIX, MSDOS, and CP/M.

VAX/RT documentation should include instructions and examples to facilitate user-written ACP's.

2.4 Non-file structured I/O

Non-file structured I/O (e.g. reading and writing individual disk sectors) is handled by the executive.

3.0 Accessories

The following is a list of "accessories" deemed necessary for VAX/RT. The left-hand column in each category lists those that should be available concurrent with first release of the operating system. The right-hand column lists those that should be made available as soon as possible within two years of first release. Needless to say, there are many wish list items, not included here, that have been requested. This list is a subset of those features thought to be both necessary and realistic. Names of specific products used here need not be taken literally. In most cases the name of the closest corresponding RT-11 feature is used in the interest of brevity. It is the functionality that is really being requested.

At First Release

Soon Thereafter

3.1 Languages

MACRO (absolute must)
FORTRAN-77
C

BASIC
PASCAL

3.2 User Interface

DCL (e.g. RT-11 with factoring)
UCL/UCI/UCF (supplied and support
for user-written)
Single-line Editor (SL)
@FILE processing
IND

3.2 Utilities

Language Processors
Linker & Librarian
Symbolic Debugger (MACRO)
Editor (screen & line)
(with Macro & indirect file)
Search/Patch (binary)
Compare (source & binary)
DUMP
HELP
SPOOL

TECO
Backup/Restore
RUNOFF
SPLIT

At First Release	Soon Thereafter
3.3 Communications	
Virtual Terminal & file transfer (e.g. VTCOM/TRANSF/KERMIT)	
Ethernet (basic support)	Ethernet/DECNET (end-node only OK)
	Arpanet
3.4 ACP's	
RT-11	RMS
ANSI magtape	UNIX/ULTRIX
3.5 Other (not to be confused with incidental)	
Global Regions	
Shared Libraries	
STARTUP command file	
Virtual Console hooks (e.g. TSX-like virtual lines)	
VM handler (bootable)	
NL handler	
4.0 Unwish List	
There are several features present in RT-11 which should be abandoned. They are, for the most part, obvious, but are included here in the interest of completeness and emphasis. Features to be avoided include:	
64kB job size limitation	
32k block limit on devices	
8 unit limitation on devices	
Multiple monitors	
4.1 In addition, the following exclusions should be in effect:	
All device dependent code should be in handlers. This includes console I/O and formatting of those devices which support formatting in their hardware.	
The executive should be fixed. e.g. There should be no self-relocating KMON or swapping USR.	
Although the .FORK process is good for low-priority cleanup in device handlers, it should not be mandatory - even for timeout support.	

5.0 Miscellaneous Comments

In this section are included general subjects which have been addressed or need to be addressed. Their order is in no particular priority.

5.1 User-Written Device Handlers

An experienced System Programmer should be able to write a simple handler (e.g. CR or PC) in a day or less. An experienced programmer (novice System Programmer) should be able to do so in 3 days. One should be able to write a device handler in a high/middle-level language such as C.

5.2 Bit-mapped Devices

Bit-mapped I/O devices are becoming common (e.g. graphics displays and high-speed data cachers). Consideration should be given to making sure that the executive can be made aware of such "forbiddin'" memory. The availability of Global Regions may be sufficient here.

5.3 Multiprocessing

Although not an initial design feature, the capability of operating more than one processor on the same bus is a trend not to be neglected. Consideration should be given to providing the "hooks" in the executive for future multiprocessing applications. These considerations should include the possibility of the following:

- 5.3.1 Multiple copies of operating system in multiple processors, each with local I/O capabilities, but sharing resources on the bus.
- 5.3.2 Master/Slave Multiprocessing: One copy of the operating system is run by a Master processor which may download jobs to Slave processors. (e.g. RUN/CPU:n job_name)
- 5.3.3 Symmetric Multiprocessing: One copy of operating system in globally shared memory. Any and all processors run executive software and/or jobs. Executive/Kernel assigns jobs to processors in real-time.

5.4 Character-oriented I/O

It is likely that character-oriented device I/O will continue to be a special case requiring optimization and/or specialized coding for some devices. VAX/RT should provide some standard, explicitly-supported, well-documented way to accomplish this.

RUNOFF Fix

5.5 "Console" I/O

Console I/O should be device-independent. It should be possible, for example, to assign a job's console input to a file, and it's output to some other device.

If two or more jobs are sharing a console, both "polite" and "impolite" modes of I/O should be available. Impolite mode would be similar to the method currently used under RT-11, where a high-priority job can force console output. Polite mode should be a "virtual line" approach in which console output would be queued until the operator explicitly requests it. (This latter approach is particularly useful in a program development environment.)

5.6 Supported Devices

It is assumed that at first release all devices supported on MicroVAX systems will be supported by VAX/RT. (This includes MicroVAX-Station graphics processors.) Like RT-11, however, VAX/RT should be thought of as a component-level product. Many Q-bus users will be upgrading LSI-11 systems to MicroVAX and expect to use existing peripherals. While one does not expect every Q-bus device ever sold to be supported, there should at least be handlers provided for RX02 (with buffer forwarding), and RL02.

When the Q-bus was first introduced, its specifications were made public, and third-party vendors were encouraged to manufacture compatible hardware. In an effort to provide support for large disks, filling a then existing gap in DEC's offering, several manufacturers elected to emulate the RK06/07 disks. That VAX/RT initially support foreign devices is, perhaps, an unreasonable request. Consideration should be made, however, for the migration of existing Q-bus systems with RK06/07 emulation. Those same third-parties should have access to sufficient advance information so that they could, if they chose, supply the necessary device drivers. Alternatively, if the demand for such drivers is sufficient, they should be made available through the Digital Unified Software (DCS) program.

Date: Sun 14-JUL-1985 12:19
From: Jack J. Peterson

TO: Robert Walraven (WALRAVEN)
TO: Ken Demers (DEMERS)
CC: RTSIG DISTRIBUTION LIST (@RTSIG)

Subject: Bonner Labs Runoff Bug Fix

Bob:

I've found and corrected another bug in Bonner Labs Runoff Version 6.2L. A summary of the bug and fix follows: [Ken: Please publish this in the next newsletter; Bob: Please forward this to Bonner Labs].

BUG FIX FOR BONNER LABS RUNOFF BL6.2 - #2

Jack J. Peterson
Horizon Data Systems
1901 Wildflower Terrace
Richmond VA 23233

If you use the .AUTOBREAK, .NOAUTOBREAK, .UNDERLINE, or .NOUNDERLINE commands, you will no longer get correct justification of a line that ends with a .PERIOD character (normally .?!.;) when some blank filling is required. The reason is that the routine that processes these commands (in STYLE.MAC) always copies the flags associated with a space to those for the non-expanded space. Unfortunately, this also makes the non-expandable space a break character if space is a break character (the normal case).

To correct the bug, edit STYLE.MAC and find label UNLEND:. Immediately before the RETURN that follows, insert:

```
BICB #CH.BRK,CHTABL+NXS ;Non-expandable never breaks
```

Reassemble STYLE.MAC and relink as usual.

NEW USE FOR VIRTUAL MEMORY (VM) HANDLER
FAST-FILE MAPPED MEMORY INPUT/OUTPUT (I/O)
FOR RT-11* OPERATING SYSTEM

Tillman B. Smith and David G. Piskura

UNITED TECHNOLOGIES RESEARCH CENTER
OPTICS AND APPLIED TECHNOLOGY LABORATORY
P.O. Drawer 4181
West Palm Beach, Florida 33402

28-Jun-85, Page 1 of 14

ABSTRACT

In real time applications that are I/O bound between local computers, the VM handler can be used in a different manner to increase the I/O speed between computers. The VM handler organizes the virtual memory into files, and the use of dual-ported virtual memory provides the local computers with fast data transfer capability. Each computer is assigned its' own write-only and read-only files for I/O data file exchange. The file then becomes the basic data packet in the local information exchange between computer systems.

I. ORIGINAL PURPOSE OF VIRTUAL MEMORY (VM) HANDLER

The original purpose of the VIRTUAL MEMORY (VM) HANDLER is to increase computer speed by eliminating hard disk I/O operations for program overlays. Many applications run on 16-bit word computers that require more than 64K-bytes of memory to contain both Application Program (AP) and Operating System (OS) at the same time. This AP requires hard disk I/O operations to load AP and OS segments of convenient size from disk to be time shared in the the first 64K bytes of direct addressable memory. These I/O operations from the hard disk are much slower than memory-to-memory operations.

Memory Management Units (MMU) were added to increase the computer's memory capacity to 4 Megabyte with 22-bit addressing. The first requirement to speed-up the computer's operation is to load both the OS and AP into virtual memory (memory above the first 64K bytes) at boot time, eliminating further I/O operations to the hard disk, since all the AP and OS now reside in virtual memory. The AP and OS segments are now swapped in and out of the first 64K bytes of memory using the faster memory-to-memory operations from the virtual memory.

*Registered trade marks of DIGITAL EQUIPMENT CORPORATION (DEC).

I. PURPOSE Continued

The VM handler organizes the virtual memory into files just as if they were on the hard disk. The VM file directory is complete with the file names, files size in blocks, and blocks remaining in the virtual-memory section under VM control. This facilitates the loading of the AP and OS into virtual memory, since it is already file structured by the linker onto the hard disk.

II. NEW APPLICATION FOR VM HANDLER

In real time applications which are I/O bound between local computers, the VM handler can be used in a different manner to increase the I/O speed. Since the VM handler organizes the virtual memory into files, dual-ported virtual memory provides the local computers with fast file exchange capability. Each computer is assigned its' own write-only and read-only files for data I/O or in this case I/O file exchange. The file then becomes the basic data packet in the local information exchange system.

Of course this approach is limited to computers with the same file structure and/or from the same manufacturer.

III. SAMPLE PROGRAMS

The sample programs and data files in Figures 1 through 7 were generated on the PDP-11/23+* Computer running the RT-11 OS, with 512K bytes of memory, and written in the high-level FORTRAN IV language.

The text file in Figure 1 contains the information, for listing on the terminal, concerning the command file. The VM command file shown in Figure 2, can be appended to the start-up command file or run separately as a command file to allocate the desired file size in virtual memory.

To determine the VM starting address (offset in octal), used in the command file, take the physical memory start address (in octal) and drop the two-least significant digits. The VM default starting address is 1000000 (octal) (the beginning of the second 128K words of the total 256K-word memory available). VM will use the memory up to the 4K words of the I/O page at the top of the virtual memory. In this example in Figure 2, the two (2) files named TEST1.DAT and TEST3.DAT are created in VM with 96 blocks each to run a test case.

*Registered trade marks of DIGITAL EQUIPMENT CORPORATION (DEC).

FILE: VMXMDI.TXT

Date: 27-Jun-85

VMXMDI.COM is the command file used to set the Virtual Memory (VM) to begin at your selected physical address (PA). VM is installed at boot time, at a default physical base address (PA) of 1000000 (octal), at the top of first 256K bytes.

The new base address is determined by taking your selected PA and dividing by 100. Then a SET command for VM BASE equal to this number is done.

FIGURE 1. THE TEXT FILE FOR VMXMDI COMMAND FILE

III. SAMPLE PROGRAMS Continued

The FORTRAN program in Figure 3, generates two (2) 256-word arrays. This length was chosen to match the array length to the one (1) block length on the disk, for transfer speed improvement to and from the hard disk. The first array is numbered in reverse sequence with block number in place of the last word in the array. The second array is numbered in sequence with block number in place of the first word in the array. The data made from 96 of these arrays is written into TEST1.DAT file in VM and then to the system hard disk into file TEST2.DAT and back to the TEST3.DAT file in VM. After the two (2) files in VM are written and read, a comparison is made on the data to verify that it is the same.

These two files, TEST1.DAT and TEST3.DAT may be copied to and from other files in VM or the system hard disk, DK. These files can be operated on with the Command Language. The Directory Command shows the file size, and the free blocks remaining in the VM, and the DUMP Command prints out the data, and the file can be deleted.

Figure 4 contains the compiled code from Figure 3. Figures 5 through 7 contain the data from the last block of each file to show the data was transferred as stated.

IV. SUMMARY AND GOAL

The implementation and use of this idea of data file exchanges through VM between local computers is left for the application programmer. This paper shows how to structure the VM into files at your chosen PA, and to copy files to and from VM. The goal of this paper is to save the programmers time in implementing their application programs on dual computer systems.

```

TYPE VMXMDI.TXT
!           File: VMXMDI.COM           Date: 27-Jun-85
!
! The command to run this file is @VMXMDI <RETURN>.
!
! VMXMDI.COM is the command file used to set the Virtual Memory
! (VM) to begin at your selected physical address (PA). VM is
! installed at boot time, at a default physical base address
! (PA) of 1000000 (octal), at the top of first 256K bytes.
!
! The new base (offset) address is determined by taking the
! PA and dividing by 100. Then a SET command for VM BASE
! equal to this number is done.
!
SET VM BASE=7000           !MY SELECTED OFFSET ADDRESS
! SET VM BASE=10000       !DEFAULT OFFSET ADDRESS
!
! VM Handler is reinstalled and loaded below the new base
! address.
UNLOAD VM:
!
REMOVE VM:
!
INSTALL VM:
!
LOAD VM:
!
INIT/NOQ VM:
!
CREATE VM:TEST1.DAT/START:0/ALLOCATE:96.
CREATE VM:TEST3.DAT/START:0/ALLOCATE:96.
!
SHOW DEVICE
!
DATE
!
TIME
!
RUN VMDCMP                !RUN TEST PROGRAM
!                        !TIME ~1 1/2 MINUTES
DUMP/OU/TE/ON:95. VM:TEST1.DAT !DISPLAYS LAST
!                        !BLOCK 95 FILE 1
DUMP/OU/TE/ON:95. VM:TEST3.DAT !DISPLAYS LAST
!                        !BLOCK 95 FILE 3
COPY DK:TEST4.DAT VM:TEST3.DAT !COPIES FILE 4 ON SYSTEM
!                        !DISK TO FILE 3 ON VM.
DUMP/OU/TE/ON:95. VM:TEST3.DAT !DISPLAYS LAST BLOCK 95
!                        !IN FILE 3 AGAIN
! NOTICE THE ORDER OF THE NUMBERS AND LOCATION OF THE
! BLOCK NUMBERS CHANGED WHEN DK:TEST4.DAT WAS COPIED
! TO VM:TEST3.DAT.
    
```

FIGURE 2. COMMAND FILE VMXMDI

```

PROGRAM VMDCMP
C
C FILE NAME: VMDCMP.FOR
C
C MODULE NAME: VMDCMP VERSION 1.0 DATE: 06-JUN-85
C
C UNITED TECHNOLOGIES RESEARCH CENTER
C OPTICS AND APPLIED TECHNOLOGY LABORATORY
C WEST PALM BEACH, FLORIDA
C
C AUTHOR: TILLMAN B. SMITH DATE: 07-MAY-85
C
C MODIFIED BY: TILLMAN B. SMITH DATE: 15-MAY-85
C
C LANGUAGE: FORTRAN IV, V2.6
C
C REQUIRED HARDWARE:
C PDP-11/23+ PROCESSOR, RT-11 OPERATING SYSTEM, V5.1
C MEMORY 512 KBYTES, EXTENDED MONITOR (XM), VIRTUAL
C MEMORY (VM) HANDLER.
C
C FUNCTIONS:
C GENERATES 'A' ARRAY IN REVERSE ORDER, AND 'F' ARRAY
C IN ORDER. WRITES 'A' ARRAY WITH BLOCK NUMBERS INTO
C A FILE ON VM AND THE SYSTEM DISK, DK. WRITES THE
C 'F' ARRAY INTO A FILE ON DK. THE DATA IN 'A' ARRAY
C IS WRITTEN BACK TO VM. READS THE TWO FILES FROM VM
C INTO 'C' ARRAY AND 'D' ARRAY BLOCK BY BLOCK, AND
C COMPARES EACH ARRAY WORD BY WORD FOR THE 96 BLOCKS.
C
C INPUT VARIABLES: NONE
C
C OUTPUT VARIABLES: NONE
C
C INPUT FILES:
C VM:TEST1.DAT, 96 BLOCKS (32K 16-BIT WORDS)
C DK:TEST2.DAT, 96 BLOCKS (32K 16-BIT WORDS)
C VM:TEST3.DAT, 96 BLOCKS (32K 16-BIT WORDS)
C DK:TEST4.DAT, 96 BLOCKS (32K 16-BIT WORDS)
C
C OUTPUT FILES:
C VM:TEST1.DAT, 96 BLOCKS (32K 16-BIT WORDS)
C DK:TEST2.DAT, 96 BLOCKS (32K 16-BIT WORDS)
C VM:TEST3.DAT, 96 BLOCKS (32K 16-BIT WORDS)
C DK:TEST4.DAT, 96 BLOCKS (32K 16-BIT WORDS)
C
C EXTERNAL REFERENCES: NONE
C
C LOCAL VARIABLES: A,B,C,D,E,F,NREC,I,J,N,M
    
```

FIGURE 3. THE TEST OF GENERATION AND COMPARISON OF FILES
Continued

FIGURE 3. THE TEST OF GENERATION AND COMPARISON OF FILES
Continued

```

C PROCESS DESCRIPTION:
C GENERATES 'A' ARRAY (256 WORDS) IN REVERSE ORDER
C AND NUMBERS THE LAST WORD WITH THE BLOCK NUMBER.
C GENERATES 'F' ARRAY IN ORDER AND NUMBERS THE FIRST
C WORD WITH THE BLOCK NUMBER. WRITES 'A' ARRAY
C 96 TIMES INTO VM:TEST1.DAT FILE. READS TEST1
C FILE INTO 'B' ARRAY A BLOCK AT A TIME. WRITES 'B'
C AND 'F' ARRAYS INTO DK:TEST2.DAT AND DK:TEST4.DAT
C RESPECTIVELY. READS TEST2.DAT INTO 'C' ARRAY AND
C WRITES 'C' ARRAY INTO VM:TEST3.DAT. READS
C TEST3.DAT FILE INTO 'D' ARRAY AND READS TEST1 FILE
C INTO 'C' ARRAY A BLOCK AT A TIME AND COMPARES 'C'
C ARRAY WITH 'D' ARRAY FOR EACH OF THE 96 BLOCKS,
C WORD BY WORD, FOR THE ENTIRE FILE.
C
C DATA TYPES:
C
C INTEGER*2 A,B,C,D,E,F,NREC,I,J,N,M
C DIMENSION A(256),B(256),C(256),D(256),E(256),F(256)
C N=256 !ARRAY SIZE
C M=96 !TOTAL NUMBER OF BLOCKS
C
C OPEN (UNIT=1,NAME='VM:TEST1.DAT',ACCESS='DIRECT',
C 1 FORM='UNFORMATTED',RECORDSIZE=128, !BLOCKSIZE=512,
C 2 TYPE='UNKNOWN') !ERR=14
C
C OPEN (UNIT=2,NAME='DK:TEST2.DAT',ACCESS='DIRECT',
C 1 FORM='UNFORMATTED',RECORDSIZE=128, !BLOCKSIZE=512,
C 2 TYPE='UNKNOWN',ERR=14)
C
C OPEN (UNIT=3,NAME='VM:TEST3.DAT',ACCESS='DIRECT',
C 1 FORM='UNFORMATTED',RECORDSIZE=128, !BLOCKSIZE=512,
C 2 TYPE='UNKNOWN',ERR=14)
C
C OPEN (UNIT=4,NAME='DK:TEST4.DAT',ACCESS='DIRECT',
C 1 FORM='UNFORMATTED',RECORDSIZE=128, !BLOCKSIZE=512,
C 2 TYPE='UNKNOWN',ERR=14)
C
C DO 20 I=1,256
C A(I)=256-I !GENERATES REVERSE ORDER 255-0
C F(I)=I-1 !GENERATES F ARRAY 0-255
20 CONTINUE
    
```

FIGURE 3. THE TEST OF GENERATION AND COMPARISON OF FILES
Continued

FIGURE 3. THE TEST OF GENERATION AND COMPARISON OF FILES
Continued

```

C
C DO 30 NREC=1,M !GENERATES 96 RECORDS SETS
C A(256)=NREC-1 !OVERWRITES LAST WORD
C !WITH RECORD NUMBER
C WRITE (1'NREC) A
C READ (1'NREC) B
C WRITE (2'NREC) B
C F(1)=NREC-1 !OVERWRITES FIRST WORD
C !WITH RECORD NUMBER
C WRITE (4'NREC) F
C READ (2'NREC) C
C WRITE (3'NREC) C
30 CONTINUE
C DO 40 NREC=1,M
C READ (1'NREC) D
C READ (3'NREC) E
C DO 50 I=1,N
C IF (D(I) .NE. E(I)) GOTO 130
50 CONTINUE
40 CONTINUE
C
C 150 CLOSE (UNIT=1,ERR=99)
C CLOSE (UNIT=2,ERR=99)
C CLOSE (UNIT=3,ERR=99)
C CLOSE (UNIT=4,ERR=99)
C
C GOTO 110
14 WRITE (5,70)
C GOTO 300
130 WRITE (5,140)
C WRITE (5,100) (I,A(I),I,D(I))
C GOTO 150
99 WRITE (5,90)
C GOTO 300
110 WRITE (5,120)
120 FORMAT (T15,' WR/RE AND COMPARISON OF TWO FILES COMPLETED')
70 FORMAT (' * ERROR ON OPENING FILE *')
90 FORMAT (' * ERROR ON CLOSING FILE *')
140 FORMAT (' * ERROR ON COMPARISON OF A AND D *')
100 FORMAT (' A(',I3,',') = ',I3,',', D(',I3,',') = ',I3)
300 STOP
END
    
```

FIGURE 3. THE TEST OF GENERATION AND COMPARISON OF FILES

```

FORTRAN IV      V02.6      Thu 06-Jun-85 07:13:08      PAGE 001
0001      PROGRAM      VMDCMP

C
C FILE NAME:      VMDCMP.FOR
C
C MODULE NAME:    VMDCMP      VERSION 1.0      DATE: 06-JUN-85
C
C UNITED TECHNOLOGIES RESEARCH CENTER
C OPTICS AND APPLIED TECHNOLOGY LABORATORY
C WEST PALM BEACH, FLORIDA
C
C AUTHOR:         TILLMAN B. SMITH      DATE: 07-MAY-85
C
C MODIFIED BY:    TILLMAN B. SMITH      DATE: 15-MAY-85
C
C LANGUAGE:       FORTRAN IV, V2.6
C
C REQUIRED HARDWARE:
C PDP-11/23+ PROCESSOR, RT-11 OPERATING SYSTEM, V5.1
C MEMORY 512 KBYTES, EXTENDED MONITOR (XM), VIRTUAL
C MEMORY (VM) HANDLER.
C
C FUNCTIONS:
C GENERATES 'A' ARRAY IN REVERSE ORDER, AND 'F' ARRAY
C IN ORDER. WRITES 'A' ARRAY WITH BLOCK NUMBERS INTO
C A FILE ON VM AND THE SYSTEM DISK, DK. WRITES THE
C 'F' ARRAY INTO A FILE ON DK. THE DATA IN 'A' ARRAY
C IS WRITTEN BACK TO VM. READS THE TWO FILES FROM VM
C INTO 'C' ARRAY AND 'D' ARRAY BLOCK BY BLOCK, AND
C COMPARES EACH ARRAY WORD BY WORD FOR THE 96 BLOCKS.
C
C INPUT VARIABLES:      NONE
C
C OUTPUT VARIABLES:     NONE
C
C INPUT FILES:
C VM:TEST1.DAT, 96 BLOCKS (32K 16-BIT WORDS)
C DK:TEST2.DAT, 96 BLOCKS (32K 16-BIT WORDS)
C VM:TEST3.DAT, 96 BLOCKS (32K 16-BIT WORDS)
C DK:TEST4.DAT, 96 BLOCKS (32K 16-BIT WORDS)
C
C OUTPUT FILES:
C VM:TEST1.DAT, 96 BLOCKS (32K 16-BIT WORDS)
C DK:TEST2.DAT, 96 BLOCKS (32K 16-BIT WORDS)
C VM:TEST3.DAT, 96 BLOCKS (32K 16-BIT WORDS)
C DK:TEST4.DAT, 96 BLOCKS (32K 16-BIT WORDS)
C
C EXTERNAL REFERENCES:  NONE
C
C LOCAL VARIABLES:      A,B,C,D,E,F,NREC,I,J,N,M
C
C PROCESS DESCRIPTION:
C GENERATES 'A' ARRAY (256 WORDS) IN REVERSE ORDER
C AND NUMBERS THE LAST WORD WITH THE BLOCK NUMBER.
C GENERATES ARRAY IN ORDER AND NUMBERS THE FIRST

```

FIGURE 4. GENERATION AND COMPARISON OF FILES COMPILED CODE Continued

```

FORTRAN IV      V02.6      Thu 06-Jun-85 07:13:08      PAGE 002
C
C WORD WITH THE BLOCK NUMBER. WRITES 'A' ARRAY
C 96 TIMES INTO VM:TEST1.DAT FILE. READS TEST1
C FILE INTO 'B' ARRAY A BLOCK AT A TIME. WRITES 'B'
C AND 'F' ARRAYS INTO DK:TEST2.DAT AND DK:TEST4.DAT
C RESPECTIVELY. READS TEST2.DAT INTO 'C'
C ARRAY AND WRITES 'C' ARRAY INTO VM:TEST3.DAT.
C READS TEST3.DAT FILE INTO 'D' ARRAY AND READS TEST1
C FILE INTO 'C' ARRAY A BLOCK AT A TIME AND COMPARES 'C'
C ARRAY WITH 'D' ARRAY FOR EACH OF THE 96 BLOCKS, WORD
C BY WORD, FOR THE ENTIRE FILE.
C
C DATA TYPES:
C
0002      INTEGER*2      A,B,C,D,E,F,NREC,I,J,N,M
0003      DIMENSION      A(256),B(256),C(256),D(256),E(256),F(256)
0004      N=256          !ARRAY SIZE
0005      M=96          !TOTAL NUMBER OF BLOCKS
C
0006      OPEN (UNIT=1,NAME='VM:TEST1.DAT',ACCESS='DIRECT',
1 FORM='UNFORMATTED',RECORDSIZE=128, !BLOCKSIZE=512,
2 TYPE='UNKNOWN') !ERR=14
C
0007      OPEN (UNIT=2,NAME='DK:TEST2.DAT',ACCESS='DIRECT',
1 FORM='UNFORMATTED',RECORDSIZE=128, !BLOCKSIZE=512,
2 TYPE='UNKNOWN',ERR=14)
C
0008      OPEN (UNIT=3,NAME='VM:TEST3.DAT',ACCESS='DIRECT',
1 FORM='UNFORMATTED',RECORDSIZE=128, !BLOCKSIZE=512,
2 TYPE='UNKNOWN',ERR=14)
C
0009      OPEN (UNIT=4,NAME='DK:TEST4.DAT',ACCESS='DIRECT',
1 FORM='UNFORMATTED',RECORDSIZE=128, !BLOCKSIZE=512,
2 TYPE='UNKNOWN',ERR=14)
C
0010      DO 20 I=1,256
0011          A(I)=256-I !GENERATES REVERSE ORDER 255-0
0012          F(I)=I-1 !GENERATES F ARRAY 0-255
0013 20 CONTINUE
C
0014      DO 30 NREC=1,M !GENERATES 96 RECORDS SETS
0015          A(256)=NREC-1 !OVERWRITES LAST WORD
C !WITH RECORD NUMBER
0016          WRITE (1'NREC) A
0017          READ (1'NREC) B
0018          WRITE (2'NREC) B
0019          F(1)=NREC-1 !OVERWRITES FIRST WORD
C !WITH RECORD NUMBER
0020          WRITE (4'NREC) F
0021          READ (2'NREC) C
0022          WRITE (3'NREC) C
0023 30 CONTINUE
0024      DO 40 NREC=1,M
0025          READ (1'NREC) D
0026          READ (3'NREC) E

```

FIGURE 4. GENERATION AND COMPARISON OF FILES COMPILED CODE Continued

FIGURE 4. GENERATION AND COMPARISON OF FILES COMPILED CODE Continued

```

FORTRAN IV      VQ2.6      Thu 06-Jun-85 07:13:08      PAGE 003
0027           DO 50 I=1,N
0028           IF (D(I) .NE. E(I)) GOTO 130
0030 50        CONTINUE
0031 40        CONTINUE
C
0032 150       CLOSE (UNIT=1,ERR=99)
0033          CLOSE (UNIT=2,ERR=99)
0034          CLOSE (UNIT=3,ERR=99)
0035          CLOSE (UNIT=4,ERR=99)
C
0036          GOTO 110
0037 14        WRITE (5,70)
0038          GOTO 300
0039 130       WRITE (5,140)
0040          WRITE (5,100) (I,A(I),I,D(I))
0041          GOTO 150
0042 99        WRITE (5,90)
0043          GOTO 300
0044 110       WRITE (5,120)
0045 120       FORMAT (T15,' WR/RE AND COMPARISON OF TWO FILES COMPLETED')
0046 70        FORMAT (' * ERROR ON OPENING FILE *')
0047 90        FORMAT (' * ERROR ON CLOSING FILE *')
0048 140       FORMAT (' * ERROR ON COMPARISON OF A AND D *')
0049 100       FORMAT (' A(' ,I3,' ) = ' ,I3,' , D(' ,I3,' ) = ' ,I3)
0050 300       STOP
0051          END
FORTRAN IV      Storage Map for Program Unit VMDCMP
Local Variables, .PSECT $DATA, Size = 006022 ( 1545. words)

Name  Type  Offset      Name  Type  Offset      Name  Type  Offset

I     I*2  006012      J     I*2  006014      M     I*2  006020

N     I*2  006016      NREC I*2  006010

Local and COMMON Arrays:

Name  Type  Section Offset  -----Size-----  Dimensions
A     I*2  $DATA  000000  001000 ( 256.) (256)
B     I*2  $DATA  001000  001000 ( 256.) (256)
C     I*2  $DATA  002000  001000 ( 256.) (256)
D     I*2  $DATA  003000  001000 ( 256.) (256)
E     I*2  $DATA  004000  001000 ( 256.) (256)
F     I*2  $DATA  005000  001000 ( 256.) (256)
    
```

FIGURE 4. GENERATION AND COMPARISON OF FILES COMPILED CODE

```

.DUMP/OU/TE/ON:95.      VM:TEST1.DAT
VM:TEST1.DAT
BLOCK NUMBER 000137
000/ 000377 000376 000375 000374 000373 000372 000371 000370 *..}.|.f.z.y.x.*
020/ 000367 000366 000365 000364 000363 000362 000361 000360 *w.v.u.t.s.r.q.p.*
040/ 000357 000356 000355 000354 000353 000352 000351 000350 *o.n.m.l.k.j.i.h.*
060/ 000347 000346 000345 000344 000343 000342 000341 000340 *g.f.e.d.c.b.a.'. *
100/ 000337 000336 000335 000334 000333 000332 000331 000330 *_.^].\.[.Z.V.Y.X.*
120/ 000327 000326 000325 000324 000323 000322 000321 000320 *W.V.U.T.S.R.Q.P.*
140/ 000317 000316 000315 000314 000313 000312 000311 000310 *O.N.M.L.K.J.I.H.*
160/ 000307 000306 000305 000304 000303 000302 000301 000300 *G.F.E.D.C.B.A.@.*
200/ 000277 000276 000275 000274 000273 000272 000271 000270 *?.>.=.<.;.:9.8.*
220/ 000267 000266 000265 000264 000263 000262 000261 000260 *7.6.5.4.3.2.1.0.*
240/ 000257 000256 000255 000254 000253 000252 000251 000250 */...-.+.*)).(.*
260/ 000247 000246 000245 000244 000243 000242 000241 000240 *'&.%$.#."! . *
300/ 000237 000236 000235 000234 000233 000232 000231 000230 *.....*
320/ 000227 000226 000225 000224 000223 000222 000221 000220 *.....*
340/ 000217 000216 000215 000214 000213 000212 000211 000210 *.....*
360/ 000207 000206 000205 000204 000203 000202 000201 000200 *.....*
400/ 000177 000176 000175 000174 000173 000172 000171 000170 *..}.|.f.z.y.x.*
420/ 000167 000166 000165 000164 000163 000162 000161 000160 *w.v.u.t.s.r.q.p.*
440/ 000157 000156 000155 000154 000153 000152 000151 000150 *o.n.m.l.k.j.i.h.*
460/ 000147 000146 000145 000144 000143 000142 000141 000140 *g.f.e.d.c.b.a.'. *
500/ 000137 000136 000135 000134 000133 000132 000131 000130 *_.^].\.[.Z.V.Y.X.*
520/ 000127 000126 000125 000124 000123 000122 000121 000120 *W.V.U.T.S.R.Q.P.*
540/ 000117 000116 000115 000114 000113 000112 000111 000110 *O.N.M.L.K.J.I.H.*
560/ 000107 000106 000105 000104 000103 000102 000101 000100 *G.F.E.D.C.B.A.@.*
600/ 000077 000076 000075 000074 000073 000072 000071 000070 *?.>.=.<.;.:9.8.*
620/ 000067 000066 000065 000064 000063 000062 000061 000060 *7.6.5.4.3.2.1.0.*
640/ 000057 000056 000055 000054 000053 000052 000051 000050 */...-.+.*)).(.*
660/ 000047 000046 000045 000044 000043 000042 000041 000040 *'&.%$.#."! . *
700/ 000037 000036 000035 000034 000033 000032 000031 000030 *.....*
720/ 000027 000026 000025 000024 000023 000022 000021 000020 *.....*
740/ 000017 000016 000015 000014 000013 000012 000011 000010 *.....*
760/ 000007 000006 000005 000004 000003 000002 000001 000137 *.....*
    
```

FIGURE 5. TEST1.DAT FILE DATA

```
.DUMP/OU/TE/ON:95.      VM:TEST3.DAT
VM:TEST3.DAT
BLOCK NUMBER 000137
000/ 000377 000376 000375 000374 000373 000372 000371 000370 *.,~.}.|.|{.z.y.x.*
020/ 000367 000366 000365 000364 000363 000362 000361 000360 *w.v.u.t.s.r.q.p.*
040/ 000357 000356 000355 000354 000353 000352 000351 000350 *o.n.m.l.k.j.i.h.*
060/ 000347 000346 000345 000344 000343 000342 000341 000340 *g.f.e.d.c.b.a.`.*
100/ 000337 000336 000335 000334 000333 000332 000331 000330 *.,^.]|\.[Z.Y.X.*
120/ 000327 000326 000325 000324 000323 000322 000321 000320 *W.V.U.T.S.R.Q.P.*
140/ 000317 000316 000315 000314 000313 000312 000311 000310 *O.N.M.L.K.J.I.H.*
160/ 000307 000306 000305 000304 000303 000302 000301 000300 *G.F.E.D.C.B.A.@.*
200/ 000277 000276 000275 000274 000273 000272 000271 000270 *?.>.=.<.;:~.9.8.*
220/ 000267 000266 000265 000264 000263 000262 000261 000260 *7.6.5.4.3.2.1.0.*
240/ 000257 000256 000255 000254 000253 000252 000251 000250 */.-.,;+.*.)(.(*
260/ 000247 000246 000245 000244 000243 000242 000241 000240 *'&.%$.#.'!.*
300/ 000237 000236 000235 000234 000233 000232 000231 000230 *.....*
320/ 000227 000226 000225 000224 000223 000222 000221 000220 *.....*
340/ 000217 000216 000215 000214 000213 000212 000211 000210 *.....*
360/ 000207 000206 000205 000204 000203 000202 000201 000200 *.....*
400/ 000177 000176 000175 000174 000173 000172 000171 000170 *.,~.}.|.|{.z.y.x.*
420/ 000167 000166 000165 000164 000163 000162 000161 000160 *w.v.u.t.s.r.q.p.*
440/ 000157 000156 000155 000154 000153 000152 000151 000150 *o.n.m.l.k.j.i.h.*
460/ 000147 000146 000145 000144 000143 000142 000141 000140 *g.f.e.d.c.b.a.`.*
500/ 000137 000136 000135 000134 000133 000132 000131 000130 *.,^.]|\.[Z.Y.X.*
520/ 000127 000126 000125 000124 000123 000122 000121 000120 *W.V.U.T.S.R.Q.P.*
540/ 000117 000116 000115 000114 000113 000112 000111 000110 *O.N.M.L.K.J.I.H.*
560/ 000107 000106 000105 000104 000103 000102 000101 000100 *G.F.E.D.C.B.A.@.*
600/ 000077 000076 000075 000074 000073 000072 000071 000070 *?.>.=.<.;:~.9.8.*
620/ 000067 000066 000065 000064 000063 000062 000061 000060 *7.6.5.4.3.2.1.0.*
640/ 000057 000056 000055 000054 000053 000052 000051 000050 */.-.,;+.*.)(.(*
660/ 000047 000046 000045 000044 000043 000042 000041 000040 *'&.%$.#.'!.*
700/ 000037 000036 000035 000034 000033 000032 000031 000030 *.....*
720/ 000027 000026 000025 000024 000023 000022 000021 000020 *.....*
740/ 000017 000016 000015 000014 000013 000012 000011 000010 *.....*
760/ 000007 000006 000005 000004 000003 000002 000001 000137 *.....*
```

FIGURE 6. TEST3.DAT FILE DATA

```
.DUMP/OU/TE/ON:95.      DK:TEST4.DAT
DK:TEST4.DAT
BLOCK NUMBER 000137
000/ 000137 000001 000002 000003 000004 000005 000006 000007 *.....*
020/ 000010 000011 000012 000013 000014 000015 000016 000017 *.....*
040/ 000020 000021 000022 000023 000024 000025 000026 000027 *.....*
060/ 000030 000031 000032 000033 000034 000035 000036 000037 *.....*
100/ 000040 000041 000042 000043 000044 000045 000046 000047 *!."#.%&'.*
120/ 000050 000051 000052 000053 000054 000055 000056 000057 *(.)*.+.,,-./.*
140/ 000060 000061 000062 000063 000064 000065 000066 000067 *0.1.2.3.4.5.6.7.*
160/ 000070 000071 000072 000073 000074 000075 000076 000077 *8.9.:;.<.=.>.*
200/ 000100 000101 000102 000103 000104 000105 000106 000107 *@.A.B.C.D.E.F.G.*
220/ 000110 000111 000112 000113 000114 000115 000116 000117 *H.I.J.K.L.M.N.O.*
240/ 000120 000121 000122 000123 000124 000125 000126 000127 *P.Q.R.S.T.U.V.W.*
260/ 000130 000131 000132 000133 000134 000135 000136 000137 *X.Y.Z.[\].^._.*
300/ 000140 000141 000142 000143 000144 000145 000146 000147 *'a.b.c.d.e.f.g.*
320/ 000150 000151 000152 000153 000154 000155 000156 000157 *h.i.j.k.l.m.n.o.*
340/ 000160 000161 000162 000163 000164 000165 000166 000167 *p.q.r.s.t.u.v.w.*
360/ 000170 000171 000172 000173 000174 000175 000176 000177 *x.y.z.{|}.~...*
400/ 000200 000201 000202 000203 000204 000205 000206 000207 *.....*
420/ 000210 000211 000212 000213 000214 000215 000216 000217 *.....*
440/ 000220 000221 000222 000223 000224 000225 000226 000227 *.....*
460/ 000230 000231 000232 000233 000234 000235 000236 000237 *.....*
500/ 000240 000241 000242 000243 000244 000245 000246 000247 *!."#.%&'.*
520/ 000250 000251 000252 000253 000254 000255 000256 000257 *(.)*.+.,,-./.*
540/ 000260 000261 000262 000263 000264 000265 000266 000267 *0.1.2.3.4.5.6.7.*
560/ 000270 000271 000272 000273 000274 000275 000276 000277 *8.9.:;.<.=.>.*
600/ 000300 000301 000302 000303 000304 000305 000306 000307 *@.A.B.C.D.E.F.G.*
620/ 000310 000311 000312 000313 000314 000315 000316 000317 *H.I.J.K.L.M.N.O.*
640/ 000320 000321 000322 000323 000324 000325 000326 000327 *P.Q.R.S.T.U.V.W.*
660/ 000330 000331 000332 000333 000334 000335 000336 000337 *X.Y.Z.[\].^._.*
700/ 000340 000341 000342 000343 000344 000345 000346 000347 *'a.b.c.d.e.f.g.*
720/ 000350 000351 000352 000353 000354 000355 000356 000357 *h.i.j.k.l.m.n.o.*
740/ 000360 000361 000362 000363 000364 000365 000366 000367 *p.q.r.s.t.u.v.w.*
760/ 000370 000371 000372 000373 000374 000375 000376 000377 *x.y.z.{|}.~...*
```

FIGURE 7. TEST3.DAT FILE CONTAINING DATA FROM SYSTEM DISK

NOTE

All of this material is contained on the attached 5 1/4-inch diskette, single-sided, quad-density, and 96 tracks per inch, which is the DEC standard format for the PRO-350 Computer with the Professional Operating System (P/OS). This material was written and revised with the EDT editor, and printed on a DEC Model LA50 Printer.

Annotated Directory
RT-11 SIG Tape, Spring 1985

The Spring, 1985, RT SIG tape contains 64 Files, 17783 Blocks.

It was prepared by:

R. W. Barnard
Sandia National Laboratories
Division 7523
P. O. Box 5800
Albuquerque, NM 87185

DCS - BARNARD

It is distributed by:

Thomas J. Shinal
General Scientific Corp.
1681 East Gude Dr.
Rockville MD 20850
(301) 340-2773

DCS - SHINAL

DECUS Symposium RT-11 SIG Tape

Spring, 1985
New Orleans, LA

Annotated Directory

IMPORTANT

Read the file, README.1ST, first.

README.1ST 10 01-Jul-85 SIG tape copy instructions
and new information for everyone.

NOTE! We are interested in maintaining the quality of the submissions to the RT SIG tape. Therefore, we welcome feedback regarding your use of these files, any bugs you find, and any bug fixes or improvements you devise. Please send any correspondence regarding the tape to:

John Crowell
Crowell Ltd.*
145 Andanada
Los Alamos, NM 87544
(505) 662-3893

* (But not very)

DCS - CROWELL

SPRING, 1985 RT SIG TAPE DIRECTORY

VIRTUL - Subdevice retriever for RSTS.

E.F.Beadel, Jr., Manager
LAUSE Instructional Computer Center
SUNY at Oswego
Oswego, NY 13126
(315) 341-3055

This program allows RSTS/E users to break down the subdevice files from this tape after they have been copied to disk. It has been modified by David Smith, Galileo Computer Center, to remove a few bugs and to be able to read multi-segment directories. See README.1ST for details.

VIRTUL.BAS 1 File, 43 Blocks

DIR1 - Annotated tape directories, part 1.
DIR2 - Annotated tape directories, part 2.

R. W. Barnard
Sandia National Laboratories
Minicomputer Software Division 7523
P. O. Box 5800
Albuquerque, NM 87185
(505) 844-5115

DIR1 has annotated directories from Spring, 1978, through Spring, 1981. It is being included again on this tape so that the directory files will be automatically indexed.

DIR2 contains annotated directories of the DECUS Symposia RT-11 tapes from the Fall of 1981 through the Spring of 1985 (this symposium).

DIR2.DSK 8 Files, 385 Blocks
DIR1.DSK 7 Files, 485 Blocks

F77KIT - FORTRAN-77 OTS Update.

Robert Walraven
Multiware, Inc.
139 G Street, Suite 161
Davis, CA 95616
(916) 756-3291

This is an upgrade kit for FORTRAN-77/RT, Version 5.0 (i.e., Release 1). It includes improvements for virtual support, and a few bug fixes. F77OTS.OBJ can be updated to give improved performance when "virtual" support is needed. You can now also access the I/O page while using virtual arrays by means of a patch to a newly-defined global.

F77KIT.DSK 8 Files, 28 Blocks

SPRING, 1985 RT SIG TAPE DIRECTORY

IMAGE* - Image Display Program for the PRO.

John M. Crowell
Crowell, Ltd.
145 Andanada
Los Alamos, NM 87544
(505) 662-3893

This program displays image files on the PRO-3xx's bit-mapped display. It is a variation on the original program by Greg Adams and David Fingerhut of DEC. In most cases the options and their defaults are the same as in the original program. The images produced from the image files may be either monochrome or color.

IMAGE1.DSK 4 Files, 242 Blocks
IMAGE2.DSK 4 Files, 480 Blocks

INCBUP - IND Control Files.

R. W. Barnard
Sandia National Laboratories
Minicomputer Software Division 7523
P. O. Box 5800
Albuquerque, NM 87185
(505) 844-5115

INCBUP provides for an incremental (since the last backup) facility for RT-11 users. UP, DOWN, HOME allow movement among subdevices. When you go DOWN to a filespec, it becomes your default device. CUR reports your current whereabouts (with respect to subdevices).

INCBUP.DSK 8 Files, 54 Blocks

LDUDK - Load VT200 User-Defined Keys.

Stephen Cribbs
Atomic Energy of Canada Limited
Pinawa, Manitoba
Canada R0E 1L0
(204) 753-2311

The LDUDK (Load the User Defined Keys) program provides RT-11 and TSX+ users with a convenient method for programming the VT200 series terminal's keyboard. The submission also includes the file TCFL.CSL, which is a library of Terminal Control Functions configured for the program's console terminal.

LDUDK.DSK 11 Files, 129 Blocks

SPRING, 1985 RT SIG TAPE DIRECTORY

KER* - KERMIT File transfer protocol.
K11*.HEX

Brian Nelson
Computer Services, University of Toledo
2801 West Bancroft
Toledo, OH 43606
(419) 537-2841

This is release 2.23 of Kermit-11. It requires RSTS Version 7.2 or later, RSX11M v4.0 or later, or RSX11M Plus Version 2.0 or later, or RT11 Version 4.0 or later, or P/DS Version 2.0 or PRO/RT11 Version 5.1. See the files K11AAA.AAA, and K11INS.DOC for more information.

The distribution on this tape differs from previous distributions by having bug fixes for TSX+ and RT-11. Edit history is given in the file K11CMD.MAC.

Minimum system requirements to run Kermit:

RT11 (excluding the PRO/350).
This Version of Kermit will run on RT11 Version 4.0 or later as long as the monitor has multiple terminal support generated. The use of this feature enables the user to access DZ11's as well as DL11's. If you are using XM then you should use K11XM, otherwise use K11RT4.

PRO/RT
For PRO/RT11 Version 5.1, please note that you MUST make two modifications to the XC handler. See K11PRT.MAC for more information.

The distribution has been subdivided roughly by operating system. The subdevice files KERCM*.DSK contain documentation and files common to all operating systems. The other subdevices are operating-system specific. The distribution also contains both save (binary, executable) images and .HEX (ASCII) versions of the save images. See the installation document for information on how to create a binary from the hex file. For RSX and RSTS, the HEX files are not contained in subdevices. Please note that the allocation of specific files to the operating system-specific subdevices was done without a great deal of research - If you can't find a file, try another subdevice!

KERCM1.DSK 10 Files, 388 Blocks (Common Files)
KERCM2.DSK 12 Files, 466 Blocks (Common Files)
KERCM3.DSK 14 Files, 195 Blocks (Common Files)
KERT1.DSK 13 Files, 412 Blocks (RT Files)
KERT2.DSK 3 Files, 380 Blocks (RT Files)
KERT3.DSK 2 Files, 453 Blocks (RT Files)
KERST1.DSK 14 Files, 437 Blocks (RSTS Files)
KERST2.DSK 6 Files, 396 Blocks (RSTS Files)
K11NRS.HEX 1 File, 726 Blocks (RSTS File)
KERSX1.DSK 13 Files, 349 Blocks (RSX Files)
KERSX2.DSK 2 Files, 400 Blocks (RSX Files)
K11RSX.HEX 1 File, 539 Blocks (RSX File)
K11POS.HEX 1 File, 363 Blocks (POS File)

SPRING, 1985 RT SIG TAPE DIRECTORY

MODULA - Modula-II Source file Conversion.

Don Stanger
Neurometrics
17 John Street
New York, NY 10038
(212) 267-1840

This program is used to convert a PIC macro subroutine into a .LNK file suitable for use in the RT11 Version of Modula-II distributed by the MODULA RESEARCH INSITUTE in UTAH.

MODULA.DSK 20 Files, 55 Blocks

NBS* - NBS Pascal Compiler.
FIS*
FPU*

Submitted by:
Paul Lustgraaf
32 Carver Hall
Iowa State University
Ames, IA 50011
(515) 294-1832

This is Version 1.6g of the NBS Pascal compiler for RT-11. It includes floating point support for either FIS or FPU. EIS is required in both versions. Included in this submission are:
NBS Pascal documentation.
NBS Pascal for both FIS and FPU machines.
Run-time library for FIS and FPU versions.
Pascal utility programs.
NBS Pascal compiler source for both versions.
Compiler objects for both version.

NBSDOC.DSK 14 Files, 415 Blocks
NBSFPU.DSK 6 Files, 198 Blocks
NBSFIS.DSK 6 Files, 202 Blocks
NBSSRC.DSK 9 Files, 459 Blocks
FISLIB.DSK 26 Files, 154 Blocks
FISOBJ.DSK 8 Files, 296 Blocks
FPULIB.DSK 29 Files, 150 Blocks
FPUOBJ.DSK 8 Files, 291 Blocks

SPRING, 1985 RT SIG TAPE DIRECTORY

PROM68 - Motorola 6800 Code Cross Assembler.

John T. Davies III
Thermo Electron Instruments, Inc.
524 Alpha Drive
Pittsburgh, PA 15238-2912
(412) 963-0903
(312) 972-6964

This program creates a .MAC program from a hex object code, as generated by the Motorola cross assembler for the M6800 series microprocessors. This program is written in a non-standard Version of RATFOR, a FORTRAN file of the output of this RATFOR preprocessor is also included.

PROM68.DSK 4 Files, 48 Blocks

RESEQ - FORTRAN Statement Number Resequencer.

R. W. Barnard
Sandia National Laboratories
Division 7523
Albuquerque, NM 87185
(505) 844-5115

RESEQ is a FORTRAN source file resequencer for statement numbers. This Version includes some bug fixes and enhancements. These include:
* Separating READ, IF, etc by <TAB>, as well as by spaces.
* Embedded <FF> characters are now handled correctly.
The file RESEQ.DOC gives more details; the comments in RESEQ.SAV also are helpful.

RESEQ.DSK 15 Files, 159 Blocks

RUNOFF - Bonner Labs Runoff Update.

Submitted by:
Jack J. Peterson
Horizon Data Systems
1901 Wildflower Terrace
Richmond VA 23233
(804) 740-9244

This distribution is intended to update Bonner Labs Runoff Version 6.2 as distributed on the Fall '84 RT-11 Symposium Tape. Included is a new Version of INIT.MAC which corrects a problem with reassigning control characters, such as &, _, =, etc. Included here are a non-overlaid and an overlaid pre-built Version of RUNOFF. RUNOFF.DOC is included, ready for printing on an 80-column printer.

RUNOFF.DSK 6 Files, 296 Blocks
RUNOFF.DOC 1 File, 649 Blocks

SDIR - Subdevice file utility programs.

D. N. Tanner
 Sandia National Laboratories
 Livermore, Ca 94550
 (415) 422-2314

FIND is a device and sub-device file search utility. The program will search devices and .DSK files for a file or files. If the file is in a sub-device, that name is also included. SDIR will will display the directory of any subdevice file. There is no need to MOUNT the file to a logical device.

SDIR.DSK 21 Files, 141 Blocks

SKUNK1 - "Skunk LUG" utilities.

Submitted by:
 Paul Lustgraaf
 Iowa State University
 32 Carver Hall
 Ames, IA 50011
 (515) 294-1832

These are some useful utility programs. The programs BUPRES, VIRHND, and WATCH have been modified since their last submission on RT SIG tapes.

- AVAIL Monitors available time on TSX+
- BEEP Beeps the terminal bell 10 times slowly.
- BUPRES Documentation for BUPRES.
- CRASH Documentation on fixing crashed disks.
- DSC Compares all the files on two disks.
- FIND2 Searches multiple devices for files.
- LINCNT Counts the number of lines in an Ascii file.
- LOAD Converts HEX files (from UNLOAD) to binary.
- STDTM Forces entry of the date and time.
- UNLOAD Converts binary files to HEX files.
- VIRHND Logical disk for disk virtual array handler.
- WATCH Logical disk for TSX+ detached job scheduler.

SKUNK1.DSK 22 Files, 283 Blocks

TERUTL - Terminal Handling Utilities.

Steve C. Cribbs
 Atomic Energy of Canada Limited
 Pinawa, Manitoba
 Canada R0E 1L0
 (204) 753-2311

TALKMT provides "Pass-all" communications between two RT-11 multi-terminal ports. Every character typed on the console is sent to the output side of the attached port. Every character received by the attached port is sent to the console. In "Snoopy" mode, all non-printing ASCII characters received by the attached port are converted into printable characters before being displayed on the console.

MTED was written to allow dynamic redefinition of the vectors and CSR's of multi-terminal ports. This prevents corruption of the monitor by indiscriminate manipulation of vectors and CSR's.

Another RT-11 User Command Linkage (UCL) program. Features include:
 * A memory for the last file specification edited by KED!
 * 10 defineable function-key-like commands available for VT100 and printing terminals that were built without them.
 * Small - this program requires little more memory than KMON and thus is usable in multi-task situations.
 * Written mainly in FORTRAN with extensive use of SYSLIB string handling routines.
 * A HELP command has been provided.
 * Display of command translations; "chain" to SY:DECUCL.SAV of unrecognized command strings; immediate mode

TERUTL.DSK 17 Files, 136 Blocks

SYSSIM - Operating System "Simulators".

Robert Walraven
 Multiware, Inc.
 139 G Street, Suite 161
 Davis, CA 95616
 (916) 756-3291

This submission is a "simulator" for various operating systems; it will provide a few laughs. In a more serious vein, the program is an example of good FORTRAN-77 programming style. It illustrates some techniques which you may not have known were possible with FORTRAN.

SYSSIM.DSK 5 Files, 52 Blocks

SPRING, 1985 RT SIG TAPE DIRECTORY

UCLPLS - User Command Language (UCL) Program.

William K. Walker
Monsanto Research Corp.
P. O. Box 32 OS-123
Miamisburg, OH 45342
(513) 865-3557

UCL+ is upward-compatible with the UCL distributed with RT-11, Version 5.1B and later. The Version submitted to this tape is V07.39, an update from all previous versions. UCL+ contains a number of extensions, including chaining to additional UCL's, "run-by-name", path definition, display of command expansions, etc. Symbols are defined by entering a "symbol definition string" in the format: symbol==definition. The DISPLAY command can be used to output ASCII strings to the console or printer (handy for sneaky escape sequences). A UC "pseudo-device" handler is provided as an option which allows UCL+ to "remember" the "input-spec" part of the last UCL+ command. This text can be retrieved, at the command level, by using the "^" character in place of the argument in a subsequent command.

This Version of UCL+ supports many new features of RT-11 and TSX+. It can be used with TSX+ as a "User Command Interpreter" (UCI). It minimizes disk access to improve efficiency; included on this distribution is a "memory-resident" UCL.

UCLPL1.DSK 19 Files, 408 Blocks
UCLPL2.DSK 19 Files, 408 Blocks

VRTARY - Virtual Array Access Routines.

N. A. Bourgeois, Jr.
NAB Software Services, Inc.
P. O. Box 20009
Albuquerque, NM 87154
(505) 298-2346

VRTARY contains routines for declaring, accessing, and eliminating a two dimension array in extended memory. The routines are written in MACRO-11 and follow the FORTRAN subroutine calling protocol. The size of the array is limited only by the amount of memory that is available. Three sample FORTRAN programs are included in the package, one test program and two application programs.

VRTARY.DSK 24 Files, 226 Blocks

SPRING, 1985 RT SIG TAPE DIRECTORY

UTIL1 - Clever Utilities.

Submitted by:
Wolfgang Leber
Max-Planck-Institut
Deutschordenstr 46
Frankfurt/Main 71
D-6000
West Germany

This submission has a number of utilities, including:
A UCL example (Funny UCL, in German)
A UCL by B. Kup, featuring parameter substitution and expansion of .COM files (TSX style)
A F4 INCLUDE statement pre-processor (generates temporary file with INCLUDE files inserted)
The famous GREP utility to do multi-file wildcard searches
A file to show how RT-11 CSI returns file specs & switches
A Help utility which first looks for .HLP file on DK:, HLP:. If not found, it will chain to SY:HELP.SAV
A complete DAYtime command, which also will save the current system date over bootstraps
An extension of a DEC dummy handler (never installs) to allow for the following system SET options: SET SY CACHE/NOCHACHE
SET SY DATE/NODATE
SET SY CLOCK=50/60
SET SY BUGS/NOBUGS
SET SY HELP
Help files with switches of MACRO, LINK, LIBR, FORTRA
Sample command files to demonstrate the TSX compatible UCL

UTIL1.DSK 55 Files, 207 Blocks

ASK
ASKF77 - Terminal I/O routines.

(Obtained from the Fall, 1984, European RT-11 SIG tape)

Ray Carpenter
c/o Shell Research Ltd
PO Box 1
Chester, CH1 3SH
UK

ASK is a set of FORTRAN terminal I/O routines. (ASKF77 are routines for use under FORTRAN-77 ONLY). They give you similiar (although more powerful) facility to the .ASK directives of IND.

ASK.DSK 36 Files, 251 Blocks
ASKF77.DSK 8 Files, 37 Blocks

SPRING, 1985 RT SIG TAPE DIRECTORY

COMM1 - Interprocessor communications.

(Obtained from the European Best of RT tape)

Eckart Meyer
Institut fuer Nachrichtentechnik
TU Braunschweig, FRG
Schleinitzstr. 23
D-3300 Braunschweig
West Germany

NET-I is a simple network for PDP11 computers running under RT11. It is ideal for sharing resources like backup floppy, mag-tape or line printer. It is easy to use and transparent to the user. NET-I currently uses standard serial lines (RS 232C) and is tested up to 9600 baud on a PDP11/03 and PDP11/23 link. NET-I works under RT11 V4.0 F/B.

Robert Walraven
University of California, Davis
Davis, CA

COM is a VAX-TSX+ communications package that runs at high baud rates. It is currently in use at 1200 baud, but has not been checked out at higher baud rates. The package consists of two parts: MO.MAC, a TSX+ general modem handler, and COM.FOR, a FORTRAN communications package that uses the MO handler.

COMM1.DSK 25 Files, 302 Blocks

DIAP05 - Slide Editor for VT100.

(Obtained from the Fall, 1984, European RT-11 SIG tape)

J. P. Lamargot
P. Pfeuty
I.R.S.I.D. 185 Avenue du President Roosevelt
78105 Saint-Germain en Laye
France
Tel 3 451-24-01

DIAP05 is a "KED-like" slide editor for use on a VT100 (or compatible) terminal. Text insertion is made just in the same manner as if you were using KED. Some keys on the auxiliary keypad have been redefined to provide new commands.

DIAP05.DSK 44 Files, 283 Blocks
DIAP01.DSK 72 Files, 448 Blocks

SPRING, 1985 RT SIG TAPE DIRECTORY

CPMRT - CP/M-RT11 ASCII File Translator.

(Obtained from the European Best of RT tape)

F. Kuster, dipl. El'Ing.
Eidg. Techn. Hochschule
Hybridrechenzentrum AIE
Gloriastr. 35, ETZ J96
CH-8092 Z u e r i c h
Tel. 01 - 256 5336

(The following is a translation of the .DOC file for this submission. Don't tell me how bad my German is... RWB).

The program CP/M-RT11 reads and writes CP/M 8"-SD-floppies onto RX01 or RX02 [diskettes]. The program CP/M-RT was written by Mr. Stoehrel and was modified by Messrs. Kuster and Sempert. A version for VAX/VMS is in an early phase of implementation; the corrections to the RT-version are however not completed. Distribution and use of this program is without guarantee and liability. Please send any bug reports to the above address.

(Translation of directory comments)
CPMRT .COM 1 08-Nov-84 Translated CPMRT
CPMRT .FOR 12 09-Nov-84 Main program
SSEC .MAC 7 08-Nov-84 reads and writes 'single sectors'
CPMRT .SAV 67 20-Nov-84 linked with 'NHD'
for the CP/M - VAX Version:
(only under development, not complete, not the actual Version)

CPMRT.DSK 17 Files, 154 Blocks

DTX - Read and write DOS magnetic tape.

(Obtained from the Fall, 1984, European RT-11 SIG tape)

Ray Carpenter
c/o Shell Research Ltd
PO Box 1
Chester, CH1 3SH
UK

DTX is a utility to allow you to read and write DOS formatted mag-tapes under RT-11. This version supersedes an earlier version issued about two years ago.

DTX.DSK 4 Files, 126 Blocks

SPRING, 1985 RT SIG TAPE DIRECTORY

CROSS1 - 8080 Cross Assembler.

(Obtained from the European Best of RT tape)

Eckart Meyer
Institut fuer Nachrichtentechnik
TU Braunschweig, FRG
Schleinitzstr. 23
D-3300 Braunschweig
West Germany

This is a computer program written in PDP11 MACRO-11 to assemble Intel 8080 assembly language under RT11. It uses an identical algorithm to the Intel assembler version 4.1 and can produce output files identical in format and content to the "LIST" and "HEX" output produced by the Intel assembler. It completes assemblies in about one fifth of the time, and has potentially much more space available for symbol tables than the Intel assembler.

The cross assemblers CAXX uses the full functionality of MACRO 11 to provide things like conditional assembling, program sections or the macro feature for a microprocessor assembly language. Crossassemblers currently available are:

- CA68 - for 6800
- CAB5 - for 8080/8085

CROSS1.DSK 24 Files, 299 Blocks

MONUTL - Monitor Utilities.

(Obtained from the Fall, 1984, European RT-11 SIG tape)

Ian Hammond
Hammond Software
Goettingen
West Germany

Handy utilities for checking monitor offsets, device tables, and logical assignments.

MONUTL.DSK 4 Files, 24 Blocks

SPRING, 1985 RT SIG TAPE DIRECTORY

HANDLR - EIS emulator and Handshake handlers.

(Obtained from the European Best of RT tape)

P. C. Waggett

EI.SYS was written to emulate the EIS instructions MUL, DIV, SOB, ASH, ASHC, SXT and XOR on a BIS machine. EIS instructions will be emulated, transparently to the user, by software. The driver emulates all sensible (and probably all non-sensible) EIS assembler instructions using any combination of registers, including PC and SP.

Gary Preckshot
University of Arizona
Tucson, AZ

HS is a handshaking serial line handler which allows convenient file transfer between two machines running under DEC RT-11 Version 3B or later, and whose protocol is so simple that a complementary serial I/O handler may be written easily for other operating systems. Synchronization is achieved by transmitting an endless series of ENQ characters until the receiver responds with a single ACK.

HANDLR.DSK 7 Files, 154 Blocks

UTIL2 - Miscellaneous Utilities.

(Obtained from the Fall, 1984, European RT-11 SIG tape)

Ronald Beetz
AKZO PHARMA
Oss, Holland

BACDAT creates a file on the DK: disk with the current date and time. This program should be run just before a backup of the disk to indicate when the last backup was made.

UPDCOM.TEC is a teco macro program to change common areas of a FORTRAN-IV program in all modules of that program. FORTRAN-IV lacks the INCLUDE statement which is quite annoying for programs with many modules. UPDCOM can help remedy this omission.

DAYTIM is program that asks for date and time. The program checks to see if a legal date has been entered; if so, the current date and time are displayed; if not then date and time are asked.

TEK125 is a program to display TEKTRONIX PLOT-10 information on a VT125 terminal. TEK125 is maybe the easiest solution for your conversion problems.

UTIL2.DSK 13 Files, 91 Blocks

SPRING, 1985 RT SIG TAPE DIRECTORY

CROS2* - AIM65 Cross Assembler.

(Obtained from the European Best of RT tape)

F. Kuster, dipl. El'Ing.
Eidg. Techn. Hochschule
Hybridrechenzentrum AIE
Gloriastr. 35, ETZ J96
CH-8092 Z u e r i c h
Tel. 01 - 256 5336

This is a cross-assembler for the AIM65 microcomputer. It runs under RT11 SJ, Version 4 or 5, using OMSI Pascal, Version 1.2.

CROSS2.DSK 6 Files, 221 Blocks
CROS2A.DSK 28 Files, 290 Blocks

INCLUD - FORTRAN-IV Include file capability.

(Obtained from the European Best of RT tape)

B. Kup
TH Darmstadt
D-6100 Darmstadt
West Germany

INCL is a FORTRAN/RT pre-processor to simulate an INCLUDE statement. It will handle multiple include statements, and accepts file specs with various syntaxes, such as:
INCLUDE 'filnam.ext'
include filnam.ext

INCLUD.DSK 4 Files, 34 Blocks

PIPB - PDP11-PDP8 File Transfer Program.

(Obtained from the European Best of RT tape)

John Yardley
National Physical Laboratory
Teddington, Middx, England

PIPB is a file transfer and file maintenance utility program for both OS/8 and RT/11 (ASCII) files. It runs under RT/11 and enables you to transfer files in the same format or from one format to another. You can obtain directories of discs, rename or delete files in either format. It is possible to copy files in the following formats:
RT/11 TO RT/11.
OS/8 TO OS/8.
RT/11 TO OS/8.
OS/8 TO RT/11.

PIPB.DSK 61 Files, 486 Blocks

SPRING, 1985 RT SIG TAPE DIRECTORY

RMQ - Interprocessor communications, and other stuff.

(Obtained from the European Best of RT tape)

P. Wegmann
F. Kuster
G. Maier
Eidg. Techn. Hochschule
Hybridrechenzentrum AIE
Gloriastrasse 35, ETZ-J
CH-8092 Z u e r i c h

A program which operates much like IND, in that it allows comments and operator decisions to be interspersed with KMON commands.
(Translation of the file README.TXT)

RMQ is a program for the printing of a commentary in a running RT-11 command file. The execution of the command file can be stopped and, depending on the application, continued or aborted from TT:.. You may also flush part of the command file. Documentation is in the program. The command files SQDK.COM (Squeeze DK:) and COPRT4.COM (Selective copying of a 'master disk') are examples of the use of RMQ.

DAZEIT (DATIME) will, at the end of a startup file, force the date (and the time) to be entered. The date will be checked for correctness (including leapyear). The month entry can be numerical, as well as in the 3-letter format (English or German). A missing year numeral will be supplied.
Examples: 1.Jan 82 5/MAR 25-11

The 'Default' year can be set in the source code with DAZUPD.COM. Further documentation in the program. STARTS.COM shows the use of DAZEIT as well as several further ideas together with DIR.COM, VT100.COM, LA36.COM, INSDXY.COM.

SELIN and DRVERB are two file-transfer programs between two RT-11 computers. SELIN supports a (local) serial connection, DRVERB uses a 16-bit connection with DR-11C or DRU-11 interfaces. All the equipment in both computers must be addressable before the service will commence. The commands can be read from the consoles or from a command file. Common documentation in the file SELIN.DOC.

RMQ.DSK 15 Files, 170 Blocks

SPRING, 1985 RT SIG TAPE DIRECTORY

SUP - Subsystem Utility Program.

(Obtained from the European Best of RT tape)

Jens-Peter Jensen
Institut f. Nachrichtentechnik
TU Braunschweig
Schleintzstr 23
3300 Braunschweig
West Germany
Tel: 391-2489

SUP is a program to create and maintain subsystem files. You can use SUP to copy files on physical devices to files in subsystems and vice versa. You can create complete, bootable subsystems by using SUP exclusively. The SUP command syntax resembles normal RT11 Command Language. SUP commands include:

HELP
COPY
INITIALIZE
BOOT
DIRECTORY
RENAME
KILL
EXIT

SUP.DSK 6 Files, 271 Blocks

COOKIE - Writes a clever saying.

(Obtained from the Spring, 1983 RT SIG Tape)

Joel Berez
Infocom
64 Jacqueline Rd.
Waltham, MA 02154
(617) 492-1031

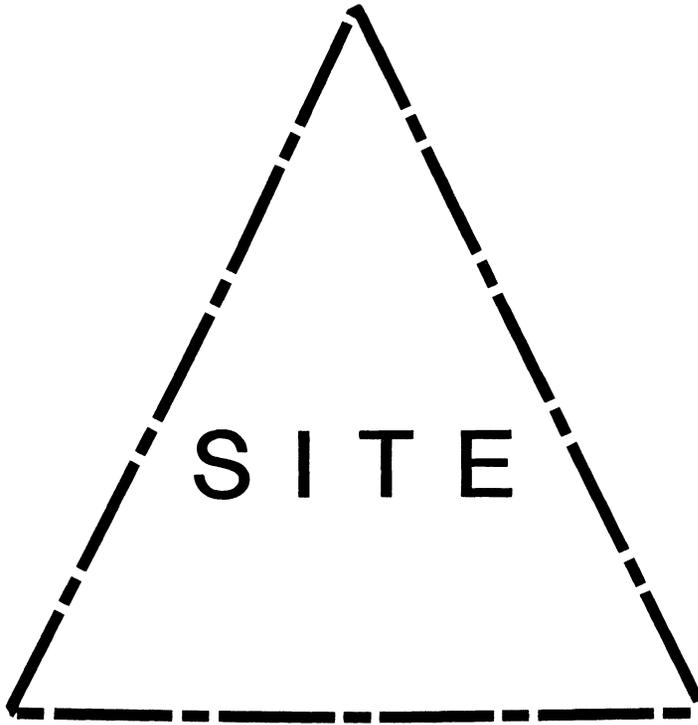
Bye prints a saying or definition from BYE.LNS when run. If that file is edited, the index must be rebuilt with the ByEBLD program.

COOKIE.DSK 4 Files, 134 Blocks





DECUS



SITE

Chairman

David Hunt
Lawrence Livermore National Lab
Livermore, CA

Librarian

Larry W. Hicks
Relational Database Services
Kernersville, NC

Symposium Coordinator

Michael Weaver
OJCS/JAD/TSD
Washington, DC

Hardware Management

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

Communications Committee Representative

Terry C. Shannon
DEC Professional Magazine
Springhouse, PA

Staff Management

Adam Zavitski
Tulsa, OK

Newsletter Editor

Gregory N. Brooks
Washington University Beh. Res. Lab
St Louis, MO

Members-At-Large

George Hilton
CompuCard International
Stanford, CT

Debbie Boole
Texas Instruments
Dallas, TX

LUG Coordinator

Doug Bickford
UVM Academic Computing Center
Burlington, VT

DEC Counterparts

Rogers Bent
Digital Equipment Corporation
Stow, MA

Session Note Editor

Gary Bremer
Emerson Electric Co.
St. Louis, MO

Lil Holloway
Digital Equipment Corporation
Bedford, MA

HMS SIG Liaison

Jim Corrigan
TI Research
Richardson, TX

Susan Porada
Digital Equipment Corporation
Marlboro, MA

Assistant Librarian

Timothy Frazer
Specialized Bicycle Components
Morgan Hill, CA

Gary Siftar
Digital Equipment Corporation
Tulsa, OK

SITE MANAGEMENT AND TRAINING SIG NEWSLETTER INDEX

Letter From the Editor.....page 2

VAX SPM: An Advanced Performance Tool for VMS Systems..page 3

The Site SIG System Manager's Manual.....page 14

WANTED.....page 14

The Importance of Power.....page 15

Wiring the Campus.....page 23

FROM THE EDITOR

Welcome to the first issue of the combined DECUS SIG Newsletters.

Such a combined format is a departure from the way SIG Newsletters have been published in the past. Months of debating issues and considering the advantages and the consequences have been taking place behind the scenes within the DECUS Leadership.

We hope this format will meet the needs of the majority of DECUS members and, be cost effective.

We invite your comments and questions on the format and the content of this publication. Our desire is to serve you, the readers. By your comments we are able to judge whether we are meeting that objective.

We hope you will enjoy the five articles we present in this issue and urge you to fill out the survey form submitted by Jim Corrigan.

Also, I would like to thank all of you who participated in our last survey.

As always, your comments and questions are welcome. Please send them to my address, listed below.

Gregory N. Brooks
Washington University
Behavior Research Labs
1420 Grattan Street
St. Louis, MO 63104
(314) 241-7600, ext. 257

VAX SPM: AN ADVANCED PERFORMANCE TOOL FOR VMS SYSTEMS

By Terry C. Shannon
Professional Press
Springhouse, PA

Computer-assisted management is an aspect of data processing that's gaining acceptance and popularity in a variety of management-oriented disciplines, including MIS, material requirement planning, bill of material processing, and workload projection. With the release of Version 2 of the VAX Software Performance Monitor, DEC brings computerized management to the computer itself.

The best way to explain how this system management tool works is to show its use in the system optimization process. Therefore, this paper presents a brief overview of VAX/VMS system tuning and capacity planning issues, highlighting the role SPM plays in both applications.

SOFTWARE MONITORS DEFINED

DEC defines a software monitor as "software that collects and reports on system resource usage." While this statement is true, it's neither a description nor a complete definition of VAX SPM. The ACCOUNTING, DISKQUOTA, and MONITOR utilities supplied with every VMS distribution kit share the foregoing definition to a certain extent, but the capability and functionality of these default VMS utilities is eclipsed by the features provided in SPM.

The nature of SPM is such that a novice system manager can use its tools effectively, while at the same time it supplies features that are equally useful to the VMS "guru." The software provides information and reports that, when properly analyzed, can be used to accurately measure the performance of a VAX/VMS system. Using data collected by SPM, system managers can identify performance bottlenecks, quantify current and projected system workloads, and base the need for hardware upgrades on facts, not conjecture.

SPM COMPONENTS

The components provided in the SPM toolkit fall into three broad areas: system tuning, system capacity planning, and application program tuning. SPM supplies each functional area with components and utilities that collect and report key statistical data germane to that area. These tools are divided into the seven following categories.

- o Data archiving - storing capacity planning data in history files.
- o Data collection - accumulating tuning and performing statistics.
- o Data conversion - converting history files created by SPM Version 1 to SPM Version 2.
- o Data definition - specifying the individual data elements you want SPM to monitor.
- o Data display - on a ReGIS video terminal or a graphics printer.
- o Data reporting - generating reports suitable for system managers and programmers as well as supervisory personnel.

- o The system event trace facility - a tool for writing your own monitor to trace any VMS event you deem to be of particular interest.

Armed with the detailed information available through the use of these tools, the system manager is in a good position to analyze, evaluate, and make recommendations in the areas of system tuning, capacity planning, and application program tuning.

SYSTEM TUNING IN BRIEF

System tuning is a process undertaken to enhance the overall short-term performance of a VAX/VMS system. Tuning is normally conducted system-wide, the goal being to optimize the performance of the entire system rather than specific applications. The best indicator of proper tuning is an increase in productivity measured on a system-wide basis.

Establishing VAX/VMS productivity gains can be a job fraught with as much difficulty as calculating the annual GNP of the United States. There is no quick and dirty algorithm or benchmark available to accomplish the task. Unfortunately, you can't invoke a utility that will tell you at what percent of maximum productivity your VAX is performing. So, how do you accurately assess, and then increase, the productivity of a VAX/VMS system?

THE MEDICAL APPROACH

Basically, you approach the issue as if it were a disease. You must define the problem at hand, taking into account your system, its workload, and the specific indicators of poor system performance. Having established the system's present condition and "symptoms," you are ready enter the diagnostic phase. This involves determining and recording the system state, gathering relevant performance data, and analyzing the results. Next, you must prescribe a reasonable course of therapy by establishing goals for improvement and taking actions that will help you meet these goals. The final tuning step is to re-examine the system to determine how effective your therapy has been.

Two techniques may be used to carry out the tuning process. The traditional method involves recording and analyzing statistics produced by the VAX/VMS ACCOUNTING and MONITOR utilities. Even though the MONITOR utility has been enhanced significantly in VMS Version 4, tuning remains a frustrating, time consuming task that presupposes an extensive knowledge of the intricacies of VMS and the nuances of your system and its workload.

VAX SPM provides a viable alternative to the traditional method of tuning VMS, and greatly facilitates the process in the bargain. The SPM components that provide data essential to system tuning include utilities that collect and report system-wide tuning and program counter (PC) data, a video graphics utility used to display this information, a disk usage utility, a chargeback utility, and a utility for tracing events in system space.

SYSTEM TUNING UTILITIES

Five SPM utilities are of value in the tuning process. These programs collect, report, and display various elements of system information on several levels. To obtain a summary of an entire system, you collect and report information through the use of the SPM COLLECT=TUNE and REPORT=LOG_FILE utilities. The REPORT=DISK_SPACE utility provides a detailed view of data storage on

individual or multiple disks. REPORT=SYSTEM_PC uses the data gathered by COLLECT=SYSTEM_PC to provide a comprehensive look at CPU utilization. The DISPLAY utility enables you to produce a real-time display of system activity on a graphics terminal. Finally, there's an event trace facility that lets you follow the activity of the specific system events that you feel may be affecting system performance.

GATHERING PRELIMINARY INFORMATION

Prior to collecting system performance data, you should identify the data elements that are of interest to you. This is important because inadequate data makes problem diagnosis difficult if not impossible, and collecting extraneous data wastes disk space and can affect system performance adversely. It's also a good idea to collect data on a regular basis even when you are not actively tuning your system. Periodic data collection allows you to establish a system performance baseline, making it possible for you to determine what constitutes "normal" and "abnormal" system behavior.

COLLECTION PARAMETERS

You should establish data collection parameters that specify a reasonable sampling interval, provide enough information to diagnose most problems, and create a log file that you can archive for future use. To this end, it is suggested that you collect all classes of data using a five minute sample interval, and close out the log file once per day. Initializing a new file on a daily basis and choosing a 300 second sample interval will save considerable amounts of disk space that would otherwise be consumed by a continually expanding log file. Like image level accounting, SPM can develop an unhealthy appetite for mass quantities of disk space.

To curb the appetite of SPM, you should omit those data classes which do not apply to your system. If system activity changes significantly at less than five minute intervals, you should specify a shorter collection interval. Conversely, if your system activity is relatively static for longer periods of time, there's no sense in collecting data every five minutes when a 15 minute collection interval would suffice. The command line below illustrates the collection of system-wide tuning data.

```
$ PERFORMANCE COLLECT=TUNE log_file_name -  
  _/INTERVAL=300/CLASS=(ALL,PROCESS=IMAGE) -  
  _/DISK=disk_name(s)/DEVICE=device_name(s) <RETURN>
```

This command creates and places into execution a detached process named SPM_TUNE. The process collects default data including CPU states, memory utilization, paging and I/O rates, file cache information, and swapper counts. Optionally, you can direct SPM to collect information including device I/O rates, scheduler and resource wait states, disk, controller and process statistics, and file system primitive counts. System configuration data is collected by default, and additionally you may collect information on hardware configuration, installed images, global sections and SYSGEN parameters.

REDUCING THE DATA

A raw SPM log file is similar to the ACCOUNTNG.DAT file generated by the VMS ACCOUNTING utility. It contains a vast amount of detailed information, much of

which may have no bearing on your specific tuning issue. Your first course of action should be to obtain a brief report of system usage that will enable you to identify and isolate potential problem areas. This is done by obtaining an abbreviated "notabular" report of system utilization from the data in the log file. This graphic report details system usage on a percentage basis and breaks the usage down into CPU, I/O, and CPU - I/O components. The command you use to do this is shown below.

```
$ PERFORMANCE REPORT=LOG_FILE/NOTABULAR log_file_name <RETURN>
```

The timeframes that reflect complete system utilization are likely problem areas or "hot spots." Having identified these areas, run the REPORT utility again, specifying a tabular report that summarizes all classes of information. Restrict the report to the desired timeframe by supplying the appropriate values for the /BEFORE and /SINCE parameters.

```
$ PERFORMANCE REPORT=LOG_FILE/NOGRAPH/TABULAR=(INTERVAL,FINAL) -  
  _/CLASS=(ALL,PROCESS=IMAGE)/SINCE=16-AUG-85:xxxxx/BEFORE=17-AUG-85:xxxxx -  
  _log_file_name <RETURN>
```

This command line produces a system utilization report for the time period defined by the /BEFORE and /SINCE parameters. The report includes an extensive series of system-metrics tables that break system usage down into a format that facilitates tuning. A portion of a final statistical report is reproduced at the conclusion of this paper. The information contained in this example report is the basis for the following discussion on VAX/VMS resource limitations.

DETERMINE THE LIMITED RESOURCE

VAX/VMS system degradation is almost invariably a function of a limited system resource. The culprit may be memory limitation, I/O limitation, CPU limitation, or a combination of these conditions. It's important to remember that these factors impact upon and interact with each other. For example, limited physical memory causes excessive swapping and paging. This results in a high I/O rate, which in turn places an unreasonable burden on the CPU. As you can see, a memory limitation will manifest itself as a function of all three system resources. Thus, it is vital to determine not only which resources are limiting factors, but to examine the resources in a logical sequence. Because it is a precursor to other system limitations, your tuning investigation should begin with an analysis of your physical memory resources.

MEMORY LIMITATION

The most frequent cause of poor system performance is inadequate memory, hence the often-repeated admonishment, "Buy more memory." Digital is right more often than not about memory limitations, but it can be foolhardy to act on this advice without first giving it due consideration. Before you rush out and buy a few more megs of VAX memory, you should make sure that (a) your system actually needs more memory, and (b) that memory will result in a significant increase in system performance.

Memory limitation has three primary symptoms; no free memory, a high paging rate, and an excessive swapping rate. Analysis of the system-wide tuning data compiled by SPM can help determine the presence and severity of each of these

symptoms. The tabular form in which these system-metrics are displayed makes data analysis straightforward and relatively easy.

No Free Memory

The AVE Process-Memory Counts and Memory Utilization portions of the system-metrics report provide a rapid answer to the free memory question. If free pages are low and total memory utilization approaches 100 percent, your system is memory-bound.

High Paging Rate

The Paging Rates (per second) section of the report lets you zero in on excessive paging. Of course, you need to know what constitutes a "normal" paging rate for your system, but if system faults make up a large proportion of the overall paging rate, chances are you have a problem.

Excessive Swapping Rate

The AVE Mem/CPU Queues and Swapper Counts entries pinpoint excessive swapping. If processes are waiting in memory queues or a significant amount of CPU time is devoted to swapping, a detailed investigation of working set and balance set sizes is advisable.

I/O LIMITATION

Now that you've established the culpability or innocence of the memory boards in your VAX, you can begin the second phase of the tuning investigation by determining the presence and evaluating the impact of I/O constraints on system performance. The symptoms of I/O limitation are excessive buffered I/O rates or direct I/O rates, or a combination of both. To determine which symptom or symptoms your system suffers, consult the "I/O Rates (per second)" portion of the report.

CPU LIMITATION

The final element of the investigative phase of system tuning involves determining the effect of CPU limitations on system performance. CPU limitations are indicated by processes in the CPU queue, no idle CPU time, and an inordinate amount of system-related CPU usage. The presence or absence of these conditions can be determined from the system-metrics report in the following manner:

Processes In The CPU Queue

The "AVE Mem/CPU Queues" section of the report lists the number of processes waiting for service in the CPU queue. A large number of processes in the queue is indicative of CPU limitations.

No Idle Time

The "CPU Idle" portion of the report lists idle CPU time and the percentage of time that the CPU is waiting for swapping and/or paging activity. For all

SIT-7

intents and purposes, this latter figure represents "lost" CPU time.

System CPU Usage

The "CPU Busy" portion of the report breaks down CPU usage by mode. If a high percentage of CPU time is devoted to interrupt stack and kernel/executive mode processing, and a low proportion of time is being spent in the user mode, the system itself is consuming too much CPU time.

ISOLATE THE CAUSES OF LIMITATIONS

Having determined the resource or resources that act as limiting factors in your system, you need to isolate the specific contributors to poor performance so you can deal with them individually. This is accomplished by referring to the system-metrics report and several other reports available in SPM. A brief look at some of the parameters and system events that may be impairing the efficiency of your system is presented in the following text.

Memory Limitations

If memory appears to be a problem, you should take a close look at working set sizes and adjustment parameters. If these are not set properly, excessive page faulting will result. Examine the page fault rate and working set size of each process on the system. A high page fault rate may indicate that the working set size for a given process is not large enough.

You should also look for excessive image activations. A process characterized by a high page fault rate and minimal CPU time is likely to be doing an inordinate amount of image activation.

Problems can also arise from inappropriate memory loan parameters, indicated by a high page fault rate coupled with an abundance of free memory. And improper swapper trimming, an inadequate balance set count, or a situation in which a few processes are consuming the bulk of available memory all manifest the need for adjusting swapping parameters. For instance, a privileged user may have disabled swapping for his or her process. Alternatively, the SYSGEN BORROWLIM, GROWLIM, or SWAPOUTPGCNT parameters may be set to unrealistically high values.

I/O Limitations

I/O limitations are directly related to system peripheral devices. A disk with a high rate of I/Os per second is likely to be a contributor to direct I/O problems. It's also important to determine whether excessive I/O activity is attributable to paging or swapping.

Poor file system caching also has a negative impact on I/O effectiveness. File I/O rates per second are useful for determining the likelihood of another I/O bottleneck, disk fragmentation. A high rate of window turns and a high split I/O rate are both indicators of a fragmented disk.

CPU Limitations

Determine the presence of processes in the CPU queue. If this is a significant problem, look to see if one or several processes are pre-empting the remainder of the processes in the system. Determine the percentage of the time that the

SIT-8

CPU is idle because it is waiting for paging and swapping. Finally, isolate and quantify system CPU usage with the SPM COLLECT=SYSTEM_PC and REPORT=SYSTEM_PC utilities. These utilities enable you to see where and in what mode the system is using the CPU.

TREAT THE PROBLEM

You should take the necessary steps to alleviate poor performance manifested by any of the factors discussed in the preceding text. Some of these steps have been mentioned in conjunction with explanations of the causes of poor system performance. However, this article is not intended to be an in-depth, comprehensive guide to system tuning. For detailed information of this nature, refer to the VAX/VMS Guide To System Performance Management.

FOLLOW-UP

Having treated the poor performance disease with the appropriate therapeutic measures, you must re-examine your system to evaluate the impact of your tuning treatment. This is done with the SPM video display and system event trace utilities. These utilities permit you to display histograms and Kiviat graphs on a ReGIS-compatible monitor, and to obtain detailed information on specific system events that may have contributed to the system performance problem.

Displaying System Performance Data

You can display load balance, system, and working set data on a monitor, or print the display on a graphics dot matrix printer. If your monitor and/or printer support color, multicolor displays are generated.

The LOAD_BALANCE display produces a Kiviat graph of key CPU and disk performance statistics. The eight system metric values included in the graph are plotted in a circular manner, so the completed image resembles a compass rose. Each value originates at the center of the circle, which represents a value of zero percent. The eight equidistant points on the circle's diameter indicate a one hundred percent value for each metric. The four metrics plotted on vertical and horizontal axes represent "good" system conditions, while those that are offset 45 degrees from the vertical or horizontal axes are indications of unsatisfactory system performance.

Ideally, a Kiviat display will resemble a four-pointed star, indicating an equal balance between CPU and I/O operations. The more the image deviates from this ideal, the less the system's load is balanced. For example, if the top two quadrants of the circle reflect the bulk of system activity, the system is devoting much more time to servicing the CPU than to I/O operations. The format of the Kiviat chart makes this fact immediately evident, allowing you to take corrective action based on a real-time display rather than the subsequent reduction and analysis of log files and reports.

A five-section SYSTEM graph, plotted as a histogram, may be displayed to monitor real-time system performance. System-metrics including paging and swapping statistics and the average working set size are displayed across the top of the graph. From left to right across the histogram portion of the display are bar charts that monitor process priorities, CPU modes, paged memory statistics, and disk utilization information for each disk you have selected. If you have a VT240 or ReGIS-compatible color monitor, the graph is displayed in an easy-to-read, multicolor format. On a monochrome monitor, different colors are represented by shading or crosshatching.

SIT-9

Finally, you can direct the video display utility to produce a four-part working set graph. Similar to the SYSTEM graph, the WORKING graph includes a system-metric display. Beneath the system-metric data are histograms that summarize working set statistics, priorities, and CPU modes. Properly interpreted, the data displayed by the WORKING graph can save you much time and aggravation as you determine the effect of tuning measures on your system. For example, the graph will show you, in real-time, whether your system actually is saturated or the first two or three users have inordinately large working set sizes.

CAPACITY PLANNING ISSUES

Once you have optimized short-term system performance through proper tuning, the resolution of long-term performance issues through system capacity planning and workload projection is in order. This is another task you don't want to approach lightly. Planning hardware upgrades in a manner that strikes a cost effective balance between anticipated workload growth and performance constraints can be haphazard if you don't have the proper tools to assist you in the process. To undertake capacity planning effectively, you need to know which resources are in demand on your system, the users that consume the bulk of these resources, and projected workload growth.

No software monitor can take the place of a realistic business plan when extrapolating future workload growth. It is the business plan that defines how you expect your system workload to change. But, SPM does a thorough job of collecting the resource utilization data you need to formulate an upgrade plan based on factual historical information instead of conjecture.

Gathering Relevant Information

To determine resource usage you must first determine what data to collect, and the frequency with which to collect it. DEC recommends that you collect information continuously, using a 10 to 12 minute sampling interval. To direct SPM to do this, you enter a command similar to the following:

```
$ PERFORMANCE COLLECT=CAPACITY log_file_name /INTERVAL=600/ -  
_CLASS=(PROCESS:IMAGE, DISK)/SYSTEM_COMMUNICATION/DISK=device_name(s) <RETURN>
```

This command line tells SPM to collect system statistics useful for capacity planning at a 600 second or ten minute interval and to store this information in the specified log file. You may specify as many as 11 optional collection classes on the command line, affording a high degree of flexibility. The SPM component that does the actual data collection runs as a detached process, leaving your terminal free to receive other interactive commands.

Archiving The Information

It's advisable to archive capacity data from the log file into a history file on a daily basis. This permits more efficient data storage in a centralized database and provides you with the ability to add, replace or delete history data as desired. A command line like the example below is used to archive log file data into a history file.

SIT-10

\$ PERFORMANCE ARCHIVE=LOG_FILE/history_file_name log_file_name <RETURN>

The history file provides SPM with the information it needs to generate reports on which resources are being used, when they are being used, and which users consume what proportion of the available resources.

WHICH RESOURCES ARE BEING USED?

The last phase of resource utilization monitoring with SPM involves the generation of a variety of reports that reflect resource use over a period of time. Because these reports likely will be used by management personnel with little if any background in VAX/VMS as well as by technical staff, DEC has incorporated a high degree of flexibility into the reporting process. You can select graphic or tabular output, generating as many as 17 optional graphs and specifying optional sections for tabular reports. Reporting intervals and durations are user-definable, and "typical" daily graphs can be produced by excluding holidays and non prime time periods.

WHO IS USING THE RESOURCES?

You now have an extensive graphic representation of your system and its workload, so you now know which resources are in demand. The next step in the capacity planning process is to determine which users are the primary consumers of your system resources. SPM helps you do this by enabling you to characterize and then analyze your system workload.

Workload characterization is done by dividing your total workload into classes. For example, you can choose to categorize workload by account name, image, process name, or UIC identifier. This categorization yields the "who" element of the resource utilization equation.

Now you need to find out what percentage of available resources is being consumed by each group you have defined. This process, referred to as workload analysis, may be accomplished with VAX SPM, the VMS ACCOUNTING utility, or a combination of both software modules. Your choice in this regard depends on what information is important to you. Both SPM and ACCOUNTING provide CPU and I/O data, but ACCOUNTING can summarize process information while SPM lacks this capability. On the other hand, SPM provides memory data not available through the VMS ACCOUNTING utility.

To produce a tabular report that includes process metrics for each group, select a log file that is representative of your typical system workload and enter a command like the following.

\$ PERFORMANCE REPORT=LOG_FILE log_file_name/NOGRAPH/TABULAR=INTERVAL-
_/_CLASS=PROCESS:IMAGE <RETURN>

A few lines of code will yield a summation program that reads the log file produced by the preceding command line, calculates percentages, and generates a listing that will tell you what slice of the resource pie each group is consuming. Combined with a good business plan and judicious extrapolation based on the projected growth of each group of users, this information should enable you to develop and implement a realistic upgrade plan for your installation.

A FRINGE BENEFIT

Although VAX SPM exists to provide detailed information useful for system and application program tuning and capacity planning, it does provide a resource chargeback facility. By using the REPORT=CHARGE utility and the standard VAX/VMS ACCOUNTING.DAT file, you can equate dollar values to a number of system resources and then charge individual system users for the resources they consume. Billable resources include hard and soft I/Os, CPU and elapsed time, volume mounts, image activations, page faults, and pages of printed output. Utilization can be reported on six detail levels, and account totals and grand totals are calculated by default.

The only significant feature the CHARGE utility lacks is the ability to collect disk usage information and factor it in to the chargeback algorithm. To obtain this information, you'll have to use the VMS DISKQUOTA utility or the DISKUSE program available from the DECUS Program Library.

MEDIA AND DOCUMENTATION

VAX SPM is supplied as a binary distribution kit on magtape, floppies, or TU58 tape cartridges in VMS BACKUP format. A brief set of release notes and the VAX SPM Reference Manual complete the SPM package. The manual is not light reading, and you would be well advised to read the VAX/VMS Guide To System Performance Management before you attempt to tackle SPM. You also may want to obtain the cassette recordings of the VAX SPM sessions presented at the Fall 1984 or Spring 1985 DECUS symposia as well as the appropriate session notes.

INSTALLING SPM

SPM installation is a straightforward procedure done through VMSINSTALL. The entire installation and verification procedure should take you less than an hour, and is done as follows: Log into [SYSMGR] and run SYSGEN, ensuring that the VIRTUALPAGECNT parameter is set to a value greater than 9000. Then, run AUTHORIZE, modifying the UAF as necessary to obtain SETPRV, a PGFLQUOTA greater than 9000, and a BYTLM in excess of 8600. Nonpaged dynamic memory requires a minimum value established by the algorithm (MAXPROCESSCNT * 4) + 4000. You'll also need at least 3500 free blocks on the system disk, and access to that disk's BITMAP.SYS and INDEXF.SYS files.

After you restore the SPM saveset from the distribution media, reboot the system, invoke VMSINSTALL to install the product, responding "YES" when you are asked if you want to run the Installation Verification Procedure. Finally, refer to the installation notes and edit your site-specific startup and shutdown files to load the SPMTIMER driver and install or delete SPMshr.EXE, the image data collection procedure, as a shareable image.

Once installation is complete and you start SPM up, you'll note that your system features an added device called SWAO. This is a pseudo device used in conjunction with the SPM Timer Driver to implement timing functions, and as such it should have no discernable effect on your system. In the unlikely event that your system already has a device named SWAO, you easily can rename the SPM pseudo device to another unique name of your choosing. You are now ready to use SPM to collect, report, and display all of your system's vital statistics. When combined with a general knowledge of VAX/VMS tuning and capacity planning techniques, the data supplied by SPM should allow you to keep your system running at peak efficiency.

Editor's note:

This article originally appeared in a slightly different format in the April 1985 issue of The VAX/RSTS Professional Magazine and is reprinted with the permission of the author and Professional Press.

THE SITE SIG SYSTEM MANAGER'S MANUAL

"... finally, one place to look for information I need to do my job."

"Brings together the best ideas and hints from experienced DEC system managers"

Have you been looking for just such a manual to help you manage the hardware, environment, and personnel at your site? Surely, you say, someone has solved this problem before?

Well, we in SITE SIG believe that also. While no single site/system manager may have solved and documented the solution to all the problems, we think we can assemble from many sources within the SIG a very useful collection of the information you need.

To do this, we need two things and we want to ask you to help. First, we need to confirm and supplement our list of topics which you think should be covered in a SITE SIG System Manager's Manual. Second, we would like to know if you think you may be able to contribute some expertise (or documents/articles already prepared at your site) on some of these topics.

If you would like to help, please write to:

Douglas Bickford
 University of Vermont
 Academic Computing Center
 Cook Science Building
 Burlington, VT 05405-0125

We would like to hear from you. Even a quick postcard of your "hot topic" would help.

W A N T E D

Authors to write articles on topics of interest to members of the Site Management and Training SIG for publication in the Site SIG Newsletter.

All those interested, please contact:

Gregory N. Brooks
 Washington University
 Behavior Research Labs.
 1420 Grattan St.
 St. Louis, MO. 63104
 (314) 241-7600 Ext. 257

```

2-AUG-1984 21:28:11.37                               Page 9
***** FINAL STATISTICS - data collected from 2-AUG-1984 21:12:15.38 to 2-AUG-1984 21:26:50.44 *****

--- AVE Process-Memory Counts --- Memory Utilization --- AVE Mem/CPU --- Swapper Counts ---
Proc  Bkset  Free  Modifs  Total  Pased  User  Modifs  Mem  CPU  InSWP  OutSWP  Header  Header  Swapper
Count Count  Passes  Passes  REMutl  REMutl  REMutl  REMutl  Mem  CPU  InSWP  OutSWP  InSWP  OutSWP  CPU %
-----
17    14    5955    201    63.7%  57.62  55.82  1.52    0    4    0    0    0    0    0.0%

--- CPU Idle --- CPU Busy --- CPU and I/O Overlap ---
Total  Pass  Swap  PassSw  Inter  Kernel  Exec  Super  User  Comwat  CPU+IO  CPU  I/O  Multi  CPU+IO
Idle  Wait  Wait  Wait    Stack  Exec  Super  User  Comwat  Idle  Only  Only  I/O  Busw
-----
0.02  0.02  0.02  0.02    0.9%  3.22  2.42  0.13  93.2%  0.02    0.01  94.52  0.02  2.02  5.52

--- Passes Rates (per second) ---
Pass  Suctem  Passes  Read  Passes  Write  Free  Modifs  Red  Dzero  Gvalid  ReadIN  WritIN  Read  Rel  DeadPT
Faults Faults  Read  I/Os  Written  I/Os  List  List  List  Faults  Faults  Fros  Fros  Error  Peno  Scan
-----
14.4  0.1  1.0  0.1  0.0  0.0  3.2  3.0  0.0  7.5  0.8  0.0  0.0  0.0  0.0  0.0

--- I/O Rates (per second) --- File I/O Rates (per second) ---
Direct  Buffrd  Lonsae  Mailbx  Mailbx  Window  Window  Split  Erase  File
I/Os  I/Os  Trans  Reads  Writes  Mitt  Turns  I/Os  I/Os  Opens
-----
1.3  0.8  1.1  0.1  0.1  1.3  0.0  0.2  0.2  0.1

--- File Cache Attentat Rate (per second) --- File Cache Effectiveness ---
Dir  Dir  Data  Quota  File  File  Bit  Dir  Dir  Data  Quota  Id  Hdr  Extent  Bit
FCB  Data  Quota  Id  Hdr  Extent  Map  FCB  Data  Quota  Id  Hdr  Extent  Map
-----
0.1  0.4  0.3  0.0  0.4  0.4  0.0  0.02  99.82  100.02  100.02  89.02  100.02  0.02

--- Lock Rates (per second) ---
New ENG  Converted ENG  DEQ  Blockins AST  Directory  Deadlock
Functions  Messages
-----
Local  1.0  Local  0.9  Local  1.0  Local  0.0  In  0.0  In  0.0
In  0.0  In  0.0  In  0.0  In  0.0  In  0.0  In  0.0
Out  0.0  Out  0.0  Out  0.0  Out  0.0  Out  0.0  Out  0.0
    
```

Final Statistics Report Example

THE IMPORTANCE OF POWER

(Computers can't run without it, and dp managers shouldn't run
computer centers without knowing the basics.)

by Dan M. Bowers

(Reprinted with permission of DATAMATION Magazine. Copyright by
Technical Publishing Company. A Dun and Bradstreet Company (1985)
- all rights reserved.)

Electrical power in a computing system is usually taken for granted, like the cabinetry, the instruction manual, and the dog wagging its tail when we get home. Compared with MIPS, CMOS, POKES, mice, and voice synthesis, the provision of electrical power seems mundane and routine; one visualizes a technology unchanged since Edison and Steinmetz.

But the subsystem that provides electrical power is more than just an umbilical to the wall plug; it is one of the four basic elements without which the computer cannot operate--try playing Ms. Pac-Man on your expensive pc in the absence of the processor, memory, I/O, or power. And power technology has advanced through the years along with computer technology, although it is still based on the work of Thomas A. Edison and Charles P. Steinmetz, just as ours still depends on the insights of Claude E. Shannon and John Von Neumann.

In this age of digital circuitry that practically never fails, and peripherals that are orders of magnitude more reliable than only a decade ago, the major adverse influences upon our computer systems come from outside the system: the operators, the physical environment, the communications lines, and perhaps the greatest of all these, the power source. Moreover, the very advances in semiconductor technology that have made the modern computer age possible have increased our susceptibility to the age-old vagaries in the power source. A 1950s-era vacuum tube circuit might require a 40-volt logic signal, and thus would not even notice the five-volt spike that sends today's micros off the wall. The importance of a particular mote depends upon the size of the eye of the beholder.

Thus, it is important to remember the fact that power is supplied to a computer by a power subsystem, a vital one of many subsystems that make up the total computer system. We shall analyze the constituents of this subsystem and study its potential problems in terms of their sources and effects. We will also consider the practical remedies that exist in system design, paying particular attention to the ultimate protection, an uninterruptible power system (UPS).

Fig. 1 illustrates the total scope of the power subsystem. Power problems can and do originate in any portion of this network. Since most of its components are external to the computer system they are also beyond the control of the computer system designer. What the designer must concern himself with are those problems involving system constituents that he can control. The user could, of course, take control of the entire power subsystem by installing his own generator and having his own oil well, waterfall, or windmill, but few find it economical or convenient to do so.

Primary energy source. Electrical energy does not occur in nature in any convenient form, and therefore an available primary source must be used. Except for nuclear fission and tidal flow, all primary sources use solar energy, either direct (including wind) or stored (coal, oil, natural gas, hydroelectric). Arranging for the primary source is the power company's problem, but if it runs out, the user inherits the problem. And if the sun goes out, we're all in big trouble.

Converter to electrical power. Usually called a generator, this component is always a rotating machine that Edison and Steinmetz would still find quite familiar, except in the case of solar cells, where the conversion is done directly. In most power generation, there is a triple conversion of the energy: from the prime source to heat (steam), which is then used to force rotary mechanical motion of the generator, and then to electrical power.

Distribution system. Alternating current (which would be new to Edison but not to Steinmetz) is distributed over a system of wires to all users of the power company's product, and even to other power companies and their users. Our computers can thus be viewed as being connected by wires to virtually everyone else on the continent, and thus can be affected by them. As a practical matter, we are at the mercy of only our nearest neighbors, but even they can number in the tens of thousands.

CONTROL OVER OUR DESTINY

AC to DC converter (commonly called our system power supply). At last we can begin to control our own destiny, and here is our first line of defense against the problems sent to us by the world without. The designer must ensure the power supply chosen solves problems rather than creates new ones.

DC distribution system. Power distribution in a digital system is a science of its own. All of the benefits of a proper power supply, line regulators and filters, and even a UPS, can be negated by an inadequately designed and implemented distribution system.

AC distribution system. AC, rather than DC, is usually used to run motors and some relays and lamps in a computer system.

These components are inherently less susceptible to the usual vagaries of the power line, but nonetheless require design attention. They can also be local sources of the same kinds of disruptions that come in over the line.

In the power subsystem there are many generators of noise and disruption, and constantly varying loads, most of which are separated from the power plant by the electrical bulk of the distribution system. Therefore, no matter how pure the output of the generating station, the power delivered to the user can have all kinds of anomalies (see Fig. 2). These influences appear on the power line in the following ways, with the effects described, and require the remedies indicated.

Noise, spikes, and transients. These are high-frequency phenomena, and thus are of relatively local origin since high frequencies, like delicate wine, do not travel well over long distances. Among the sources are sparking brushes on DC motors and power tools, the chopping effect of switching power supply in the next room, X-ray equipment, capacitor charge-ups, poor grounding of electronic equipment, relay and controller closures, and atmospherics. At the very least, these enemies can couple unwanted signals into the computer system and cause errors, interruptions to processing, and loss of data. At the worst, they can destroy computer circuitry since they can have amplitudes in the kilovolt range. Radio frequency interference (RFI) filters and isolation transformers are the principal weapons against these enemies, along with careful design of the system layout and wiring.

Surges and sags. The voltage on the power line can vary up or down by 25% or so, due to other large loads being removed from or being put on the system, power company problems with generating equipment, or direct lightning hits. Their duration can be from milliseconds to minutes; a long-term sag is known as a brownout, and can last for hours. Both overvoltages and undervoltages can cause processing errors and loss of data, and damage to motors and power amplifiers can also occur. A voltage autoregulator and a power supply with good regulation specifications are the best defenses.

Frequency variation. Serious damage could result if there were even a 10% error in the frequency of the AC power, but as a practical matter it never happens in this country--how often does your electric clock show the wrong time? In foreign countries, or in mobile and portable applications and self-operated power systems, the possibility of frequency variation must be allowed for in special power system design. Special power supplies can be obtained, and some types of UPS can be made impervious to line frequency.

CONCERN DICTATED BY USE

The application for which a computer system is intended should dictate the amount of concern and cost that is justified in power protection equipment. No system should be left susceptible to random errors or physical damage, so every system should be protected against noise, spikes, transients, surges and sags; a first-class power supply, an RFI filter, and an isolation transformer should be routinely provided.

Thus, the major design decision is how to deal with interruptions, and the design thinking in this area has not changed substantially in 20 years. Any of four modes of operations can be incorporated.

1. Relax and enjoy it. If the functions being performed are not critical, if the data being processed can conveniently be recreated, if the system is satellite to a host that provides the primary intelligence, there may be no need to provide any means to continue processing during power interruption, nor justification for the cost of doing so. Nearly all personal computers, remote terminals on large systems, and data entry stations are in this category.

2. Orderly shutdown. If the system need not continue operation, but the data being processed at the moment of power failure are critical, then power must be maintained long enough to complete current processing, save data, close files, place external machinery into a safe condition, and so forth. Examples are real-time data collection, machine tool control, and most accounting and business applications.

3. Graceful degradation. This pretty sounding term, AKA fail-soft, means the orderly shutdown of the high-power consumption portions of the system, but with sufficient power provided so that basic functions can continue. For example, in a facility management system, the disk memory will contain sophisticated programs for boiler and chiller sequencing, enthalpy management, and multilevel personal access; under a gracefully degraded mode, the disk would be shut down and the building would operate from ROM-based programs providing standard heating and cooling settings, single-level (or no) access, and so forth. One perfectly acceptable form of graceful degradation is to shut down into a mode that allows operations to be performed manually.

THE MOST EXPENSIVE SYSTEM

4. Damn the torpedoes, full speed ahead. The ultimate, and most expensive, power system is one that continues to provide full power without interruption, heedless of any hiatus on the power company's supply line: a UPS. The designer's challenge in this instance is to provide adequate protection without squandering money. For example, for how long a period of time will power be needed? A day? A week? An eon? Should you provide a second UPS to take over if the first UPS fails?

Readers interested in the details of power protection and filtering, UPS operation and selection, and so on can choose from a staggeringly extensive volume of articles. Rather than add unnecessarily to that volume, let's continue to view the UPS as a problem in effective system design.

There really are only three parameters of importance in selecting a UPS.

1. Failover. By definition, there cannot be any interval of time, however small, during which power is absent due to switchover from the power line to a backup source. This means that the backup source must be active and on-stream (and therefore itself subject to failure) at all times. There are three ways to accomplish this, but only the first two are commonly used:

--Entire computer runs from a battery kept continuously charged by the line. This is known as continuous service or direct (and some kinds of reverse transfer) UPS.

--Battery backup is provided at a slightly lower voltage than the DC from the power supply, and provides uninterrupted power when the normal source fails. This leaves the AC motors without power after their inertia has dissipated, and thus provides only graceful degradation or orderly shutdown. skip 1 --Rotating motor-generator continuously operates in readiness to take over supply of power. This is a heroic measure for very critical application.

2. Amount of power. Should the UPS supply sufficient power (typically 300 watts to 400 watts for a microcomputer system) to operate the entire system as though primary power had not failed, or should some form of graceful degradation be instituted?

3. Time duration. For what period of time should the UPS provide full power? Should there be a degraded operation on reduced power if the outage persists for a period of minutes to hours, although theoretically there is no time limit if the user continues to provide enough batteries. There are also nonautomatic remedies for the long term, such as a dual set of batteries and a means to charge the off-line set. Multistage solutions also exist, like a motor generator that can be brought on-line after a few minutes, with only the batteries having to serve during that interval.

Since the available UPSs almost universally rely upon batteries as the power source, it is important that the user and system designer develop some familiarity with the various kinds of batteries and their areas of usefulness.

Beginning at the bottom, the relatively new lithium-based batteries (lithium-iodide, lithium-carbon-monofluoride,

lithium-manganese-dioxide, et al) can supply power in the range of microwatts to half a watt, for a time ranging up to two years or so. They are used for computer memory backup, LCD clocks and watches, and pacemakers. They are very small in size and are not rechargeable.

Also very small and not rechargeable are the silver oxide (microwatts for a few months) and mercury (up to half watts for a few days) cells. They are useful for the same kinds of low-power applications.

The familiar carbon-zinc dry-cell flashlight battery is also not rechargeable, notwithstanding the come-on advertisements for rejuvenators in Popular Science and Road and Track magazines. They can deliver a few watts for several hours, and can be ganged to provide backup for a substantial portion of a microcomputer system for a few days, at nominal cost. Alkaline cells come in both rechargeable and nonrechargeable versions, and have similar capacities and applications, and a roughly equivalent cost-per-watt.

Nickel-cadmium batteries are the current favorite for rechargeable backup of low-power portions of systems. They can provide up to a watt or so for several hours, can conveniently be ganged for more power or longer supply time, and require relatively simple charging circuitry.

For bulk power, the only realistic source is the lead-acid automobile-type battery, which now comes in sealed versions and with gelled electrolyte instead of splashy acid. It has the most attractive cost-per-watt in high-power applications, and is rechargeable, although a relatively sophisticated charging circuit must be used to avoid both overcharging and undercharging which can greatly shorten the battery life. (On the other hand, if those creative people in Detroit who design spare tire and jack stowage--which, like road maps, can never be put back the way they came--can design a charger for this battery, this problem would be solved. How difficult can it be?) Single units can deliver on the order of 20 watts for a day, and these batteries have been ganged by the roomful for many years to provide power in the kilowatt range for large systems.

CRITIQUE DESIGNER DECISIONS

A thorough treatise on the selection and design of battery backup systems is interesting and useful, but beyond the scope of this article. Most users will purchase an entire packaged UPS, including the battery provisions, and the necessary engineering decisions will have been made by the UPS designer. But users should become educated enough to critique those decisions when selecting and evaluating UPS vendors.

Growth of the power conditioning equipment business, including everything from filters to UPSs, is currently at a 15%

to 18% per year rate, and will probably increase somewhat as a higher percentage of new computing equipment is so equipped. The newly opened market for UPSs of under 1KW to service microcomputer systems is giving the industry a boost.

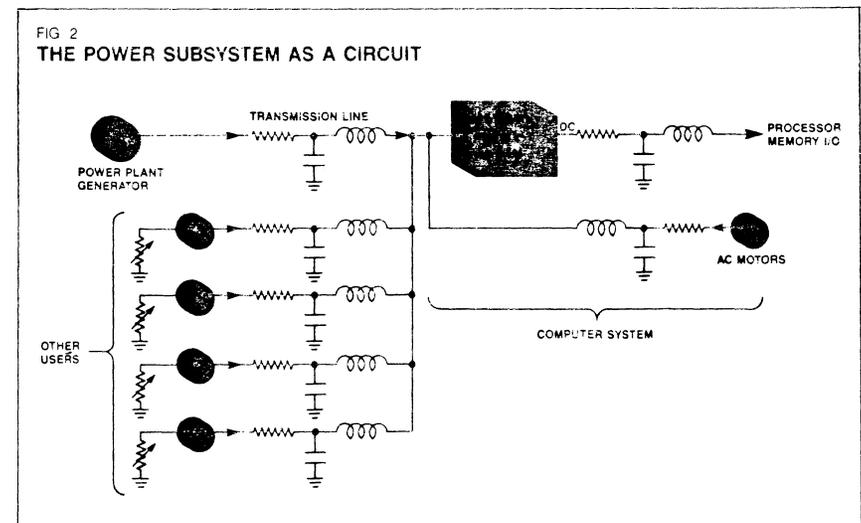
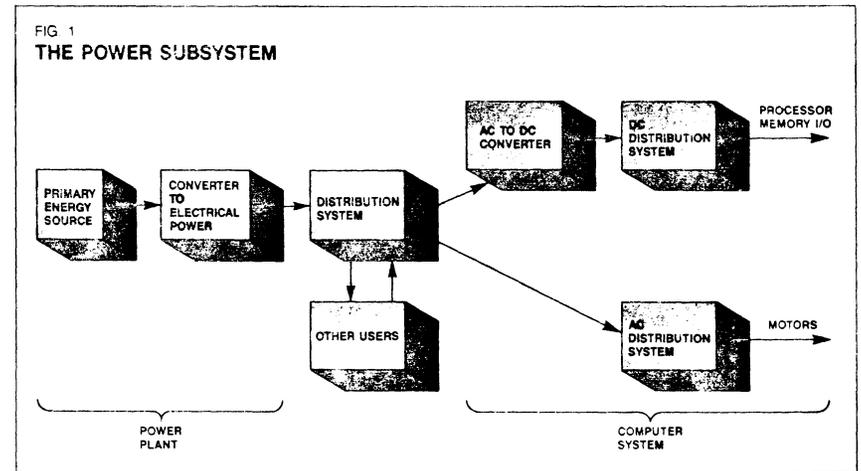
Batteries have come a long way, but have gone about as far as they can go with respect to major breakthroughs. They will continue to form the backbone of most UPSs and degraded operation and orderly shutdown systems, since there currently is not a realistic alternative. Batteries, however, are not a good solution for the long-term (beyond a day) UPS, and there will likely be increasing attention paid to combination units, with batteries as the short-term (minutes to hours) solution, and a rotating generator to take over for the long haul. This will also provide desirable redundancy in the UPS. One cannot discount the use of alternative energy devices (fuel cells, nuclear, solar) for eventual UPS use, but right now there is no serious prospect.

Power required from the UPS will continue to be reduced, as the more-circuitry-on-a-chip syndrome continues, and CMOS circuitry maintains its growth in popularity. The eventual replacement of the rotating memory by bulk solid-state memory (which has been widely predicted for a decade by many in the industry) will be a notable milestone in power reduction.

System design to incorporate graceful degradation of operation is an important step that can be taken today to reduce the size and cost of UPSs, but it is seldom done because it's easier to throw money at the problem and acquire a full-system UPS; this is commonly known as the government-at-Washington approach. Computer system designers and UPS vendors should both be in the forefront of this effort.

Lastly, like everything else, UPSs will become more intelligent. Built-in diagnostics to test the failover mechanism, the readiness of the battery bank and each of its cells to perform its function, are already becoming available. Some big-system UPSs incorporate sensors that monitor the state of the computer system and the power line, and anticipate the need for backup power. The microcomputer-controlled UPS will give us these functions and more over the next few years.

Dan M. Bowers is president of Bowers Engineering Co., Southport, Conn., which has been designing new computer systems and doing recovery and remedial work on existing systems since 1966. Bowers has been active in dp since the 1950s.



WIRING THE CAMPUS

May 8, 1985, Digital Equipment Corporation and Washington University (St. Louis, MO.) signed an agreement to develop and implement a computer system network to encompass the entire Washington University campus. The \$15 Million dollar agreement would in effect "wire" the entire campus and allow high speed data, voice and video communications between departments.

Several other Universities have entered similar agreements with DEC, most recently Indiana University. Although DEC's main emphasis in all such agreements has been on "wiring the campus" with DECnet, these agreements are and will provide many other benefits to the Universities and to Digital.

For instance, the DEC/Washington University agreement is really matching funds grant that allows a 50% discount on the list price of DEC products. (a modified, no cash up front matching funds grant.) So, Washington University can purchase 15 Million dollars worth of computer products for 7.5 Million dollars. A rather substantial savings.

Such price reductions make it possible for many University departments to purchase a VAX, a MicroVAX, or high end PDP 11 system which they could not otherwise afford without this program.

That means that all those who are plodding along with Apples, IBM PC's, older PDP 11's, or hodge podge conglomerations of other computer hardware, and need more computing power, can think seriously about a VAX, a PDP 11, or a Pro 380 of their own, or pooling their funds within a department in order to purchase the machine that can meet their demands.

This was the case in our department. We were considering (and needing) a VAX but were having trouble fitting it into our budget. Under this program, we were able to purchase the system and some extra software without any problem. All of which will aid us greatly in carrying out our research in a timely fashion and allow us to tackle more complex projects.

This agreement will also make available more computers and more computing power available to students, especially computer science and engineering students. Equipment dollars will go much further with the 50% discount.

With the advent of the MicroVAX II and the coming of the MicroVAX III, labs and departments that never dreamed of having such computing power available, will be making a VAX login a part of their morning.

University based researchers will benefit greatly since most research projects tend to eat up whatever computing power is available. If its there they will use it. Unfortunately, the majority of researchers suffer from equipment budget shortages. Under this program those constraints will be eased quite a bit and allow many researchers to purchase the tools they need to carry out their research.

Professors and students will benefit from the program, too. With more computing resources available many non-technical computer users will put the power of computing to use in educating students. This will result in the development of computer based education and increase our understanding and help determine the best methodologies to use in implementing computer based instruction. These developments in computer based instruction will aid the schools involved, the student, and DEC in designing and developing its own course ware, or marketing the University developed packages.

Networking will also provide major benefits to the campus community as well. Timely communications are important in any business and to Universities as well. Telephone tag is a common occurrence on a University campus. Professors must go to classes, counselors are busy seeing students and potential students through out the day, and staff members often must run between offices and departments.

Having a campus network with mail facilities, voice and data communications would be an obvious boon in such an environment. It allows people to move about freely, not tied down to their office phone, but prevents losing important or valuable information.

Adding to those capabilities, video communications, the possibilities of video conferences, distributed classes, transfer of images (engineering graphics, medical images, etc.), distributed access to expensive graphics devices, etc., you get a idea of the power, flexibility, and the time savings such a network brings to a campus.

Computers being heavily loaded could "borrow" some computing power from systems that are underloaed on the network. Researchers involved with other researchers will be able to "share" data as it is processed, all reveiving data at the same time, communicating with each other, and never leaving their office to do so.

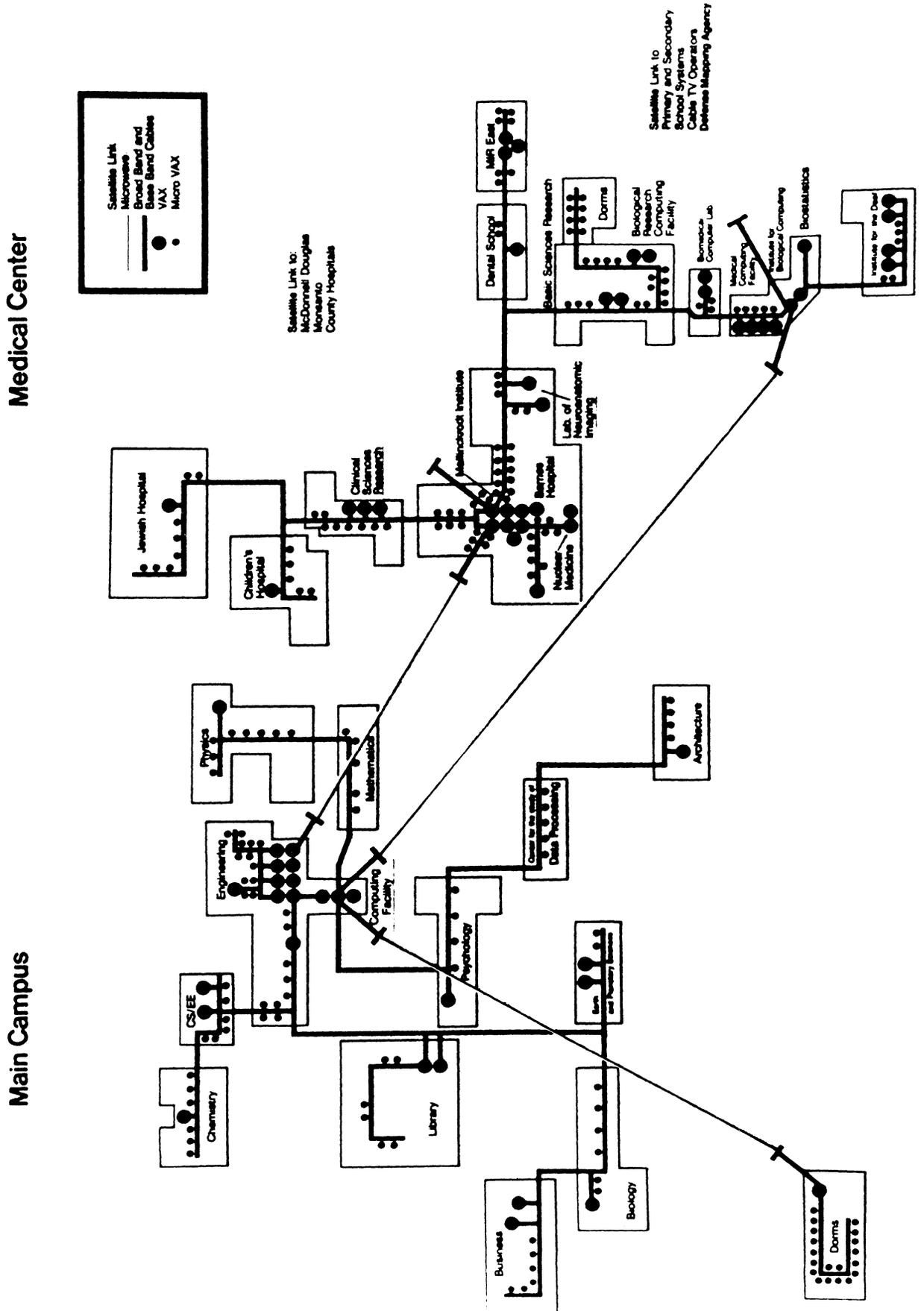
If I seem to be painting a rather rosey picture, I am. There will be the problems to overcome, the system crashes, the "this is not working right" complaints, but from an overall perspective the future of computing on the campus' involved looks very promising.

If only 1/2 of the possibilities envisioned become realities, we will still see a vast ammount of progress in computing as a result of these joint agreements with DEC.

Gregory N. Brooks

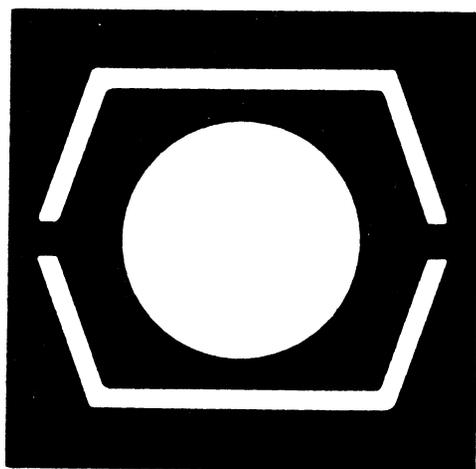
The following page shows the progress envisioned hoped to be gained at Washington University by 1988.

Washington University Master DECNET Plan 1988





UNISIG



DECUS



UNISIG Chair

James W. Livingston, Jr.
 Measorex Corporation
 1 Results Way
 Cupertino, CA 95014
 408-255-1500 X4468
 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

Bruce A. Bergman
 2235 Meyers Avenue
 Escondido, CA 920251070
 619-741-8825

UNISIG Administrative Daemon

Dorothy Geiger
 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!dedwest!ucbvax)
 !sdcsvax!mplvax!cdl

UNISIG Usenet Liason

Joe Kelsey
 John Fluke Mfg. Co., Inc.
 PO Box C9090, M/S 223B
 Everett, WA 98206
 206-356-5933
 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 Blvd.
 Merrimack, NH 03054
 603-884-5702
 decvax!maclean

TOOLKIT is the newsletter of UNISIG, the DECUS UNIX-oriented special interest group. **TOOLKIT** articles range widely over UNIX† topics: DEC's ULTRIX-11 and ULTRIX-32, BSD 2.9 and BSD 4.2, device drivers, operating systems comparisons, varieties of user interfaces, performance analysis, multi-operating system applications, UNIX standards, local area networking, long haul networking, uucp, USENET, UNIX communication with DEC and IBM operating systems, UNIX hints and kinks, C, f77, ratfor, lex, yacc, nroff/troff, developments in the Software Tools movement, site surveys, DECUS Symposia, and more.

The editors are:

William Toth
 Harvard-Smithsonian Center for Astrophysics
 60 Garden Street, P-353
 Cambridge, MA 02138
 617-495-7181
 harvard!hrvsmth!toth

Bruce A. Bergman
 2235 Meyers Avenue
 Escondido, CA 920251070
 619-741-8825

We invite readers to submit articles, book reviews, software reviews, and thoughtful letters for publication in **TOOLKIT**. Text should be submitted in machine-readable form, if possible. We can read RX02 floppy disks and half-inch magtape (800, 1600, or 6250 bpi); computer-to-computer text transfers via telephone can be accomplished by using **uucp** or **KERMIT** or by arranging ordinary dialups with either of the editors.

† UNIX is a trademark of AT&T Bell Laboratories

FROM THE UNISIG CHAIR

James W. Livingston, Jr.

UNISIG Chair

This is the first issue of the new, improved SIG newsletter, **TOOLKIT**. I'll let the editor talk about the details of that, and what you can expect from it. My job here is to speak as the chair of UNISIG, the DECUS UNIX† special interest group, and to tell you what I see to be our purpose, our place in the DECUS U.S. Chapter, and our relationship with the rest of the UNIX community.

Also, let me note that UNISIG is a special interest group which has been around for a fair amount of time; we have found new impetus in Digital's coming out with UNIX products, and the explosion of UNIX in the industry, at large. We are a lively group, with some ideas on how to help bring together the users of UNIX on Digital hardware with Digital's engineers, product managers, sales and support people, and each other. I'll mention some particular examples of these notions later on, as well.

Before launching into the detailed discussion of specifics, I'd like to observe that the UNIX community, today, is in the middle of the greatest growth period since Western Electric decided to begin sending out snapshots of the current state of Ken, Dennis and friends' work to deserving universities. This is certainly not news to most of us, but worth mentioning to give a somewhat broader context to what I want to say.

We're all aware that there are other user groups available to users of UNIX; /usr/group and Usenix come immediately to mind. There are also a growing number of UNIX publications; I get four, at this writing. Why then, you should ask, is there a need for yet another user group, with yet another newsletter?

For my money, there are at least a couple of reasons: first, we've already been in existence for six years (not always with our present name) and we're not going to go away now, when the fun is really starting. Second, Digital have finally come out of the closet as UNIX users and vendors. What they offer seems to many to fit well the needs of a partic-

ular group of UNIX users, whom most of us in the SIG steering committee see as under-represented by the other user groups.

The folks I'm talking about are those whose machines are often Digital products, who are more-or-less engineering, scientific, manufacturing, or system-integration in their orientation, and who may or may not run Digital's Ultrix product. These are the users who may have relatively few, and often very expensive, people to support their UNIX systems, or who may feel that their purposes are simply better served by using a fully-supported version of UNIX for software development or applications. These users demand a UNIX which performs well, and which is a supported product from a first-rate vendor.

This leads me to the discussion of the topics I mentioned earlier, and I'd like first to address the issue of the SIG's purpose. In my description of the users I characterized as "under-represented", it was not my intention to suggest that the more traditional kinds of UNIX users should think themselves excluded from UNISIG, or that the we wouldn't enthusiastically welcome their participation. There are, however, forums already available to that community. It is my objective, here, in this statement of UNISIG's purpose, especially to single out a community I believe to be not now provided with a proper forum for the exchange of useful ideas and information. Our purpose, then, is to provide a means for this group of users to exchange views with Digital, with each other, and with the larger UNIX community, as well.

In this statement of purpose, I would emphasize that we hope everyone in the community of UNIX users who run on Digital hardware will join DECUS, and participate in the SIG's activities. We hope also that those OEM users, or prospective users, of UNIX, who have felt like no group quite fit their needs, will feel especially welcome in UNISIG.

Our job as a DECUS Special Interest Group, as I see it, is like that of any other operating-system-specific SIG: to provide a means whereby Digital can get input from users who are affected by Digital's decisions, and to provide a communication channel for Digital to supplement their formal means of getting specific information to users of Ultrix. We are not unlike, say, the VAX/VMS SIG, currently

called "VAX System SIG", in that regard.

We differ from other OS-specific groups, however, in that we have users whose interests include both 16- and 32-bit hardware hosts for the same operating system. That gives us a unique need to provide input to Digital on what we want from them. They, likewise, have a unique opportunity to use their significant catalog of software products and crew of talented developers to make available tools which might otherwise not be available for smaller and less expensive systems.

For their part, they have taken account of our unusual situation by promising to make Ultrix-11 and Ultrix-32 as much alike as the hardware will permit; I'd like to see us hold them to their promise. How about the rest of you? I'd like to point out, by the way, that they're already showing their willingness to provide the things that we've said we need: note that DECNet, which we asked for in large numbers, is now available from them to permit us the use of UNIX with other Digital OS's, as well as with their large and small machines.

The way we carry out our functions, as a SIG, is to sponsor paper sessions on UNIX-related topics at the biannual DECUS symposium; to publish articles in **TOOLKIT** which may be of interest to the UNIX user community in general, and the community of Ultrix users, in particular; to collect and distribute software which runs under UNIX (or look-alikes), and has been put in the public domain by Digital and others.

In addition, we sponsor, to whatever extent funds permit, other activities on behalf of the UNIX user community e.g., participation in standardization efforts for the operating system, collection of data on the performance of various UNIX implementations, coordination with other DECUS SIGs whose interests coincide with ours, and the like. Please look for more details on all these activities in the job descriptions submitted to **TOOLKIT** by the other members of our steering committee.

Our relationship with the larger UNIX community has been mentioned, in the things I've said so far. I'd like to make some additional remarks on the topic here, with an eye to greater specificity. Clearly, we have the greatest claim on the attention of those who run UNIX on Digital hardware. In addition, I think we can make the case that a fair number

of new UNIX users may want to use Digital, rather than a resident guru, for maintaining their systems. Both sets of users are likely, I think, to find the sort of information they need in the exchanges to be found under the auspices of DECUS than they are in many other places. They will have greater access to Digital engineers and fellow users of Digital products than they might at other groups' activities, if nothing else.

So what are you to conclude from all this? Let me try to make some concrete comments. As we'd all probably agree, there are enough demands on the time of most of us so that we are loath to consider yet another way to use it up. Still, I think that a case can be made for everyone who runs UNIX on Digital hardware to consider participating in our activities.

For attending symposia, the justification is simple: nowhere else can you get more timely information on what Digital's doing to the hardware of your choice. Also, if you happen to be an Ultrix-11 or -32 user, you have access to the developers of those systems; those folks are eager to hear what you think of their efforts, and ready to take back your notions to their discussions of product futures.

Certainly readers of UNIX network news know that there's a wealth of useful information to be had from that source. What all the readers of news may not have noticed is that there's a lightly-used newsgroup, net.decus, to which we, UNISIG, intend to make more frequent postings. In addition, we intend to glean from the net, news which will be useful and interesting to those of our members who have, for one or another good reason, no means of accessing it. More folks fall into that category, by the way, than many traditional UNIX users would believe. These pearls will then be contributed to **TOOLKIT**, for the enlightenment of UNISIG members who may not read the news, or not read the newsgroups that we scan. Our usenet liaison steering committee member is responsible for coordinating that job, but he'll need a great deal of help, if any netnews freaks feel the desire to take a more active role in the SIG.

We hope also to be able to create a working group to bring net.sources and mod.sources contributions into our library of public domain software, after they're verified. That activity

† UNIX is a trademark of AT&T Bell Laboratories

will be very demanding, but has potential for making much good software available where it has not been before. Any closet QA types out there?

We think we can help the community in other ways, too, and are still looking for more and better ideas, volunteers, and new things to do. If you think we've a good direction, and that you'd like to join in, or that we're full of beans, and that you'd really like to replace us with a better program, please jump in! We should all have only one objective: to make things better for the users of our favorite operating system, by using the resources of DECUS to accomplish that goal.

DIGITAL'S UNIX PROGRAM OFFICE

Roseann Maclean
UNIXIG DEC Counterpart

Digital's Merrimack, New Hampshire, facility has been home to the engineering organization that has supported Digital's UNIX users for more than six years. Originally, the team consisted of only a handful of software engineers and a like number of marketing support personnel. Today, over 140 people work in the Merrimack UNIX Program Office. They are just part of the total team committed to Digital's native UNIX software. Throughout the world, Digital has software support, hardware support, sales, and educational services personnel focused on the ULTRIX products.

Real growth started almost three years ago, when Digital announced its intent to market its native UNIX-based products, the ULTRIX offerings for the PDP-11 and VAX families. Two important activities had begun. First, the small Merrimack organization began adding additional UNIX trained software engineers and writers to its ranks. At the same time, Digital's worldwide support team began preparation for full support of the ULTRIX products.

The Merrimack organization continues to be the focal point for UNIX-related issues within DEC. The largest segment of the organization is the engineering group, UEG, which has developed ULTRIX-32, ULTRIX-32m, and

ULTRIX-11. Within UEG, there are several subgroups:

- 32-bit and 16-bit kernel development
- Languages, utilities and data base development
- Networking and communications development
- Workstations development
- Documentation
- System quality management, performance testing, and system test tools
- Engineering services
- Product management and licensing management

Rounding out the UNIX Program Office are some integral marketing functions:

- Marketing communications
- Third party acquisitions and marketing
- Strategic marketing
- Marketing technical support

At this fall's DECUS symposium, you'll have an opportunity to meet members of the Merrimack engineering team and also members of the key support groups that provide support services for all the ULTRIX offerings. UEG's attendees at DECUS are carefully chosen to give you easy access to the right people to answer most of your questions on the kernel, utilities, networking, layered products, documentation, and licensing of our ULTRIX software. Likewise, the support groups will be sending representatives to answer your questions about available services and courses. We look forward to meeting you at our sessions, in our booth, or in the UNIXIG suite.

FROM THE UNIXIG STANDARDS COORDINATOR

Jeff Gilliam
UNIXIG Standards Coordinator

As most of you are aware, a great deal of effort is being expended within the community to standardize various aspects of UNIX. ANSI, IEEE, and /usr/group are sponsoring committees to draft standards for C, the system call interface, and other areas crucial to portability. As Standards Coordinator my task will be to act as the official DECUS representative to various of these committees and report back to the membership on the work as it progresses.

FROM THE UNIXIG SESSION NOTES EDITOR

Kurt Reisler
UNIXIG Session Notes Editor

With the deadline for submissions to the UNIXIG Session Notes for the Fall '85 DECUS in Anaheim California slowly approaching, I thought I might get a head start on my nagging for submissions. The deadline for submission to the session notes is early in October. Although that seems like a long way off, deadlines have a nasty habit of sneaking up on us all.

What purpose do submissions to the session notes serve? In many cases, it is the only part of your talk or presentation that gets taken back from a DECUS Symposium. This can be due to a variety of things. Usually, there are so many session conflicts, that the attendee can only purchase a session tape and refer to the session notes for those talks that they just can not make it to. In other cases, there is just too much material to be presented at the session, and the session notes make an excellent supplement. In some cases, people attend sessions based on what is in the session notes. In any case, having something in the session notes makes your presentation that much better.

If you have submitted a session for the Anaheim DECUS, please give some thought to getting something in the Session Notes. A variety of things will do:

- hard copy of slides
- detailed diagrams
- outline of your presentation
- detailed abstract
- summation of your presentation
- where you can be reached for more information

In summary, the UNIXIG Session Notes exist to make your session better. It allows those attendees who can not make it to your talk, to take a bit of your wisdom with them anyway. And this time, you get something out of it as well. When you send me your session notes material, be sure to include your shirt size.

Thanks and see you all in Anaheim!

NETNEWS ITEMS

Joe Kelsey
UNIXIG Usenet Liaison

From: sun!di!pesnta!zencet!doug
Date: Sat Feb 5 22:28:06
Subject: New Book on UNIX

Macmillan's Small Computer Book Club has made **UNIX and XENIX: A Step-by-Step Guide** the Main Selection for March. This book gives you detailed, in-depth coverage of basic concepts, ed, vi, ex, ms, troff, and shell programming. It's filled with clear examples, carefully explained procedures, and helpful summaries. (It's also the funniest book on UNIX that money can buy.) While "the black book" may be fine for people who already know the system, this is the book for first-time users. It's been advertised in *Computer Language* and *PC Tech Journal*, and it's available in bookstores everywhere, including B. Dalton and (soon) Crown Books.

WHITHER UNIX

Steve Lazarus
UNIXIG Symposia Coordinator

AT&T presented a Business and Technical Seminar covering the direction of Unix and licensing issues on March 7-8 in Santa Clara, CA. This report covers the technical aspects of this presentation. Most of the technical innovations discussed are to be in System V Release 3. They will be supported by AT&T on the 3B machines and available in a "transition release" for vendors to port to their machines.

Introduction

AT&T's goals are to encourage broad use of Unix, establish System V as a standard, and provide customers with quality products and service. They hope to create the wave that generates the explosive growth of Unix predicted by many.

Directions

AT&T intends to maintain a continuing open architecture while shifting its prime source base to their 3B2 computer. Portability

remains a high priority. They intend to support standards, including

- /usr/group - IEEE
- International Character Sets
- ANSI-C

Standard interfaces are supported by a command line argument standard (and the getopt routine), system messages, a System V interface definition and its future, and kernel modularity.

As AT&T shifts to their own computer as a base, the VAX will be supported for "at least two years". (We don't expect to see any future enhancements for our poor old 11/70's.) AT&T will support Unix on its hardware and will, through standards and its verifications service, assist vendors in supporting Unix on their hardware.

New developments include paging, file system hardening, distributed Unix, and I/O streams. The interface specifications include the interface definition, a System V verification service, and support for international character sets. At the applications level, file and record locking is provided, the user interface is being improved, and a toolchest of applications is being provided.

System V Interface Definition

The purpose of this definition is to provide a common environment for all applications and users of System V implementations, facilitate portability of application code, and to partition services by functionality. A System V Interface Definition has been published by AT&T. It conforms to the /usr/group standard except for:

- Writing to a pipe,
- Scanf inconsistencies,
- function return types, and
- 5 error return values.

Efforts are under way to resolve these differences and to track future standards efforts through active participation.

To these ends, AT&T is establishing a System V verification service. This will allow vendors to establish conformity. Microsoft has committed to evolving Xenix to meet the AT&T standard.

Networking Plans

AT&T'S philosophy is to provide universal connectivity through standards, an open

architecture, and a networking tool kit based on standard interfaces. The key element of this is Dennis Ritchie's stream I/O system. This system is described in the October 1984 issue of the AT&T Bell Laboratories Technical Journal. The network model to be followed is that of evolving through the International Standards Organizations' (ISO) Open System Interconnections (OSI) protocols. AT&T expects to be able to make networking modular through standard interfaces between protocols and drivers, protocols and protocols, and processes and protocols. They were non-committal on what protocols would be initially supported although they did acknowledge markets for TCP/IP, XNS and SNA.

Distributed Unix System

Feature of the distributed Unix system are transparent remot file access, special devices and named pipes, streams-based networking, administrative support, and recovery. Remote access is established by mounting a branch (not necessarily the root) of a systems file tree onto a mount point of another systems file tree. Administrative features include selective resource sharing, user/group id mapping, network monitoring, and an on-line resource directory.

Languages

Languages available or planned are C, C++, Fortran, Pascal, and Basic. C++ is planned only for the 3B series. The microports will include only C and Fortran. C cross compilers for the 6800, iAPX, and 3B2/3B5 exist or are planned.

Improvements in C compiler performance and support are planned for the 3B computers. The preprocessor will add an "#ident" directive for configuration management. This identification will be in the object modules will not take up any run time space. Support for shared libraries, /usr/group standard libraries, and migration toward ANSI Standard C is planned. Support for ANSI C will add support for additional data types and typechecking of function arguments. The later will require declaration of system library functions in the system header files. A possible introduction strategy is first an ANSI lint option and a separate add-on ANSI C compiler (with perhaps a C++ option). The standard C compiler will then add the new header files, then

advisory type checking, and then full ANSI compliance. Availability of C++ is targeted for 1986.

User Friendly Features

Objectives are to make System V the standard basis from micros to mainframes. This requires that Unix be made easier at all levels. The approach is to separate the functionality from the presentation. There will be a standardized command syntax and standardized error messages. Front end interfaces will be defined to allow tailoring to particular user groups and computing environments. We can thus expect to see various menu driven and graphically oriented front ends. AT&T is evaluating the current trends of menus, windows, the desk top metaphor, and icons.

The new error message format will include the error source, a severity code, a problem description, a suggested action, and a key to the help facility. The error handling standard is currently included in the System V Interface Definition and has been submitted to /usr/group.

The help facility will provide access to help in a much more structured way than the current "man". It is now available on the 3B2 supporting 40 commands. A tool is provided for adding additional components. General availability is planned for late 1985 as part of the transition release. Future extension will support additional commands, the error handling standard, and concurrent multiple foreign languages.

A "Simple Admin" menu now exists on the 3B2 and will be generally available in late 1985 as part of the transition release. This facility provides a menu front end to Unix administrative tasks.

Internationalizing Unix

Internationalization is intended to provide the framework and tools to support local character sets, error messages in local languages, and a multi-lingual help facility. Support of other character sets requires that the 8'th bit usage be cleaned up. (The shell, for example, apparently sets the 8'th bit to tag quoted strings.) The 8'th bit can then be used to indicate that an alternate character set is to be used. The international extension is to be specified in Volume 2 of the System V Interface Definition.

The Unix System Toolchest

The toolchest is a means for electronic distribution of Unix tools in source form. Many internal Unix tools (e.g, the Korn shell, Montgomery's Emacs) have been placed in the toolchest. For each tool, a one time license is paid for unlimited internal use. Binary sub-licensing is also available. Access to the toolchest will be available for all System V source licenses in April 1985.

IN OUR NEXT ISSUE...

We will report on highlights of the 1985 Summer Usenix Conference which was held from June 11 to 14 in Portland, Oregon.

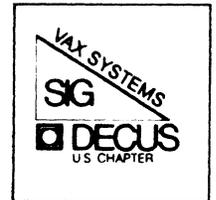


NEWSLETTER OF THE VAX SYSTEMS SIG



Our Mascot

Pageswapper



How to Backup to Magtape over DECnet	VAX-3
Plugging a Security Leak in CERBERUS	VAX-7
The Bedford Store Is No More	VAX-9
More on VMS Timekeeping	VAX-10
VAX-11 RSX Device Name Performance	VAX-11
A Proposal on RMS	VAX-19
More on a Comprehensive Backup SIR	VAX-22
LeLisp: A LISP Environment for VAX/VMS	VAX-23
"It's in the Microfiche..."	VAX-27
VAX System SIG Tape Time Bomb	VAX-31
VMS 4.2 XQP breaks previous privileged code	VAX-32
An Electronic "Campground" For VAX Users	VAX-32
Building Efficient Images on VAX/VMS	VAX-34
Global Section Problems	VAX-41
System Improvement Request Ballot	VAX-44
VAX System SIG Committee List	VAX-73
INPUT/OUTPUT	VAX-77
Steering Committee Nominations	VAX-89

Forms at the End

VAX SIG Steering Committee Nomination Form	QU-17
VAX Systems SIG Fall 1985 SIR Ballot	QU-19
INPUT/OUTPUT Submission Form	QU-21
System Improvement Request Submission Form	QU-23

VAX SYSTEM SIG COMMITTEE LIST

Chairperson

Margaret Knox
University of Texas
Austin, TX

Security

C. Doug Brown
Sandia Labs
Albuquerque, NM

System Improvement Request

Gary Grebus
Battelle Columbus Labs
Columbus, OH

Real Time/Process Control

Larry Robertson
Bear Computer Systems Inc.
North Hollywood, CA

Symposium Coordinator

Jack Cundiff
Horry Georgetown College
Conway, SC

Hardware

James R. Cutler
Software Results Corporation
Columbus, OH

Network

B. Hancock
Sohio Petroleum Company
Dallas, TX

Graphics

P. Sandwell
Seismograph Service Corporation
Tulsa, OK

Newsletter Editor

Lawrence J. Kilgallen
MIT Station
Cambridge, MA

Data Management SIG Interface

Doug Dickey
CTEC, Inc.
McLean, VA

Foreign Devices, Hardware/Software

R. Haydt
Information Consultants International Inc.
Ormond Beach, FL

LUG Coordinator

David Schmidt
Management Sciences Associates
Pittsburgh, PA

Handout Editor

Tom Danforth
Woods Hole Oceanographic Institute
Woods Hole, MA

Migration and Host Development

Jim Downward
KMS Fusion Inc.
Ann Arbor, MI

Member-At-Large

Jeffrey S. Jalbert
JCC
Granville, OH

Advisor

Al Siegel
Battelle Memorial Institute
Columbus, OH

Librarian

Joe L. Bingham
Mantech International
Alexandria, VA

Education

Dan Fleury
University of Hartford
West Hartford, CT

VAXcluster Working Group Chair

Ken Johnson
Meridian Technology Corporation
St. Louis, MO

Artificial Intelligence

D. Slater
Mantech International
Alexandria, VA

Publications Coordinator

Don Golden
Shell Development Company
Houston, TX

Real Time/Process Control

Dennis Frayne
McDonnell Douglas
Huntington Beach CA

Vice Chair and Working Group Coordinator

Ross W. Miller
Online Data Processing, Inc.
Spokane, WA

Languages and Tools

Louise Wholey
Measorex Corporation
Cupertino, CA

System Management

Joe Angelico
US Coast Guard
New Orleans, LA

Internals

Carl E. Friedberg
In House Systems
New York, NY

VAXElan

Bob Robbins
Array Computer Consultants
Sarasota, FL

Office Automation

Douglas J. Wilson
MIT Joint Computer Facility
Cambridge, MA

Member-At-Large

June Baker
Computer Sciences Corp.
Falls Church, VA

Commercial

Bob Boyd
GE Microelectronics Center
Research Triangle Park, NC

Volunteer Coordinator

Elizabeth Bailey
Tennessee Valley Authority
Muscle Shoals, AL

TOPS-VAX

Mark Graff
M/A-COM Linkabit, Incorporated
San Diego, CA

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. Please do not submit program source, as that is better distributed on the VAX SIG tape.

Material for "(THE (LINKED LIST))" section of the Pageswapper (pertaining to Artificial Intelligence) should be sent to:

Terry C. Shannon
Technical Editor
THE DEC* PROFESSIONAL Magazine
921 Bethlehem Pike
Springhouse, PA 19477
USA
(215)-542-7008

Material for "The DBMS Monitor" section of the Pageswapper (pertaining to VAX-11 DBMS) should be sent to:

Julie Llewellyn
United Technologies Microelectronics Center
1365 Garden of the Gods Road
Colorado Springs, CO 80907
USA

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.

How to Backup to Magtape over DECnet

Allen Rueter
Mallinckrodt Institute of Radiology
510 South Kingshighway Boulevard
St. Louis, MO 63110

In our shop we have 2 VAXs and 3 uVAX IIs but only one has a tape drive. Our 11/730 (SARA) is of the R80/RL02 variety. It would take a lot of RL02s to back up an R80 and even more if you plan to keep 2 cycles. At the Spring 84 DECUS I talked to a man, who works for Evans & Sutherland, and found out some interesting things you can do over DECnet at the DCL level. DEC does not support running backup with output to a tape drive on another node but at a DECnet session we were told you are allowed to write to a mail box! The problem with Backup is it wants to write variable length records. When access the tape over DECnet you need to write fixed length records to tape. This problem can be cured by using the utility CONVERT, with FDL files describing the input and output. On our 730 we have the following file which backs up the R80 to 2 tapes.

BckUpDQA0.com

```
$ ! backup dqa0 to lucy::mta0:
$ ON ERROR THEN GOTO ERROR_EXIT
$ ON CONTROL Y THEN GOTO ERROR_EXIT
$ OPEN/WRITE RMT LUCY::"0=NETDCL2"      ! 1) login via proxy
$ TYPE SYS$INPUT:                       ! 2)
```

Please mount tape DQA01 on lucy::mta0:

```
$ INQUIRE yn "Press return when ready to continue"
$ ! verify mount and init
$ WRITE RMT "Moun MTA0 DQA01"          ! 3) mount the tape
$ READ RMT s                          ! to make sure its the
$ WRITE RMT "Dism/nounlo MTA0"        ! right one
$ READ RMT s
$ WRITE RMT "Init MTA0 DQA01"         ! and initialize it.
$ READ RMT s
$ TYPE SYS$INPUT:                     ! 4)
```

Please mount tape DQA00 on Lucy::MTA0:

```
$ INQUIRE yn "Press return when ready to continue"
$ ! verify mount and init tape 0
$ WRITE RMT "Moun/assist MTA0 DQA00"
$ READ RMT S
$ WRITE RMT "Dism/nounlo MTA0"
$ READ RMT S
$ WRITE RMT "Init MTA0 DQA00"
```

How to Backup to Magtape over DECnet

```

$ READ RMT S
$ CLOSE RMT                ! 5)
$                          ! 6) do the backup
$ BACKUP/IMAGE DQA0://IGNORE=INTERLOCK -
    LUCY::"0=DQA0"/SAVE/BLOCK=4096
$ EXIT
$ERROR_EXIT:
$ CLOSE RMT
$ EXIT

```

The above command file does the following:

1. Starts process on the other node to process dcl commands via proxy access. For more info on proxy see commented out help in sys\$help:uafhelp.hlb
2. Tells the operator to mount the 1st tape to be zeroed.
3. Tell the tape node to mount the tape to verify it's the right one. Dismount it and initialize it. The '\$ READ RMT s' commands are used to keep things synchronized otherwise the 730 writes the commands faster than the 750 processes them and the messages on the console get out of sync with reality.
4. Mount and initialize the other tape.
5. End the network dcl process with the correct tape on the drive.
6. Backup through CONVERT to tape.

If our operator is on his toes, the backup takes 1 hour over Ethernet to a 75 ips Kennedy 9100. In addition I set large scratch files to nobackup with the set file command, such as PAGEFILE.SYS and SWAPFILE.SYS to reduce the time the backup takes.

The NETDCL2 file which process the command from BCKUPDQA0 is as follows:

```

$ ! process dcl commands over the network and reply with status
$ ! to start this try OPEN /WRITE RMT node::"0=NETDCL"
$ SET VERIF
$ OPEN RMT SYSSNET:/READ/WRITE
$NXT_LINE:                ON ERROR THEN CONTINUE
$                          READ RMT LINE/END=EXIT
$                          SET VERIF
$ 'LINE
$                          SET NOVERIF
$                          S = $STATUS
$

```

How to Backup to Magtape over DECnet

```

$                               WRITE RMT S
$                               GOTO NXT_LINE
$EXIT:                          EXIT

```

It simply takes in each command, executes it and replies with the status. Again this is to help keep things synchronized. A log of what happens (for debugging) can be found in NETSERVER.LOG in the default directory under which the process was started. These command procedures are somewhat error prone. We usually just have the operator start all over if something goes wrong.

The DQA0 command file which mounts the tape and invokes CONVERT is a follows:

```

$ ! network tape drive
$ SET VERIF                ! for debugging
$ SET PROC/PRIV=VOLPRO    ! the account needs
$                          ! volpro or setprv
$ ALLO MTA0
$ MOUNT MTA0 DQA00/BLOC=4096 ! at 8192 we got over runs
$ CONVERT SYSSNET:/FDL=NETIN MTA0:SARADQA0.DAT/FDL=MTA0OUT
$ DISM MTA0
$ DEALLO MTA0
$ EXIT

```

The DQA0.com file does have potential problems. The 1st problem being some one could allocate the tape drive between initializing and the backup starting. Secondly if the tape node system is slow, you may get a LINKEXIT error, which can be caused by the incoming and outgoing timers being set to low (Use NCP SHOW EXEC CHAR to check). Other restriction include no generalized method to support different tape labels, so for now we just keep different COM files. I suppose one could use NETDCL2 to create some logical names at the group level and then generalize this command procedure further. Currently we only have need to backup our R80. Those of you with 11/780's or faster machines may be able to block the records at 8192, on our machines things would just seem to hang when either node got real busy. With the block size set to 4096 we have not had any problems. In addition an account was create to do the backups with its BYTLM set to 20480, this lets DECnet work more efficiently. With SYSGEN we adjusted the number of free LRP's, before the backup starts, up to about 12 on the receiving node.

The NETIN fdl file for CONVERT's input from SYSSNET: is as follows:

```

IDENT    12-JUL-1984 15:16:24    VAX-11 FDL Editor
SYSTEM

```

How to Backup to Magtape over DECnet

How to Backup to Magtape over DECnet

SOURCE VAX/VMS
FILE ORGANIZATION sequential
RECORD CARRIAGE_CONTROL none
FORMAT variable
SIZE 4096

and the MTA0OUT fdl file for CONVERT to write to tape is

IDENT 11-OCT-1984 09:06:52 VAX-11 FDL Editor

SYSTEM SOURCE VAX/VMS
FILE ORGANIZATION sequential
RECORD CARRIAGE_CONTROL none
FORMAT fixed
SIZE 4096

This has been working quite well for over a year in our shop. One of the nice features is, that convert automatically asks for the next tape. Our operator uses the following command after mounting the next tape.

```
$ REPLY/TO=<the request number> "DQA01"
```

where DQA01 is the label to be put on the second tape.

This backup tape, and the initial VMS distribution media along with a backup of SYSSYSDEVICE:[SYS0...] to RL02s, to speed up the upgrade from V3.0 to V3.6, works. I have not been able to restore the disk from the tape, back across the network. I have managed to borrow a tape drive from another system and restore the whole disk without any problems. I went through the following steps to restore the system.

- 1) Load and install the system from the distribution media.
- 2) Then using our weekly backup of [sys0...] I brought the system V3.0 to V3.6 with:

```
$ BACKUP DQA1:SYSTUFF.BCK/SAVE -  
DQA0:[*...]/NEW_VER/OWNER=ORIG
```

VAX-6

- 3) reboot
- 4) PURGE/Keep=1 DQA0:[*...]*.* ! to get rid of old V3.0 system files
- 5) BACKUP MTA0:DQA0BCK.DAT/SAVE DQA0:[*...]/OWNER=ORIG

Ignore the error message about the system files that already exist or use the /SELECT with what ever directories you want to put in first. Another alternative, if one could find enough disk space, is to copy the saveset from tape to disk, then backup across the net from disk to disk. Or if one is not gifted with disk space(is anyone?), with liberal use of the select clause one could restore things in chunks.

Now if it was only possible to pass and retrieve optional data at the DCL level, I would only need to have one NETTAPE account and I could pass the label as optional data. Until that happens I copy a new login.com file, like the above, before I do my remote access.

Plugging a Security Leak in CERBERUS

Johan Hamaker
Netherlands Foundation for Radio Astronomy
P.O. Box 2, 7990 AA Dwingeloo
The Netherlands
telex 42043 srzm nl
telephone: 05219-7244

From the Editor's little notes in recent Pageswappers I gather that my description of the CERBERUS package for executing command procedures with temporary privileges (Pageswapper, January 1985) has roused some interest. I hope that this note comes in time to prevent security disasters from happening on sites where CERBERUS is being adopted.

One of the crucial elements in the package is that it controls all possible exits from the privileged process state, insuring that the user cannot regain control with more than his regular authorised privileges.

I recently discovered that, in the form in which I proposed it, CERBERUS gives its users access to a very easy uncontrolled exit. Indeed, before attempting to execute any command line, DCL tries to substitute the first word (the "command verb") in the line by looking for a symbol with that name. Thus, the user can override normal execution of any verb by redefining it, e.g.

VAX-7

```
IF="STOP!"
```

(The exclamation mark will cause the remainder of the line to be treated as comment.) With this definition, the first IF statement in the privileged procedure will cause an immediate return to DCL command level, without giving CERBERUS a chance to retract the temporary privileges.

The solution is simple: CERBERUS or the privileged procedures themselves must override any redefinitions which the user may have made by their own, e.g.

```
IF="IF"
```

(Note that the definition must be LOCAL so it will take precedence over all previous ones!)

To make this method work also for those cases where a privileged procedure might contain abbreviated commands, one should allow for all possible unambiguous forms of a command, e.g.

```
PU*RGE="PURGE"
```

Although the number of DCL verbs is considerable and is likely to grow in the future, the extra overhead of redefining them all was found to be unnoticeable to CERBERUS's users.

Like DCL symbols, logical names could also be abused, e.g.. to access files other than those on which a privileged procedure was designed to operate. Either CERBERUS or the privileged procedure itself should take the necessary precautions.

The Bedford Store Is No More

by Larry Kilgallen, Pageswapper Editor

A great advantage of being in the Boston area (and somewhere in California as well I am told) was the existence of a DEC Bookstore. Many manuals were in stock, and they could special-order others. I placed an order there for my VMS V4 documentation set and on the day they first came in (the Bookstore staff notified me) I learned I was not the only one, as the place was overflowing with V4 doc sets.

There is a great satisfaction in being able to walk in, buy something, and carry it out (even if it takes up three boxes). I contrast this particularly with the non-service from DECdirect who on my last encounter spent 18 months to NOT send me the update I had ordered. I sent letters every three months and struggled each Symposium to find someone who would own up to representing DECdirect. There are always plenty of technical writers to discuss the contents of the manuals, but never anyone to talk about the difficulty of getting them.

Early in July I was quite disappointed to stop by the DEC Bookstore and discover it did not exist anymore. There are Business Product Centers (or is that IBM's name for the same thing?) but the Boston instantiation seems to deal exclusively in word processing and booklets on how to conquer fear of computers. Nothing for VAX fans.

I understand that DEC makes changes now and then, but it is quite a disappointment to have them cancel the part that works. So I look for alternatives. A couple of months ago I signed up on speculation with the "Electronic Store", but didn't really put a lot of work into using it (almost as painful as the machine run by TSC) until the DEC Bookstore was no longer there.

Trying to get a VAX Pascal documentation set (greatly improved for Version 3.0, in case you hadn't heard) I found "you can't get there from here". Calling their help line (would you believe it, the person who answered the phone was the one who knew the answer!!!), I learned that taking the "purchase" option when rummaging around in the "VAX Pascal" section does not entitle you to order the full set of kits listed in that section. Instead, you must go back up to the main menu and choose an express purchase option. (Sigh)...

So I was able to order Pascal documentation, rather than walk in and buy it. It hasn't arrived yet, so tune in next month for the next installment of "Convincing DEC to Sell You Documentation".

More on VMS Timekeeping

Science and Engineering Research
Council
Rutherford Appleton Laboratory
Space and Astrophysics Division
Chilton, DIDCOT, Oxon
Oxfordshire, OX11 0QX
England
Tel Abingdon (0235) 21900 Ext 5372
Direct Line (0235)
Telex 83159 RUTHLB G
Fax (0235) 832277

10 May 1985

Mr Larry Kilgallen
Pageswapper Editor
Box 81, MIT Station
Cambridge, MA 02139-0901
USA

Dear Mr Kilgallen

Don Golden's letter in the December 1984 Pageswapper has prompted me to SPR a solution to the 'time zone' problem. I fully agree with his analysis, and my proposal differs from his only in detail.

First, the 'time zone' parameter must not be an integer as some time zones differ from UTC (Universal Time, the successor in civil timekeeping to GMT) by n.5 hours (Singapore for example).

Second, the best flavour of time to use as the standard is, in my opinion, International Atomic Time (TAI). Unlike UTC, which is tweaked every so often to fudge it back roughly into line with the Earth, TAI does not have leap seconds which might conceivably cause problems in a real time system. TAI currently differs from UTC by (exactly) 22 seconds, and jumps by one second only every couple of years; so most sites can pretend standard time is 'GMT'. Really punctilious sites can include the TAI-UTC in the time zone parameter.

Yours sincerely
Patrick Wallace, Starlink Project Manager

VAX-10

VAX-11 RSX Device Name Performance

David R. Butenhof, VAX-11 RSX Project Leader
DEC, ZK01-3/H21
110 Spit Brook Road
Nashua, NH 03062-2642

July 1, 1985

A B S T R A C T

The purpose of this article is to explain the need for "\$\$n" names, and why the implementation of \$\$n names in VAX-11 RSX Version 1.0 resulted in a severe performance degradation. This article will also explain how the VAX-11 RSX development group plans to correct these problems for VAX-11 RSX Version 2.0, and gives estimates of how much the performance will improve.

Why things had to change

Before VAX/VMS Version 3.0, a VAX/VMS device name had the format ddn: where dd represented the device type ("DR" for Massbus disks, "DM" for RK06 or RK07, etc.), a represented a controller letter, and n represented a decimal unit number, technically in the range of 0 to 32767, but usually less than 16 (with a few exceptions such as remote terminals and mailboxes).

This format is not much different from the traditional RSX-11 device name, which has the format ddn: where dd represents the device type (almost as VAX/VMS device names, with a few exceptions like "MM", which VAX/VMS refers to as "MT"), and n represents a unit number in the range of 0 to 377 (octal).

All that was necessary for the RSX-11 Application Migration Executive, (or "AME", which was then a bundled component of VAX/VMS), to translate a VAX/VMS device name into an RSX-11 device name was add the controller letter. There is no RSX-11 analogue for a controller letter. This was done by treating the controller letter as a 16 unit offset (as if "DRB0:" were actually "DRA16:"). The actual VAX/VMS unit number could then

VAX-11

be added to this value, and the result used as the RSX-11 unit number, unambiguously. If the aggregate unit number ((controller * 16) + unit) exceeded 377 (octal), the device name could not be translated, but this condition rarely occurred. In fact, the code did not even consider the possibility, and would simply use the low byte of the resulting value.

The RSX-11 device name could be converted to a VAX/VMS device name by simply reversing the process. The unit number was divided by 16, the integer result added to the ASCII character "A" became the VAX/VMS controller letter, and the remainder became the VAX/VMS unit number. Several special cases were necessary for devices like remote terminals and mailboxes that always have "A" controllers regardless of the magnitude of the unit number.

When the concept of "rooted directories" was introduced in VAX/VMS Version 3.0, the conversion algorithm no longer worked well. To treat rooted directories so that RSX-11 applications would function similarly to native mode VAX/VMS applications, it was necessary for the RSX-11 AME to treat rooted directories as "virtual devices." This meant that the root directory name could not be separated from the device name itself, nor could the root directory name "DRA0:[SYS0.]" be equivalent to the plain device name "DRA0:". Obviously, the rooted directory name could not be mapped into the RSX-11 physical device name format. Rather, the RSX-11 AME used the VAX/VMS physical device name, but maintained internal context, including the full rooted directory name and the directory's file identification value. Therefore, while the RSX-11 application might see the device "DR0:", the AME used the full root directory context "DRA0:[SYS0.]"

Unfortunately this convention was ambiguous. When a task assigned a LUN to "DR0:", the RSX-11 AME had no way to determine whether the task actually intended to use the rooted directory "DRA0:[SYS0.]", or the physical device "DRA0:". Therefore, a root directory could be associated with an RSX-11 device name only through the initial parsing operation. A subsequent assignment of another LUN to the same device would not propagate the root directory context.

For example, if 'LB = "DRA0:[SYS0.]"', then assigning a LUN to "LB0:" would preserve the root directory context, and accessing a file via that LUN would result in the correct "virtual MFD" directory context. But a GLUN\$ on that LUN would return a device name of "DR0:", and a subsequent LUN assignment, using that device name, would result in assignment to the physical device "DRA0:" with no root directory context. If this were not the case, it would have been impossible for a task to access both "LB:" and "DRA0:". The VAX/VMS Version 3.0 RSX-11 AME support of rooted directories was at best a temporary and barely satisfactory workaround.

With the advent of VAXclusters under VAX/VMS Version 4.0, this workaround was impractical. Even aside from the special case of rooted directories, VAX/VMS physical device names such as "HSC001\$DUK1000:" cannot be mapped to RSX-11 device name format. Furthermore, it is quite possible that another device called "HSC002\$DUK1000:" could exist within the same VAXcluster. Not only is the aggregate unit number of each of these names far beyond 377 (octal), but the old-style two character device name "DU" and the unit number would be insufficient to uniquely describe the target device in any case. The full physical device name is absolutely essential.

This, combined with the problem of rooted directories, and the original problems with mailboxes and remote terminals, mandated that VAX-11 RSX (which was being unbundled from VAX/VMS during the development of VAX/VMS Version 4.0), finally remove the direct connection between VAX/VMS physical device names and "emulated" RSX-11 device names.

The \$\$n connection

The problem was to reduce the full VAX/VMS physical device name and rooted directory to something which an RSX-11 task could perceive as a physical device name, and which the VAX-11 RSX AME could uniquely and bi-directionally connect with the original VAX/VMS name.

A number of proposals were considered which would minimize the impact on VAX-11 RSX users. Most of these proposals had severe drawbacks, including:

- o Preventing RSX-11 tasks from using VAX/VMS logical names such as SYS\$SYSTEM, since they do not have an RSX-11 device name format.
- o Requiring all VAX/VMS logical names definitions to be confusingly re-duplicated (or modified) so that the logical name closest to the physical name (with the "concealed" translation attribute) would resemble an RSX-11 device name.
- o Combining both problems, and adding to user confusion, by using any logical name in the translation path which resembled an RSX-11 device name.

We decided that it was important that the design we chose fulfilled the following criteria:

- o Could be documented clearly and simply (with the realization that any implementation which solved the problems would appear confusing compared to the previous behavior),
- o Was consistent and predictable,
- o Did not exclude or unnecessarily complicate the use of standard VAX/VMS logical names already defined for native mode use.

The final decision was to implement a new RSX-11 "physical device name," something currently unassigned to any device. We chose "\$\$." All VAX/VMS devices and root directories on the system would appear to RSX-11 tasks as units of the device "\$\$". These devices would be described to the VAX-11 RSX AME through VAX/VMS logical names, which it would translate when presented with a physical device name. This was relatively easy to document (although still somewhat confusing simply because it diverged so radically from the previous state of affairs); consistent and predictable because a simple "SHOW LOGICAL \$\$*" would show all \$\$n name translations; and would not exclude the use of ordinary VAX/VMS logical names, since the VAX-11 RSX AME would associate the VAX/VMS physical device name and root directory with a \$\$n name totally independent of whichever VAX/VMS logical names may have been translated to arrive at that device name.

VAX-11 RSX Version 1.0 implementation

The VAX-11 RSX Version 1.0 development schedule was constrained by the VAX/VMS Version 4.0 development schedule. Though we were technically under the management structure of the RSX-11 group, in fact we were still working for VAX/VMS. It was necessary that VAX-11 RSX Version 1.0 be available to existing customers when VAX/VMS Version 4.0 went to field test, and because of this, we did not have the freedom to take development risks.

This constraint meant that \$\$n names had to be implemented within the current structure of the VAX-11 RSX AME, without the major redesign effort actually required for the feature. The AME code implementing the ALUN\$ and ELP\$ (extended VAX/VMS parse operation) directives, the modules most strongly affected by the implementation of \$\$n names, were already unstructured and inefficient. Adding the new features without totally redesigning the code meant dramatically increasing the inefficiency, but we had to accept that in order to insure that a functionally correct AME would be available when VAX/VMS

Version 4.0 was ready for field test.

The RSX-11 physical device name (two ASCII characters and a one-byte binary unit number) supplied to the ALUN\$ directive was first converted to an ASCII string in the RSX-11 ddn: format, with n in octal. The AME then attempted to translate the result as a logical name. If there was no such logical name, it would then reformat the device name as a VAX/VMS device name in the ddn: format, with n in decimal. It would attempt to translate the resultant name, and repeat this until it reached the physical device level. If it found a translation string which included a rooted directory name, it would retain the logical name as the "VAX/VMS physical device name" for the LUN.

Once the AME had a VAX/VMS physical device name, it needed to calculate the equivalent RSX-11 device name, which could be returned to the RSX-11 task on a GLUN\$ directive (or the new GDVI\$ directive). The AME calculated the equivalent by scanning a list of pre-defined logical names, including LB, SP, and the \$\$n names, until it found one that translated to the same VAX/VMS physical device name (and root directory, if any). For each of these, the logical name needed to be translated to the level of a physical device name and root directory. The device name had to be separated and resolved to a physical device name using \$GETDVI to remove formatting variations (so that a \$\$n name defined to "DR:" would match against "DRA0:", for example). The physical device names were compared; if they matched, and if root directories were present in both strings, they were also compared.

This process was obviously slow, and often required large numbers of logical name translations during the initialization of average RSX-11 tasks.

VAX-11 RSX Version 2.0 implementation

The VAX-11 RSX development group was not constrained by time and schedule for Version 2.0, so we decided to reimplement \$\$n names correctly.

The code implementing ALUN\$ has been almost totally redesigned and reimplemented. The code for ELP\$ has been drastically modified. Many subroutines used by these routines have been totally redesigned, and the associated data structures have also been redesigned. This reconstruction not only provides improved performance, but also enhances capability: VAX-11 RSX Version 2.0 supports (limited) use of VAX/VMS search list logical names from compatibility mode.

VAX-11 RSX Device Name Performance

The redesign of the code and data structures requires fewer translations to be done when initially processing a device name. In most cases, no translations (or other VAX/VMS system services) are necessary when subsequently operating on that same device, which was not possible for VAX-11 RSX Version 1.0.

Furthermore, when LB, SP, or a \$\$n name is translated the first time, the value is cached so that the same name need not be translated again. This cache can be disabled (via the logical name RSX\$DEVICECACHE) if a site desires the ability to change (or add) \$\$n names while an RSX-11 task is running (note that dynamic \$\$n names -- with unit numbers of 350 (octal) and above -- are never cached). Data structures which are re-used frequently are kept on "lookaside" lists rather than being deallocated when not in use, reducing the overhead of reusing them.

The combined result of these changes is that VAX-11 RSX Version 2.0 is, for most RSX-11 tasks, between 2 and 3 times faster than VAX-11 RSX Version 1.0, even with the additional overhead and complication of supporting search lists. In extreme cases(1), VAX-11 RSX Version 2.0 is up to 54 times as fast as VAX-11 RSX Version 1.0.

The numbers

The performance of VAX-11 RSX Version 1.0 and Version 2.0 were measured in terms of pagefaults, CPU time, and elapsed (clock) time. The tests were run on a moderately busy VAX 11/750. The system is configured as a "common system disk" (although the processor is not in a VAXcluster), and the VAX-11 RSX Version 1.0 tests used 'LB = "SYS\$COMMON:"' while the VAX-11 RSX Version 2.0 tests used 'LB = "SYS\$SYSROOT:"'. This results in biased figures (the search list reduces performance slightly), but is more representative of how the respective versions would actually be used. Performance measurements were done using the field test version of VAX-11 RSX Version 2.0.

Five test sets were compared:

1. VAX-11 RSX Version 1.0
2. VAX-11 RSX Version 2.0 with \$\$n name caching and no unit number limits(2).

1 A test designed to simulate worst-case ALUN\$ performance, by alternately reassigning a single LUN between two devices mapped by "\$\$346" and "\$\$347", 100 times.

VAX-11 RSX Device Name Performance

3. VAX-11 RSX Version 2.0 with \$\$n name caching, and unit number limits of 10 (octal) and 340 (octal) (the fastest case measured).
4. VAX-11 RSX Version 2.0 with \$\$n name caching disabled, and no unit number limits (the slowest case for VAX-11 RSX Version 2.0).
5. VAX-11 RSX Version 2.0 with \$\$n name caching disabled, and unit number limits of 10 (octal) and 340 (octal).

Seven (7) test cases were used in each set. Each test set was repeated 20 times (to minimize the impact of fluctuating system usage). The reported values are the average of the 20 repetitions.

1. "MCR MAC RPR,RPR/-SP=RPR", assemble a small Macro-11 program.
2. "MCR TKB RPR,RPR/-SP=RPR", task build the same task.
3. "MCR F77 REQUES,REQUES/-SP/LI:7/CO:10/TR:ALL=REQUES", compile a small FORTRAN-77 program.
4. "MCR LBR LB:[1,1]SYSLIB.OLB/LI", type a listing of the contents of SYSLIB.OLB.
5. "MCR PIP A.B=*.*/LI", write a listing of the current directory to a file (the current directory had a large number of files).
6. "MCR PIP A.B/PU", purge 3 obsolete versions of a file.
7. "RUN ALUN", run the ALUN test mentioned previously. It assigns a LUN to "\$\$346:" and then to "\$\$347:", repeating this 100 times; this test forces maximum turnover of internal data structures, and therefore creates the most work for the AME. Because this task does little other than ALUN\$ directive calls, this test also serves as a rough measure of the actual ALUN\$ performance, as opposed to the other tests which provide the "real world" effect on tasks doing useful work.

On the average, with default caching enabled (no \$\$n unit number limits in effect), VAX-11 RSX Version 2.0 is about 2.41 times as "fast" (measured in terms of elapsed CPU time) as VAX-11 RSX

2 This is controlled by using RSX\$DEVICECACHE; see the VAX-11 RSX Version 2.0 Compatibility Mode Reference Manual for information on using this logical name.

Version 1.0. The actual test values range from a low of 1.14 for the "MCR PIP A.B=*.*/LI" command to a high of 3.79 for the "MCR TKB RPR,RPR/-SP=RPR" command, not including the "RUN ALUN" test, where Version 2.0 was 45.26 times as fast as Version 1.0. On the average, Version 2.0 ran with 1.02 times fewer page faults and used 2.79 times less elapsed clock time.

The following summarizes the average improvement of VAX-11 RSX Version 2.0 over VAX-11 RSX Version 1.0 for each of the Version 2.0 test sets mentioned above:

- o \$\$n cache, no limits: page fault improvement 1.02, CPU time improvement 2.41, clock time improvement 2.76
- o \$\$n cache, limits of 10 and 340: page fault improvement 1.03, CPU time improvement 2.62, clock time improvement 3.07
- o no \$\$n cache, no limits: page fault improvement 1.03, CPU time improvement 1.97, clock time improvement 2.00
- o no \$\$n cache, limits of 10 and 340: page fault improvement 1.05, CPU time improvement 2.35, clock time improvement 2.72

In a few simple tests, it appears that the VAX/VMS Version 3.7 AME runs as fast as VAX-11 RSX Version 2.0 (with default caching) or slightly faster. However, fair testing is not possible as the VAX/VMS Version 3.7 AME cannot be made to run under VAX/VMS Version 4.0. It is inevitable that the current and future versions of the VAX-11 RSX AME will be slower than the original, because more needs to be done to emulate the RSX-11 environment under VAX/VMS.

We have dramatically improved the performance of VAX-11 RSX Version 2.0 as compared to Version 1.0. We will continue to try to provide the best practical performance, while maintaining compatibility with VAX/VMS and improving our compatibility with RSX-11 in future versions of VAX-11 RSX.

We appreciate your patience through the difficult adjustment period as we unbundled VAX-11 RSX from VAX/VMS, while maintaining and extending the RSX-11 environment under VAX/VMS Version 4.0. This article has been an attempt to help you to understand why some of the changes have occurred.

A Proposal on RMS

Jeffrey S. Jalbert
J C C
(614)587-0157

This note is the result of a question asked at the "Digital Asks the DEC Users" session that was held on Monday at the New Orleans Symposium. It contains several comments, a suggestion, and an offer.

The symposium question had to do with the desirability of a specific additional feature for RMS. Why, I asked myself, shouldn't I vocalize some of those features I would like to see in RMS. This led to a further thought, why not solicit, compile, and publish a focused set of requirements from the entire VMS community? This letter is born of those thoughts.

Features of interest to me are:

1. Multi-datatype segmented index

Often we have a need to design an index which is comprised of several fields. This can be accomplished using the segmented string keys available with both prologue 2 and prologue 3 files. It is not convenient, however, to keep all such fields as string datatypes, especially when such fields are used in calculations and our programming language is FORTRAN or BASIC.

Why be stingy? We would like to see any valid VMS datatype as a segment in such an index.

2. Read backwards

The sequential nature of indexed sequential files is often useful to "locate" oneself in a file. One randomly reads a record and then reads successive records sequentially (by index.) This process is called "dynamic" access. We have as frequent a need for dynamic access both in the forward as well as BACKWARD directions. Forward alone is not enough.

If we cannot have both forward and backward, we would like to be able to define an index maintained in reverse sorting order. We understand that this may require the definition of additional indices.

We would also like to read sequential and relative files backwards. Often, the most recent data is what we are interested in.

3. Speed

Of course, we all want speed improvements for random ISAM retrieval and insertions, but what will we give up? For us the answer is "sequential". Random yes, sequential no. I believe "the other" computer company calls this type of indexed but non-sequential file organization "HDAM", hashed direct access method. Such an organization should allow RMS to avoid the bucket split issues that plague the current ISAM organization.

We are in the habit of creating non-unique indices though. We cannot afford to have records float too much. Thus, we would like to see the RFA's of the records remain fixed (and as permanent as today) in such files. We can thus process HDAM files with techniques we use today.

HDAM files should be able to support multiple indices as well as the new index features that are noted above.

Lets all recognize the dramatic decrease in the cost of disk storage, by a factor of 100 in the past ten years. We would also be prepared to pay a bit in file size too. (Hashing techniques work best when there are more record slots available than records to go in them!)

Of course we will read HDAM files sequentially. When such files are read sequentially we expect to receive a record stream in no particular order. We are prepared to accept such strange orders as long as RFA's will remain fixed. We are also prepared to receive records in a non-dependable order for sequential reads.

4. Dynamic indices

We do a lot of application design and often find a need to add indices to existing files. We have obtained significant performance improvements when doing so. However there is a problem to this -- what do we do if the file is 200,000 blocks and we have only 50,000 blocks free? That problem is real and pressing.

Today we spend hours tossing files onto a backup tape, freeing space, reorganizing, and then restoring the archived files. It seems to me that this is unnecessary. We would like dynamically to add and delete indices for an indexed file.

5. In-place reorganization

This issue goes along with the one mentioned above. When dealing with systems that have limited disk storage, it would be extremely useful to reorganize files in place. Sure we save the backup tapes, but it may be that the tape drive is located on another system that is accessible only via slow-speed DECnet.

If you would like to contribute your entries to this list or comment on mine, please send a note to:

Jeffrey S. Jalbert
J C C
P.O. Box 381
Granville, Ohio 43023

I will compile all comments and additional suggestions and report them back to you (and VMS engineering) in the January issue of the Pageswapper. If this offer prompts an extensive comment from anyone, TU58's and 9-track 1600 bpi tape will assist in inclusion of those comments within the promised report.

More on a Comprehensive Backup SIR

Lawrence J. Chiano, MIS Manager
Sippican Consultants International, Incorporated
Computer Group/SCI
1033 Massachusetts Avenue
Cambridge, MA 02138
(617) 868-1200 ext 460

Regarding the June 1985 Pageswapper article by Andy Davenport of Harvey Mudd College, I would like to add the following suggestion.

Allow the /SELECT= and /EXCLUDE= qualifiers to apply temporary filename field defaults, as do other DCL commands which use filenames for input.

For example: If I want to restore the list of files, [MYDIR]A.B, [MYDIR]C.D, [MYDIR]E.F, I need to use the following BACKUP command:

```
BACKUP/SELECT=([MYDIR]A.B,[MYDIR]C.D,[MYDIR]E.F)
```

I would like to be able to use:

```
BACKUP/SELECT=([MYDIR]A.B,C.D,E.F)
```

Currently the shorter version of the BACKUP command will select files C.D and E.F from any and all directories that are on the BACKUP save set. this is contrary to other DCL commands such as COPY, PRINT, DELETE and DIRECTORY.

LeLisp: A LISP Environment for VAX/VMS

Lee C. Rice
Marquette University
Department of Philosophy
Milwaukee, WI 53233

The Marquette user community has had an active group of students and researchers working in Lisp and artificial intelligence for several years. Some early work was done in Franz Lisp (running in UNIX compatibility mode on our cluster of VAX-11/785s) and, beginning in 1983, in InterLISP. Both of these Lisp environments, like the VAX LISP announced by DEC early in 1985, place enormous demands upon their host systems. In a large multiuser environment such as Marquette, even InterLISP running under native VMS produced a marked degeneration in response time.

In 1984 we became aware of LeLisp, a system promoted as highly efficient and extremely portable. The system was created in France by the INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE (Domaine de Voluceau, B.P. 105, 78153 Le Chesnay, France), and the VAX/VMS version is currently separately supported by the ECOLE NATIONALE SUPERIEURE DE TELECOMMUNICATIONS (46 rue Barrault, 75634 Paris Cedex 13, France). In addition to VAX/VMS, UNIX, and six other machines, the system is currently under development for still other operating systems and hardware.

LeLisp is wholly written in the virtual machine language LLM3, which was implemented by INRIA as a means of efficiently moving the language into new environments. VLISP was chosen as its base language, since this has proven more efficient than either Franz Lisp or InterLISP. The initial design of the language included compiler and debugging support, and was undertaken with a view to user extensibility.

Benchmarks of a variety of programs subsequently published by INRIA indicate that LeLisp is approximately 50% more efficient than Franz Lisp, InterLISP, or even the LISP Machine (Symbolic-3600). Our experience with it at Marquette has been highly favorable, and we have experienced no problems in response time degradation even during peak periods, with as many as 30 users on a single VAX working in LeLisp.

THE LELISP ENVIRONMENT

Entry into the interpreter loads a default core image containing about 2000 precompiled Lisp functions. While it is possible to save and reload entire user-created core images for the extensive development of large applications, the default image satisfies most user needs. LeLisp supports four types of functions: EXPRS (which evaluate their argument lists before execution), FEXPRS (no evaluation), MACROS, and a special macro form called the DMACRO. The LEXPR functional type of Common Lisp can be defined in LeLisp as a special form of EXPR.

Predefined functions include all of the standard MAP and EVAL functions of Common Lisp, with extensions of many of these to the VECTOR, which LeLisp recognizes as a distinct object type. LeLisp is also particularly strong in control functions, which include not only the classical COND, iteratives (PROG, PROGL, PROGN), but also IF, WHEN, UNLESS, WHILE, SELECTQ, REPEAT, and FOR. As well as a rich array of recursive control function, LeLisp also provides the possibility of iterative definitions, which makes it an ideal environment for learning by those more experienced with nonrecursive structures. Recursions are also optimized both by the interpreter and compiler, which minimize stack size and have special forms for handling tail recursions. An infinite table of integer squares, for example, can be printed with no stack overflow.

The Lisp interpreter recognizes both integers and floating point numbers, and also has functions for generic arithmetic which provide automatic conversion. Three additional computational packages are also provided and loadable from user libraries. The first of these, extended integer arithmetic, provides integers of up to 65535 digits. The second is a floating point library which allows calculation to arbitrary precision. Finally, new with Version 15 of LeLisp is a program for rational arithmetic, Q-Numbers of the form (x/y), which provides standard rational arithmetic functions without roundoff errors, as well as a complete package of functions for converting from and to floating point equivalents.

Because all symbol names in LeLisp are packaged, name collisions and local parameter capturing are avoided by both the interpreter and compiler. Several packages are currently provided in the user libraries, and individual users may create their own interned and auto-loadable packages easily.

LeLisp provides complete buffered I/O mechanisms on both terminal and text files, as well as an independent raw output mechanism for the terminal. Buffered I/O may be controlled by default software interrupts, but these may be fully controlled by the user. Error handling is provided by devices similar to

those found in MacLisp, by a mechanism for exceptions and local exits which is also under complete user control.

The LeLisp compiler is fully compatible with all interpreted code, and operates in two passes. It provides a runtime execution improvement at the order of about 10 times, and is quite small, so that compilation time is also typically quite short. The first pass produces LLM3 code in LAP form, which is optimized and converted to binary code by a machine dependent loader on the second pass.

Full trace, debug, and autoload facilities are also provided within the interpreter. These enable the production of personally customized environments for developing and testing programs. Three editors are also provided. PEPE is a line oriented editor which understands Lisp code and can evaluate functions being edited. EMACS is a full screen editor with extensive customizing capabilities. BIGMACS, still under development, includes a text editor, hierarchy editor, and a tree editor for the definition and manipulation of new Lisp data structures.

Version 15 of LeLisp also provides full integration with VMS by enabling the user to execute most DCL commands from within Lisp via the bang command (e.g., !EDT, !MAIL, !DIR). This holds the Lisp user environment in suspension pending execution of the command, and then returns the user to Lisp with all global variables intact.

In addition to the user libraries which they provide and support, INRIA and ENST also distribute libraries of utilities contributed by users. Their number has grown steadily over the past year. There are presently over 200 European sites for LeLisp.

DOCUMENTATION AND SUPPORT

When we received the LeLisp system in 1984, all available documentation was in French. We were particularly impressed with the LeLisp Manual. It is clear, well organized, and contains over 2000 examples of Lisp functions. The other Lisp manuals in our experience were poorly organized, lacked adequate examples, and badly written. I undertook the translation of the LeLisp Manual at once, and tested in as a general introductory text in a course offered at Marquette during the spring semester of 1985. The complete LeLisp Version 15 Manual was completed in June of 1985, and is approximately 250 pages in length.

Another support library included with LeLisp also interested Marquette users. CEYX is a Lisp extension which enables creation and manipulation of new data structures and objects. Objects are defined as the combination of a record-like structure with a set of semantical properties which are the basic operations to be performed on these structures. CEYX resembles SMALLTALK and FLAVORS in its ability to arrange objects in hierarchical structures which inherit the properties of their ancestors.

CEYX is written in CEYX, and is bootstrapped. All primitive operators are thus fully available to the user, and the system is fully extensible. Additional CEYX modules are included in the CEYX library in symbolic form, and may be modified by users for particular purposes. CEYX also includes a precompiler to optimize CEYX code prior to compilation.

French documentation for CEYX was in the form of a series of monographs, some of which deal with particular library utilities. My approach to English documentation for CEYX was to include these monographs (in translated form) in a single larger volume which contained a general introduction to CEYX structures as well. This manual, AN ENGLISH LANGUAGE INTRODUCTION TO CEYX, is approximately 160 pages, and contains source code for many of the library utilities.

The English documentation for LeLisp and CEYX is provided at no charge in machine readable form (RUNOFF format) by ENST to licensed sites for LeLisp. Also included with this documentation is a set of user utilities contributed by Marquette users, and fully commented. Educational institutions may also profit from a licensing fee (approximately \$100) which has been made low enough to encourage the growth of English-speaking LeLisp sites. The licensing form, together with further information for the VMS version of LeLisp, may be secured by writing directly to:

Michel DANA, Ingenieur
Ecole Nationale Supérieure de Telecommunications
46, rue Barrault
75634 Paris Cdex 13
France

Those interested in the UNIX or other versions of LeLisp should write directly to INRIA, whose address is given earlier. Several U.S. institutions have already licensed this new and powerful dialect of Lisp, and I would appreciate new licensees' sending me their address as well. We hope in the future to form a special interest group of LeLisp users to share resources and developing applications.

"It's in the Microfiche..."

Internals Working Group
B.C. Dow-Pleines
MAGIC ONE
5915 Southwind Drive
San Jose, CA 95138-1847

XDELTA - An example of changing the Program Counter

My favorite utility on VMS is a little known program called XDELTA. It is useful in the debugging of a device driver and code that runs at a high IPL (Interrupt Processor Level). XDELTA is not well known in the user community because 1). not all people are writing device drivers or operating system code, and 2). it has a marked impact on the machine when you use it. Basically no one else can be on the machine when someone else is using XDELTA. Only OPA0: can be used to issue commands and observe their results. This is why I like XDELTA so much, it turns the 780 and 785 I use into my own PC!

Some of the problems with XDELTA are the lack of documentation and training available. Some documentation can be gotten from the device driver manual and the device driver course offered by DEC, but they don't to my knowledge offer training in the use of XDELTA yet. Maybe the MicroVAX II's will change that. Most people have learned XDELTA either by trial and error, or from another user, or by listening to talks given at the DECUS Symposium. There was an excellent talk given by Jamie Hanrahan on XDELTA at the Spring 85 DECUS. The tape for this talk is probably available through DECUS or Eastern Audio. His notes from the talk might appear in the proceedings for the symposium. Additionally, I'd like to offer some help by writing this article.

Recently I had a problem where I needed to change the flow of control of a terminal driver. I had to make it execute an error path. Since I hadn't written the original code, I wasn't sure why it was failing in the first place. I, like many software people, assumed it was hardware that caused the driver to execute the error routine. After all, software never fails.

"It's in the Microfiche..."

So my objective in making the terminal driver fail was to simulate a hardware failure. But the question came up of how to do that. Do I walk up to the back of a terminal and pull the plug? Or, I could load a value into a register that would indicate a failure so that the code would branch to the failure routine. But why not just branch to the routine? It would save the extra work of finding the register or memory location that was being tested. Plus, all I wanted to verify was that the driver did indeed crash when it took this path.

I'm going to start my example at the point where XDELTA has already been loaded into the system, and the driver has breakpoints in the important routines. I won't discuss how to boot the system with XDELTA, nor how to put breakpoints in the code. Instead I'll refer you, for now, to the device driver manual. Since XDELTA was up, and the breakpoints were set, I started out by getting the driver to execute in the read interrupt routine. To do this I hit a carriage return on the terminal. That worked. The machine then stopped and displayed the XDELTA breakpoint shown below:

```
1 BRK AT 8000EB63
```

This is a system breakpoint, and it takes two steps into the code to actually get into the driver code. To make the machine take two steps into the code it is necessary to type S, then wait, then S again. A carriage return is not typed after the S. The OPA0: output for this sequence is shown below.

```
1 BRK AT 8000EB63
8000EB63/NOP      S
8000EB64/RSB     S
8015082B/MOVL   @(SP)+,R2
```

I continued stepping through the code until I found the branch point I desired. I needed the address of the branch point in order to figure out the address of the error code. It is the address of the error code that I wanted to load the program counter with it. The problem was to figure out that address. I had the option of 1). guessing, 2). trying my best to figure out how many bytes a BEQL instruction takes, or 3). finding it out in SDA. I chose the last option. But if I was going to use SDA that would mean I would have to get the system back to its running state. Remember, that when XDELTA runs it is actually sitting there waiting for input from the user. That means that no other process is running, and no other activity can happen on the system, that is a user cannot login. So this brings us to then next XDELTA command, ;P. Semicolon P is the

command issued that tells the system to continue on until the next breakpoint. I thus proceeded to hit semicolon P enough times to get the system backup to its usual running state.

Since I had stepped through the code successfully I had the address of the instructions before the branch. Now it was necessary to use SDA to locate the address of the error instructions. The technique for doing this is to run analyze/system, examine the branch address to make sure it is correct, and then to examine a number of bytes past that address looking for the error address. An example is shown below.

(examine the branch address with the instruction option)

```
$ analyze/system
SDA> examine/instruction 8015086B
AADRIVER+62B: BLEQ  AADRIVER+640
```

(examine the branch address and 10 bytes following it)

```
SDA> examine/instruction 8015086B;10
AADRIVER+62B: BLEQ  AADRIVER+640
AADRIVER+62D: BSBW  AADRIVER+69D
AADRIVER+630: BLBS  R0,AADRIVER+693
AADRIVER+633: CLRL  R1
```

The CLRL R1 instruction was the one I wanted to branch to. The address of it was 10 hex added to 8015086B, or 80150873.

```
SDA> examine/instruction 80150873
AADRIVER+633: CLRL  R1
```

Now I was ready to go back into XDELTA and change the program counter. I started XDELTA again by typing a carriage return at the device using the driver. Once again XDELTA stopped the system. This time I stepped through the code until I reached the branch statement - BEQL 10\$. This is where the fun began. I examined the program counter value by typing RF followed by a slash. After the slash, the system displays the value of the program counter. My next step was to change this value to 80150873. The procedure for doing this is to type a space following the current program counter, and then type the new value. The new value is then inserted into the program counter. To clarify this, the terminal output is shown below with comments.

```
CURRENT INSTRUCTION, AND DISPLAYING THE PROGRAM COUNTER FOR THAT
INSTRUCTION -
```

80150863/BEQL 80150880 (current instruction)
RF/80150863 (current program counter value)

CURRENT INSTRUCTION, AND CHANGING THE PROGRAM COUNTER, AND NEW
INSTRUCTION

80150863/BEQL 80150880 (current instruction)
RF/80150863 80150873 (modifying program counter value)
S (issuing the step command)
80150875/MOVL 0114(R5),R2 (executing instruction at
80150873 and halting before the
execution of the next
instruction.)

As you can see, I changed the program counter, and then had
to issue a step instruction to get the machine executing again.
Note, that the instruction pointed to by the program counter,
80150873, CLRL R1, is not printed out on the console device.
Instead, after execution of the CLRL R1 command, the instruction
following it is printed, in this case the instruction at address
80150875. And thus ended my play time on the VAX.

VAX System SIG Tape Time Bomb

by Glenn Everhart
RCA A&D Eng. MS 206-1
Route 38
Cherry Hill, NJ 08358

The VAX tape is just getting put to bed now, and I expect to
send it to Joe Bingham today for addition of a couple
directories and for indexing.

In going over the tape, I recalled the flap that occurred due to
the time bomb that stopped the Spring '84 Reminder from working
prior to the arrival of the Fall '84 tape. Well, the Fall '84
version has a timebomb too, set to make it fail starting on
August 1, 1985. We should have an official tape out the door by
then, but lots of sites won't have it and may call.

This is to let you know the new status.

I fixed the bomb and have put a patched version from Fall '84 on
the tape. It has an expiration date of August 1, 2984, by which
time I suspect VAXen will be ancient history.

Users can do the patch themselves if they can figure out how to
patch REMINDER.EXE off the fall tape. In block 14 of the tape
(or thereabouts) about 1/3 of the way down is a string
"01-AUG-19854.2" which can be changed to "01-AUG-29844.2" as I
did, and the problem of the timebomb goes away. Note that this
fix will allow any buried checksums to keep on thinking the bomb
is set still. I neither know nor care if there's a checksum;
the thing works when modified. Both a patched image and a
discussion of what and where the patch is are on the tape for
Spring '85. Meanwhile, be ready, and fix your own copies if you
use the thing.

VMS 4.2 XQP breaks previous privileged code

by Art McClinton, MITRE Corporation

Although the symptoms described in this article are related to the jnet product developed by Joiner Assoc., the problems described will be apparent in any security conscious program developed under pre VMS 4.0 atmosphere. Joiner is delivering a patch that allows their programs to continue running. Users and systems managers will however note that the error messages issued do not immediately point to XQP as the cause of the problems.

In an effort to tighten VMS security, the VMS V4.2 extended QIO processor (XQP) enforces the rule that programs must have sufficient privileges to close (as well as open) files. This new feature breaks programs that keep all image privileges turned off except at the instant that they are needed, including jnet V1.2. Though you may not yet be using VMS V4.2, you may have a

Near-V4.2 XQP from the Remote Diagnostic Center to fix certain problems in Files-11 disk handling under V4.0 or V4.1.

Symptoms included various privilege violation messages when nonprivileged users send VMSmail or files over the RSCS network. The files are created as JANSYS:linkname.RSC, but the linknameDaemon is not awoken and the queue count is not incremented.

An Electronic "Campground" For VAX Users

Gary Grebus

Anyone who has attended a DECUS Symposium knows about a campground: it's place to drop in and talk shop, read notes on the bulletin board, and look for help with problems. Some of the best information is shared in this informal setting.

Now imagine a campground that is available not just during a symposium, but 24 hours a day, every day of the year...one that you can visit electronically without leaving your home or office. Such an electronic "campground" is the VAX SIG on CompuServe's Information Service.

CompuServe Information Services is a commercial information "utility". One of the many services offered to their subscribers are SIG's (sometimes called Forums), a method for communication between individuals with similar interests. Stuart Fuller, a DECUS member in the Detroit area, arranged to start a SIG devoted to one of his interests, VAX's. Since 1983, Stu has been the SYSOP or caretaker of this useful information exchange. Currently, over 100 users a day visit the SIG, and the total SIG membership numbers over 1000.

The CompuServe VAX SIG offers a number of services. One central feature is the bulletin board. This is an easy-to-use mail system for posting messages on such topics as VMS software, VAX hardware, UNIX on VAX, and VMS Communications. Messages can be posted to the general public or to a specific SIG user. Content ranges from "Does anyone know of a SNOBOL compiler for VMS..." to "We've just found an interesting problem with DMF-32's..." to "Can anyone help me with this problem with terminal QIO's...". The bulletin board is the place to get or give help and to find out what other VAX users are doing with their systems.

Another service of the "campground" is the Data Library. This is a collection of files containing public-domain software, articles of interest to VAX users, and information about using the SIG. Files can be "downloaded" to your VAX using various protocols, and you can upload contributions to share with other users. Currently, about 4 megabytes of VAX related files are available online.

Other services include real-time online conferences using a facility much like the VMS PHONE utility, and a user list giving names and interests of other SIG users. The VAX SIG is also currently host to DEC Personal Computer activities on CompuServe. A wealth of information on DEC Rainbows and Robins is available.

The CompuServe Information Service is accessible by local phone call through TYMNET, TELENET, DATAPAC (in Canada), and CompuServe's own network. Usage rates vary based on time of day, baud rate, and the network used. CompuServe subscription "Starter Packs" are sold through most local computer dealers. The CompuServe VAX SIG is accessed through menu page PCS-16. Enter "GO PCS-16" at the menu prompt. Hope to see you online!

Building Efficient Images on VAX/VMS

Benn Schreiber, Commercial Languages and Tools
DEC, 110 Spit Brook Road (ZK02-3/K06)
Nashua, NH 03062-2698

Program Sections (PSECTs)

- o Describe a portion of memory by size and contents
- o Have a user-assigned name and a set of attributes
- o Typical psects produced by VAX language compilers
 - Read-only code
 - Read-only data
 - Read-write data
- o Language compilers define contents of psects
- o In MACRO and BLISS, psect attributes and contents are under programmer control

Important PSECT attributes

- o Relocatable (REL) and Absolute (ABS) - Relocatable psects can be stored in memory. Absolute psects are used for defining global symbols and do not occupy memory.
- o Concatenated (CON) and Overlaid (OVR) - In an overlaid psect, the psect is the size of the largest contribution, and the contribution for each module starts at the same virtual address. In a concatenated psect, the size of the psect is the sum of all the contributions to it.
- o Writeable (WRT) and Non-writeable (NOWRT) - Writeable program sections can be modified by your program. An access violation will occur if your program tries to

modify a NOWRT psect.

- o Executable (EXE) and Non-executable (NOEXE) - Has no effect on whether section contains executable code. Only used for creating image sections. There is no hardware support for execute-only.
- o Local (LCL) and GLOBAL (GBL) scope - Controls linker search extent. The search for global psects encompasses all clusters, while the search for local psects is limited to the current cluster.
- o Vector (VEC) and Non-vector (NOVEC) - Most psects are NOVEC, with the exception of message sections and protected code.
- o Attributes considered when creating a shareable image
 - Shareable (SHR) and Non-shareable (NOSHR) - Determines whether psect is shared among more than one process.
 - Position independent (PIC) and Non-position independent (NOPIC) - Specifies whether psect executes correctly regardless of placement in memory. Setting NOPIC does not affect the 'position-independence' of the resulting image.
- o In a shareable image, psects that have GBL, OVR, and SHR attributes are like FORTRAN COMMON.
- o Psects with "similar" attributes are collected together into image sections.

Image Sections

- o Image sections are a collection of psects with similar attributes.
- o After it has read all input (object and shareable image) files, the linker finds psects with "similar" attributes and gathers them into image sections (alphabetically within each image section).

- o PSECT Attributes considered
 - For executable images: WRT/NOWRT, EXE/NOEXE, VEC/NOVEC (8 different combinations of psect attributes)
 - For shareable images: WRT/NOWRT, EXE/NOEXE, VEC/NOVEC, SHR/NOSHR, and PIC/NOPIC (32 different combinations of psect attributes)
- o The order of psect attribute selection is fixed in the linker. This can be used to control the ordering of code and/or data within your image. Attribute order listed on page LINK-75 of the Linker Reference Manual.

Clusters

- o A cluster is a concept used by the linker (and only the linker)
 - Accumulate image sections
 - Each shareable image input is a separate cluster
- o Clusters are made up of a group of image sections, which are in turn made up of a collection of similar psects.
- o You control placement of object modules in the image by use of the CLUSTER option.
- o You control placement of psects in the image by use of the COLLECT option.

Copy-on-Reference (CRF) pages

- o With respect to an image file, CRF pages are those that will be process-private pages.

- o Initialized, per-process data section.
- o Occupy space in the image file.

Demand-zero pages

- o Uninitialized (default to all 0), per-process data section.
- o Occupy no space in the image file, created at runtime by VMS memory management services.

Transfer Vectors

- o Use transfer vectors to ensure upward compatibility in your shareable image.
- o Ensure transfer vector is at the very front of your shareable image.
 - Use the CLUSTER link option and specify the transfer vector as the first cluster.

```
$ LINK MYIMAGE, SYSSINPUT/OPTION
CLUSTER=TRANSFER_VECTOR,,,OBJ:TRANSFER_VECTOR
CLUSTER=THE_REST,,,OBJ:MYLIB/LIBRARY...
```

- Very careful placement via psect attributes and psect name - make the attributes NOWRT, NOEXE, SHR, NOPIC, NOVEC and use a name that collates before any of your other psects (\$\$\$TRANSFER for instance).

```
$ LINK MYIMAGE, SYSSINPUT/OPTION
PSECT=$$$TRANSFER,NOWRT,NOEXE,SHR,NOPIC,NOVEC
OBJ:MYLIB/LIBRARY/INCLUDE=TRANSFER_VECTOR
```

Linker GSMATCH option

- o Use the GSMATCH option to control when images linked against your shareable image (consumers) need to relink.
- o Typically, you only change the minor ident of the GSMATCH when you change the image. This prevents images linked against the new version from accidentally running with the old version.
- o If you make incompatible changes (order of elements in the transfer vector, for instance), then you must change the GSMATCH major ident to force all consumers to relink.

Shareable images

- o A collection of routines that are known to be debugged and working.
- o Greatly reduces link time as compared to linking against object libraries.
- o Improves memory sharing.
- o Reduces software maintenance costs/headaches. "Did you relink your image to pick up the new version of the FROTZ routine?"

Shareable image libraries

- o A name space translator. The linker looks in a shareable image library for a global symbol. If it finds the symbol, it determines which module (and hence the shareable image) contains the symbol.

- o The modules in a shareable image library contain no code.
- o SYS\$LIBRARY:IMAGELIB.OLB is a shareable image library.

Image File Fixup Section

- o The linker uses the image file fixup section to guide the image activator in resolving references to shareable images.
- o Two types of image fixups done
 - Code references are fixed by directing the image activator to modify a longword in the fixup section itself. All code references to a particular global name reference the same location in the fixup section.
 - Data references (MACRO .ADDRESS) are EACH fixed up by the image activator. The fixup section contains a list of locations within the image that need to be modified. SECTIONS CONTAINING .ADDRESS DATA ARE NOT SHAREABLE.

Hints for high performance images

- o Order and group modules according to usage. In a phase-oriented image, such as a compiler, group the modules for the different phases together. This can be done by specifying the object module load order, or with the linker CLUSTER option.
- o Reduce number of image sections by combining 'similar' psects where possible. In MACRO, don't randomly use psect attributes, this can generate extra unnecessary image sections.

- o To maximize demand-zero pages and minimize CRF pages, group all initialized data together, and all uninitialized data together. This avoids the checkerboard (DZRO, CRF, DZRO, CRF) image that can result.
- o Avoid using .ADDRESS data (not possible with some language compilers). In languages where you have control, use self-relative offsets in tables rather than .ADDRESS data.
- o Use the VAX Performance and Coverage Analyzer (PCA) to help you characterize your program's performance.
- o Trade off disk space (image size) vs. activation time (number of image sections) by use of the ISD_MAX and DZRO_MIN options.
 - ISD_MAX controls the maximum number of image sections in the image (default is approximately 96). Increasing this value may decrease image size by allowing the linker to generate more demand zero sections, but may increase the total number of image sections.
 - DZRO_MIN controls the minimum size of a demand zero section (default is 5 pages). Decreasing this number may increase the number of demand zero sections, but be careful of the DZRO/CRF/DZRO/CRF checkerboard.

Global Section Problems

Pageswapper Editor's Note

The only guess I can make is that this originated with Don Golden. At any rate, he is the one who sent it to me (along with articles by others) with no indication of the author.

Dear Larry,

I feel that these two problems (currently, unsolved) warrant a letter to the editor of the Pageswapper. I don't have solutions, yet, but think other users might appreciate knowing about these pitfalls.

Our software product transfers large, and I do mean LARGE, data buffers between VAXen by using DECnet non-transparent task to task communications. It may or may not matter that the VAXen are tied together via an Ethernet link. The basic transfer algorithm is really quite simple. When the process responsible for the DECnet interface in one machine perceives the need to transfer data to the other machine, it first sends an interrupt message with some information about the message to be transferred. Included in this information packet is the desired byte count. This interrupt message is sent using a QIO (no wait). The same process then issues a QIO to write the actual message to the channel. If the message is greater than 65K, multiple QIO's are issued.

The process at the receiving end wakes up when the interrupt message arrives. {The plot thickens (:>)}. What we want to do is this. The receiving process knows the size of the message being sent along with some other application dependent characteristics, for example, the name of the real destination process. It can use this information to create and map a global section. It then hangs a read or (or a series of reads) to match the write done on the other end on the channel. If this were to work, it could tell the process on the same node about the global section and then consider its work would be done.

Not so fast, champ! This doesn't work. When the byte count approaches 4000 or so, this mechanism starts failing. Let me describe the failure syndrome. The QIOs on the sending system complete without apparent error. The QIO for the receipt of the message completes with a successful system service status. Its IOSB indicates an access violation, but shows the full byte count as having been transferred!

Closer investigation of the receive buffer indicates that fewer bytes than the full byte count actually arrived in the buffer. Even more interesting is that the actual number of bytes transferred is a strange number. It is a multiple of the DECnet buffer size less 15. That is, with a DECnet buffer size of 1024, the actual number of bytes transferred is a multiple of 1009. With a DECnet buffer size of 576, the number of bytes arriving in the receive buffer is a multiple of 561.

We did a bunch of looking at our create and map global section code, assuming we had mapped it incorrectly and had thereby caused the access violation. We couldn't see anything wrong. Still, we had to make a test. These applications programs are written in that wonderful systems implementation { (:>) }language, PASCAL. Replacing the create and map section code with some PASCAL NEW commands, we could receive into a different type of buffer.

Eureka!!! It works!!! Up to 65K bytes, our test programs do not fail, we did not worry about going beyond 65K since if you can do 65K a bunch of times one way you can do it another, right? WRONG!!!

When we modified our production programs to do NEWs instead of create and map section, everything was OK for buffers up to 65535 bytes. However, when our transfer request caused us to stack two reads and the second was more than 3 or 4 thousand, we got the same error syndrome on the second read as into the global section. This was a real stumper.

While it would be fun to bang through fiche and figure this one out, we have a production system to ship and we needed to find a work around to the work around. By reading a series of smaller chunks, namely 49152 byte chunks, the work around began to play. You'll note that 49152 is roughly 16K plus 32K so it has all sorts of nice boundary conditions.

This was all reported to DEC in an SPR which has not yet been answered. This is the most strange behavior out of DEC, since usually they are really helpful in at least giving back a status on problems like this.

The Global Section Problem

Our software product has a suite of programs which create and map page file sections. These are sections which use the page file as their backing store and are just great for temporary storage and interprocess communications. We use a directory and naming convention that allows us to generate unique names, and then to find the names later. When we complete the use of a section and delete it we return the name to the directory structure. The directory module puts the name back on the top of a linked list, where it can be used again.

This particular suite of programs tend to activate one another in the following way. Process A activates process B, gets rid of its sections, and hibernates. Process B starts up gets some sections and does its thing. Reasonably often process B gets the name that process A discarded. Looking in the global section list, there will be a section mapped to B and one of the same name in the delete pending list.

If this goes on long enough, the delete pending list gets really long and the number of global sections reaches the sysgen limit. When this happens, our software product shuts down. This is not a satisfactory situation.

The problem is, the global sections marked delete pending seem never to get really deleted. We asked DEC about this and they really couldn't or didn't give us an answer.

By returning used names to the end of our directory structure list and pulling new name off the top of the list, the entire problem vanished.

It appears that there was a problem in the global section name space, once the name got reused, the section marked delete pending could not be disposed of.

*** VOTE *** VOTE *** VOTE *** VOTE *** VOTE *** VOTE ***

System Improvement Request Ballot

FALL 1985

Everyone has an opinion about what is right or wrong with VAX. Here is your chance to influence the directions of future DEC development. The VAX Systems SIG System Improvement Request (SIR) program provides a major vehicle by which the VAX user community expresses its concerns and desires to Digital. Your opinion is important, and every ballot adds to the influence of the SIR program. Don't complain about the directions VMS takes unless you help provide the direction!

Attached you will find the current collection of System Improvement Requests and a ballot form on which to record your preferences. Please take the time to review the enclosed SIR's and assess their effect on your use of VAX's. For your convenience, the ballot items have been grouped by area of interest. New for this ballot is a section of items proposing improvements for the SIR process itself. Please give us your input on how this process can be improved.

Also, please fill out the revised questionnaire portion of the ballot. Such information helps point out requests which are important to a particular segment of the VAX community.

Please return your ballot AS SOON AS POSSIBLE. To allow time for DEC to respond, BALLOTS RECEIVED AFTER NOVEMBER 4 CANNOT BE COUNTED. The results of the voting and DEC's responses will be given at the VAX SYSTEM SIG SYSTEM IMPROVEMENT REQUESTS session of the Fall 1985 DECUS Symposium in Anaheim, and will be published in the Pageswapper.

Instructions

You have a total of 100 points to allocate among the SIR's on the ballot (excluding the section on the SIR process itself). The more points you allocate to a particular SIR, the more strongly you wish to see this feature included in the VAX or VMS. You may assign your points in either a positive sense (to encourage the change) or a negative sense (to discourage the change). In order to assure a wide range of improvements, you may not assign more than 10 points to any one SIR.

To allocate points to a SIR simply record the number of the SIR

VAX-44

in the column labeled SIR NUMBER, and the number of points to allocate to it in the column labeled POINTS. Remember you can specify only 100 points total (i.e. +70 points and -30 points uses all your 100 points). Please note that any ballot not following the points assignment rules, or not specifying a DECUS membership number will not be counted. Only one ballot per member will be accepted.

VMS Internals

SIR: F85-1

Abstract: Provide a means to keep control of a drive obtained via assisted MOUNT.

Description: Operator-assisted MOUNT provides a way to have a volume mounted on an arbitrary drive. However, there is no way to avoid relinquishing control of the drive when it must be dismounted, for example to re-INITIALIZE the volume. Without this control, it is possible that the volume could be switched and the INITIALIZE or re-MOUNT be done on the wrong volume. ALLOCATE is only effective if done before the MOUNT, which is impossible when the drive assignment is done by MOUNT.

SIR: F85-2

Abstract: Implement a quota to limit the RATE of buffered I/O's.

Description: The VMS documentation suggests how to deal with excessive terminal I/O and suggests redesigning the user programs. For the usual case where this is not practical, a quota should be available to restrict the maximum rate at which a process could generate buffered I/O's. Alternatively, it should be possible to prevent VMS from boosting the priority of a process performing buffered I/O at a rate above some system-wide threshold. Currently, a process doing excessive buffered I/O degrades the whole system and is rewarded by receiving a priority boost. Reducing the BIOLM quota does not solve the problem, since the priority boost insures troublesome process receives prompt service. Putting an upper limit on the buffered I/O rate per process would preserve the system's responsiveness at the cost of degrading the response time of the offending process.

VAX-45

SIR: F85-3

Abstract: Provide facility for parsing a privilege string.

Description: VMS should provide a common facility (as both an RTL routine and a DCL lexical function) which can convert between the textual representation of a VMS privilege list and the corresponding bit mask. This would make it easy to reliably handle privilege specifications in applications and would ease tracking future enhancements in VMS.

SIR: F85-4

Abstract: Provide more detailed CPU usage information in MONITOR.

Description: The MONITOR displays which show CPU usage should optionally provide much more detailed information about where the CPU is being used. For example, it should be possible to break down the kernel mode display of MONITOR MODES by its major components (QIO, file system, paging, etc). MONITOR PROCESS/TOP_CPU should be able to break down process CPU usage by access mode. Such additional detail is frequently needed to identify the actual performance problem from among several similar possibilities.

SIR: F85-5

ABSTRACT: Enhance the ALLOCATE command.

Description: Enhance the ALLOCATE command to enable a user to optionally queue the allocation request when all qualifying devices are busy. Device allocation should be handled by a queue manager similar to the VMS V4.0 print queue manager, and the allocation request queues should be made cluster-wide to support cluster-visible devices.

User functions should include the ability to specify characteristics required of a generic device, the automatic notification of allocation, the ability to delete an allocation request, the ability to examine the allocation request queue, and the ability to do other interactive processing while waiting for an allocation request to be granted.

Operator functions should include the ability to mark failing devices as unavailable and the ability to force a deallocate. Manager functions should include the ability to define device characteristics and specify physical devices as possessing those characteristics.

Device allocation and deallocation should place records in the accounting file so that charge back accounting can be

done for allocated devices.

A mechanism for avoiding deadlocks when multiple devices are allocated should be provided.

Examples:

```
$ ALLOCATE/QUEUED/WAIT TAPE$CLASS:-  
/CHARACTERISTICS=(DENSITY:6250) LOGICAL_TAPE
```

(Queue an allocation request for a tape drive with 6250 bpi capability and wait until the allocation has completed.)

```
$ ALLOCATE/QUEUED/NOWAIT/NOTIFY DISK$CLASS:-  
/CHARACTERISTICS=(RA60) MY_DISK_PACK
```

(Queue an allocation request for an RA60 disk drive and return control to my terminal. Notify me when the allocation has completed.)

SIR: F85-6

Abstract: Provide support for running multiple versions of a layered product.

Description: In a production environment, it is almost always an unacceptable practice to blindly replace the existing version of a layered product with a new version. Some sort of local testing or quality assurance is usually required. However, the current VMS product installation procedures do not support such an installation. Installing a "test" version of a product is a tedious and error-prone task.

The VMS layered product installation should support multiple versions of a product on the same system, and should provide an easy way to switch between versions of the product. Note this may require switching a variety of images, shareable libraries, help libraries, DCL tables, etc. In the case where the product manipulates a global database (such as the Common Data Dictionary), the installation procedures should provide for the creation of a test version of the database.

SIR: F85-7

Abstract: Provide for user-supplied filters between the terminal driver and applications.

Description: VMS should provide a facility to allow user-supplied, non-privileged programs to act as filters between the user's terminal and any application. These filters, traps, or translators could provide "one-keystroke" access to a variety of useful functions, similar to the "pop-up" utilities available on various personal computers. At minimum, such a filter might allow access to the SPAWN and ATTACH facilities of DCL. Or, it might provide access to a "notepad" file, or perform keystroke mapping for a non-standard keyboard layout. In any case, this facility would act without disturbing the previously running program.

SIR: F85-8

Abstract: Provide a SYSGEN DISCONNECT command

Description: When configuring devices with the SYSGEN CONNECT command, it would be useful to be able to reconfigure the device without rebooting the system. This would allow correcting the specification of CSR address or vector due to an error in the CONNECT or in the device controller switch settings.

This capability need not be applicable to all drivers or devices. It is not necessary that devices supported by a port/class driver be disconnectable. The idle device test used by SYSGEN RELOAD (even if not highly reliable) could still be employed with no more risk than is currently present. At a minimum, it should be possible to change the CSR and vector for a device even if the device cannot be completely DISCONNECT'ed.

This request is a repeat of an item which has previously made the top 10 SIR list.

SIR: F85-9

Abstract: Propagate the file ERASE attribute to new versions.

Description: VMS Version 4 has generally added the convention of propagating file protection attributes from previous versions of a file. Since the ERASE attribute has similar security implications, it should be propagated in the same manner.

SIR: F85-10

Abstract: Allow an image to be installed with a priority and UIC.

Description: The ability to provide a base priority increment or decrement while running a specific image would be a great help in a heavily loaded interactive system. For example, the EDT editor might be installed with higher priority to help its responsiveness, while a CPU-intensive compiler might be forced to execute at a lower priority. The ability to run with a UIC other than the owner's is a great help in cases where mailbox communication is necessary. The VMS INSTALL utility should allow both of these conditions to be specified, in the same manner as enhanced privileges.

SIR: F85-11

Abstract: Provide a means to perform an in-place compression of a disk.

Description: One of the trade-offs associated with the Files-11 disk structure is that, with time, the disk space becomes fragmented into many small extents. This causes increased inefficiencies in disk I/O, since files are less likely to be contiguously allocated and since more overhead is required to keep track of the file extents.

The solution to this problem has always required that a disk be dumped to tape (or disk) and reloaded to compress the files into contiguous allocation. However, with the advent of larger, non-removable disks, the time required to perform such a compression has become extremely large. While this may be a very difficult problem, it would be extremely beneficial to have a way to perform such a compression without the need to completely dump and reload the disk or disk volume set.

Even if a complete solution cannot be achieved, a partial implementation could be useful. Any procedure which could produce a useful amount of compression in less time than required for a dump/reload would be a great benefit.

SIR: F85-12

Abstract: Expand use of page files for swapping.

Description: In the absence of a swap file, the swapper will swap to the primary page file. The swapper should be able to use any available page file when there is no swap file. It also should be able to use more than one page file for swapping when multiple page files are available. A large cluster of machines with lots of memory and a common system disk cause too high an I/O load on the common disk. This request would provide additional flexibility in spreading the paging and swapping load to other disks.

SIR: F85-13

Abstract: Add simplified insert and extract routines to callable Librarian.

Description: It is suggested that 8 additional entry points be added to the callable Librarian to assist in the insertion/extraction of data from libraries through programs written in high level languages. These entry points would be of the form LBR\$INSERT_xxx and LBR\$EXTRACT_xxx where xxx is a valid library type (MLB, OLB, TLB, HLB). The routines would have three parameters: the library index (as used by the other Librarian routines), the name of the module to insert/extract, and a filespec where the extracted module would be written to or the inserted module read from.

SIR: F85-14

Abstract: Alter the mechanism which delivers priority boosts.

Description: The VMS scheduler gives priority boosts in response to various system events, including terminal I/O completion. Often the process has to wait a long time at base priority before the scheduler sees it and actually implements the new priority by requeueing it in the higher priority queue. It is suggested that the code which does the priority boost also should requeue the process into the higher priority queue. This suggestion is aimed at improving response for highly interactive tasks like editors.

SIR: F85-15

Abstract: TCP/IP support is needed under VMS.

Description: A large number of workstations, "intelligent" test instruments, and process control systems are currently available which use the UNIX operating system. These UNIX based systems very often support the TCP/IP network protocol. There is frequently a need to connect such devices to a VAX running VMS. VMS should support TCP/IP with file transfer capability to allow for such communications.

SIR: F85-16

Abstract: Add FILE_ID item to \$GETQUI.

Description: The \$GETQUI service should have an item which returns the File Identification for the file in a job. This information is necessary to access a file which is not entered in a directory, such as a file produced through a spooled device.

SIR: F85-17

Abstract: Support a standard print control format for printable files.

Description: VMS tools, utilities, and programs use an almost capricious variety of carriage control formats when producing human readable output files. This includes FORTRAN style carriage control, implied carriage control, and PRN style carriage control. A particularly troublesome practice is creating multiple lines of printed output by embedding ASCII control characters in a record. This is done in the listings from several compilers. This variety of formats causes considerable problems when the files must be transferred through a heterogeneous network for final printing. All DEC-supplied software should be modified to produce one standard carriage control format. At a minimum, DEC should supply a utility which can convert any printable file to a standard format.

SIR: F85-18

Abstract: Develop class-based scheduling facilities for VMS.

Description: Under VMS, a long, compute-bound batch job can prevent a lower priority batch job from getting many computes. With a class-type scheduler, one can specify that a specific class of users get at most a fixed percent of the available computes; various methods are available for handling the "leftover" portions of the processor (which exist when not all classes are on the system at the same time). Also, class-type schedulers can emulate preemptive schedulers. The system should degrade "gracefully" under load, not suddenly.

CPU time guarantees do not answer the problem entirely, since this concept really addresses the inverse of the above: the ability to specify that a given class gets AT LEAST a specified percentage of the processor. Such implementations would normally be under stricter administrative controls, to avoid over-subscribing of the available resources.

With clustered systems, a class scheduler may rapidly become a necessity. One of the major market needs is for machines to act as compute servers, so that entire machines need not be dedicated to single applications. This will

not be feasible unless resources can be partitioned in a definable and reproducible fashion.

SIR: F85-19

Abstract: Provide a service to return extended error information.

Description: Many of the condition values returned from system services are not sufficiently precise (e.g. NOPRIV, no privilege for attempted operation). However, modifying the services to return more specific codes would break many existing applications. As an alternative, VMS should provide a system service which returns extended error information about the last system service failure. This would include more specific codes (e.g. which privilege was not available).

DCL and Utilities

SIR: F85-20

Abstract: Add a keyword search capability to HELP.

Description: It would be useful to have a feature in the HELP utility which would accept a keyword and display a screen showing the keyword's occurrences in context. Thus specifying something like:

```
$ HELP/KEYWORD="/date"
```

would return information showing where HELP has entries containing the keyword "/date". The context should include the hierarchical list of topics needed to locate the reference. Frequently you may know the keywords related to a topic, but not the exact path needed to locate the information.

This facility might be generally useful apart from HELP as an enhancement to the SEARCH utility. It could allow searching of text and macro libraries with a display of the context as well as the module name.

SIR: F85-21

Abstract: Add /NOIMAGE debugging option for DCL command procedures

Description: Development and debugging of complex command procedures would be simplified in many cases if a facility similar to the RSX /NOMCR switch were available. The desired functionality would be to avoid activating images such as COPY and DELETE, etc. This would eliminate check-out tests that had undesirable side effects like deleting files based on an improperly constructed symbol substitution. The primary purpose would be for syntax checking of DCL, it would be the user's responsibility to eliminate all non-DCL entries read in by user images. Some control over which images would not be activated on a session-by-session basis would be desirable.

SIR: F85-22

Abstract: Allow command procs to be invoked from a library.

Description: DCL should support (on a process-by-process basis) a default text library to search for command procedures to be executed. This could reduce run time by minimizing directory searches and file opens. If the requested command proc were not found in the library the current default directory would be searched as is currently done. This facility would not apply to an @filespec that contained any node or directory specification. Some ability to turn this facility on and off as well as redefine the library would be needed.

SIR: F85-23

Abstract: Provide an optimizer for DCL.

Description: By applying existing compiler optimization techniques it should be possible to create a utility that converts an existing command proc into an optimized form. It would not be necessary to try to preserve the proper location and context of documentary comments. As much as possible the original symbol names should be preserved.

SIR: F85-24

Abstract: Add automatic closing capability for command procs.

Description: When a command proc is aborted via <CTRL/Y> or <CTRL/C> or any other abnormal event, some facility should be provided to automatically close all files opened. This would eliminate frustration over accidentally restarting the command proc with the old files. Note for compatibility with existing command procs that rely on the current behavior this would need to be implemented in a

user-controllable fashion. It would be desirable if this behavior could be specified as either: a default for all command procs executed, applies only to this command proc execution or applies only to specified files in this command proc execution.

SIR: F85-25

Abstract: Provide enhanced <CTRL/T> capability.

Description: In order to facility debugging of systems that utilize subprocesses it would be desirable to have a mode of behavior for <CTRL/T> or some other control sequence to perform the equivalent of a <CTRL/T> for all process and subprocesses descended from the main process. This would enable the user to ascertain if the subprocesses are working or have become hung up by monitoring the information provided.

SIR: F85-26

Abstract: Provide a "catch symbol" for the DCL interpreter.

Description: If a catch symbol facility were provided in VMS DCL that would be invoked when DCL failed to understand a command, it would be relatively easy to "catch" and parse in complex fashions "other" commands. These "other" commands would not need to follow DCL SYNTAX or be parsed by DCL but would be parsed by a user-supplied image. It would also be desirable if the user parsing image were redefinable by some method.

SIR: F85-27

Abstract: Enhanced DELETE command behavior.

Description: It would be desirable to provide some method that would allow the DELETE command to detect humanely the omission of the required ";" and behave as if "*/confirm" had been typed. This behavior would then parallel the behavior of the RSX DELETE command.

SIR: F85-28

Abstract: Modified behavior of the PURGE command.

Description: A user can purge all files in a directory by giving a PURGE command with no filespec. However a PURGE command with a filename only results in an error. By changing this behavior to treat PURGE with a filename as a PURGE filename.* and purge all file types for the given filename, behavior would be consistent.

SIR: F85-29

Abstract: Improve the behavior of the /CONFIRM qualifier.

Description: Currently, all commands that accept the /CONFIRM command expect a positive response of "Y" to cause the action to be taken, with the default behavior being "N". It would be desirable to have a user-controlled method of altering this so that the default behavior would be the requested action and thus a declination response of "N" would be necessary in order to skip performance of the requested action.

SIR: F85-30

Abstract: Modify the treatment of SET VERIFY and SET NOVERIFY by DCL.

Description: It would be desirable if DCL would automatically preserve the verify state that was in effect at each level of command proc nesting. This would eliminate annoying changes of verification state caused by the interruption or termination of a subordinate command proc. That is, propagate the VERIFY status downward through nested command procs but NEVER upwards. This would also eliminate the need for users constantly to add DCL commands to their command procs to maintain the VERIFY status of their caller and restore it on exit.

SIR: F85-31

Abstract: Provide SET PROTECTION support for logical name tables.

Description: Now that logical name tables have been made protectable objects and given protection masks, it is necessary that one be able to manipulate and examine these protections. This applies to all logical name tables: SYSTEM, USER, GROUP, JOB, etc.

SIR: F85-32

Abstract: Restore the size of the LIB\$GET_FOREIGN buffer.

Description: Currently DCL restricts the total size of the command string passed by a foreign command to 255 characters. This restriction of 255 did not apply in VMS 3.6 (some large one probably did) and the new limit of 255 is now causing working programs to break. This problem is particularly aggravated by the longer filenames and resultant filespecs supported by VMS 4.0.

SIR: F85-33

Abstract: Add a utility for setting certain file attributes.

Description: Certain file attributes describe only the interpretation of the data, not its actual structure. A utility should be available which can change such attributes to alter the interpretation of the data. The most common such change is to switch a file (perhaps obtained from a foreign tape) between FORTRAN and Carriage-Return carriage control. Another example is to switch a file among the various forms of STREAM record formats. Such a utility would eliminate the need to CONVERT large files or do QIO-level programming to accomplish this type of change.

SIR: F85-34

Abstract: Add a match limit to the SEARCH utility.

Description: Currently, SEARCH scans the entire file for EVERY occurrence of the specified search string. It would be useful to be able to limit the number of occurrences which are searched for. This would be helpful when searching a large number of files and the existence of one (or some fixed number) of occurrences of a string is sufficient to identify the file. \$ SEARCH/LIMIT=1 *.FOR "PROBOZ" would identify all FORTRAN programs containing the string PROBOZ. Once the limit of successful matches is reached SEARCH would close the file and proceed to the next one.

SIR: F85-35

Abstract: Enhance DIFFERENCE utility to recognize comment delimiters for C.

Description: When invoked with the /COMMENT_DELIMITER switch, the DIFFERENCE utility, by default, recognizes comment delimiters for many of the popular programming languages (MACRO, BLISS, FORTRAN) based on DEC-defined file types. This should be extended to recognize "C" programs and the standard /* */ pairs.

SIR: F85-36

Abstract: Enhance the MAIL utility to track delivery of messages.

Description: When using MAIL it should be possible to request complete tracking of message delivery. This would make it possible for the sender to obtain a list of who had read a message, on a by-message or by-user basis. For messages sent across DECnet, it is NOT acceptable to lose the record

VAX-56

of delivery if the originating node is unavailable at the time that the message is read.

SIR: F85-37

Abstract: MAIL should be able to send mail to all usernames.

Description: MAIL should have an option which allows a suitably privileged user to send a message to all known usernames. This would provide a convenient method of delivering site information, notices, etc. in a form familiar to the recipient. To allow finer control, the MAIL SEND command should also have an /EXCLUDE qualifier which allows a user or group of users to be excluded from the mass delivery. The /EXCLUDE list would presumably be smaller and more manageable than the list of all known usernames.

SIR: F85-38

Abstract: Provide a callable interface to the MAIL facility.

Description: Many applications require automatic delivery of status messages or notifications to one or more users, something which can be very effectively done via MAIL. However, implementing such a service via exit commands or DECL subprocess spawns is cumbersome and error-prone. It would be very useful to have a MAIL facility that is callable by a user program directly. Separate entry points for each function or a facility to accept a properly formatted text string or file would be acceptable. This facility could also be used to build a gateway between mail systems in a multivendor installation.

SIR: F85-39

Abstract: Provide a /NOADVANCE option on the DCL WRITE command.

Description: DCL should have a mechanism which allows a string to be written to SYSS\$OUTPUT, without positioning to the next line. This facility would be useful when exact positioning control is required, for example when generating output onto a pre-printed hardcopy form.

SIR: F85-40

Abstract: SET TERMINAL/INQUIRE should be able to determine the mode of a VT200 series terminal without changing that mode.

Description: With VMS V4.1, a VT220 terminal set in VT100 mode is automatically changed to VT220 mode by the command SET TERMINAL/INQUIRE. This is contrary to the previous treatment of VT52 mode on VT100 terminals and is in general a disservice. Although there may be cases where changing

VAX-57

the mode is desired, there are plenty of cases where only inquiry is needed; for instance where a distinction needs to be made between video and hardcopy terminals, without disturbing a user's choice of terminal mode. Perhaps an optional qualifier would be in order.

SIR: F85-41

Abstract: Enable DCL READ command to extract fields from a record.

Description: Currently the DCL READ statement returns the entire contents of a record into a DCL symbol. READ should be enhanced to be able to divide the record among several symbols. For example,

```
READ FOO BAR(1:10), BAZ(13:15), ...
```

would return characters 1 through 10 of the record in symbol BAR, characters 13 through 15 in BAZ, etc. In addition, a simpler, delimiter-based format could also be supported.

```
READ/DELIMITER=";" FOO BAR, BAZ, ...
```

would extract the fields delimited by the specified character.

System Management

SIR: F85-42

Abstract: Enhance the network support in BACKUP.

Description: BACKUP should have enhanced networking capabilities:

1. It should generate a remote backup object to handle magtape and file selection remotely from remote tapes or save-sets.
2. Standalone BACKUP should have rudimentary DECnet capability.

3. Standalone BACKUP should be down-line loadable over Ethernet.

BACKUP is extremely useful, but it lacks certain important networking capabilities. The ability for BACKUP to invoke a remote network object version of itself would have 2 useful results: the remote object could mount and handle the magnetic tape (with increased reliability due to BACKUP's superior tape handling), and it could do file selection from remote save-sets which would greatly decrease network traffic. (At present, the whole save-set must be processed by FAL and the files locally selected by BACKUP). Some rudimentary DECnet support in Standalone BACKUP is a big piece missing from the MicroVAX support. Anyone who has loaded 31 floppies will agree. (We won't all be ordering TK50's). Having BACKUP down-line loadable over Ethernet would be the ideal in support for remote systems.

SIR: F85-43

Abstract: Summary ACCOUNTING report should list actual times of first and last entries.

Description: When generating summary reports with the ACCOUNTING utility the input file often does not cover the entire period specified. There is no way of knowing from the summary report if this is the case. The ACCOUNTING utility should extract the first and last dates represented in the data and include that in the summary report.

SIR: F85-44

Abstract: Add volume initialization parameters to Standalone BACKUP.

Description: The Standalone BACKUP utility should allow the user to initialize an output volume with explicitly specified parameters. This is particularly important for single-disk systems, where it is otherwise impossible to change volume parameters such as cluster factor.

SIR: F85-45

Abstract: Provide facilities for estimating and monitoring time required for a BACKUP.

Description: BACKUP utility should optionally be able to estimate the time and number of tapes required for a BACKUP operation. This would allow operators to budget their time and tape volumes.

BACKUP should also include the currently active file spec

as part of its message requesting a new tape volume. This would allow the operator to monitor the progress of the backup and would allow selection of tape volumes of the appropriate size.

SIR: F85-46

Abstract: Expand STANDALONE BACKUP to allow the use of some subset of DCL.

Description: The STANDALONE BACKUP utility requires the user to type the BACKUP command line manually. Often this command can be rather large and complex. In the hands of a less knowledgeable user, an error in the command line can have disastrous effects. It would be useful to be able to use a command file to invoke the utility. If some subset of DCL were available it would be easy to prompt the user through the BACKUP process.

SIR: F85-47

Abstract: Improve STANDALONE BACKUP's handling of write-protected disks.

Description: To avoid accidents with STANDALONE BACKUP, it is desirable to write-protect the disk being backed up. This is unworkable if the backup date is to be recorded following the operation. BACKUP should re-check the write-ability of the disk when it begins the recording pass, and possibly ask for operator intervention to make the disk write-able.

SIR: F85-48

Abstract: Provide a "manual recover" mode for BACKUP

Description: While BACKUP has generally excellent error recovery capabilities, there are occasionally times when some part of the tape information cannot be read. For example, a damaged label might prevent a saveset from being usable, or a damaged file name might prevent the data from being reloaded. For such cases, BACKUP should have an interactive "recovery" mode in which the user could be prompted to supply values for missing information.

Obviously, such a facility would make it impossible to guarantee the integrity of the data restored. It would be the user's responsibility to review and assess the validity of the data. However, often recovering incomplete data may be better than not recovering any at all.

SIR: F85-49

Abstract: Provide a screen editor interface to AUTHORIZE.

Description: VMS should provide a screen oriented extension to the AUTHORIZE utility. This would simplify the increasing complex task of adding and maintaining usernames. This facility should be based on the new TPU editor and should be user extensible to allow for local enhancements.

SIR: F85-50

Abstract: Add /INTERACTIVE and /IMAGE qualifiers to SHOW SYSTEM and MONITOR PROCESS.

Description: When examining the processes on the system, it is sometimes only the interactive processes which are of interest. SHOW SYSTEM should have an /INTERACTIVE qualifier similar to its /BATCH qualifier. This is particularly important on systems which have a large number of detached or batch processes which clutter the display. The same feature should also be available in MONITOR/PROCESS.

Both utilities should also have the ability to display the currently executing image for all processes. This is particularly important in MONITOR when it is essential to know what image a troublesome process is running.

SIR: F85-51

Abstract: Support SYSSWELCOME in group logical name tables.

Description: It is frequently necessary to tailor the appearance of a VAX for different user groups (e.g., in-house users, research sponsors, application users, etc.). To aid in such tailoring it should be possible to define SYSSWELCOME in the group logical name tables (instead of just in the system table). A requirement for EXEC mode logical names would provide additional control over use of this feature.

SIR: F85-52

Abstract: Provide more flexibility in the SHUTDOWN procedure.

Description: The current facility for site modifications to the SHUTDOWN procedure (SYSHUTDOWN.COM) is not sufficiently flexible. It would be useful to have an additional site exit available at the time the SHUTDOWN is initially invoked, which could be minutes or hours before the actual shutdown occurs. Such an exit would allow all shutdown related questions to be asked and answered at the same

time. It would also allow "idle-down" processing (i.e., stop accepting new work) to be started, before a final cutoff in SYSHUTDOWN.

SIR: F85-53

Abstract: Provide support for "restricted" DCL environments

Description: It should be possible to create an environment similar to a captive command procedure by reducing the facilities accessible from DCL. This would be much more convenient than custom building a new command interpreter (the captive procedure) for each application. The current SET COMMAND utility provides the base of this capability. Additional support is required to insure that only the images specified in the DCL tables could be executed, and that tables could not be modified by the captive user.

Even if this facility were somewhat complex to use, it would probably be easier than insuring each custom written captive procedure is secure and effective.

SIR: F85-54

Abstract: "Unbundle" the functions of the CAPTIVE login flag

Description: The login flag CAPTIVE is currently the only way to enforce the execution of the command file specified in the /LGICMD field. Unfortunately, CAPTIVE also disables the use of the login /DISK qualifier and implies the flag DEFCLI. This is not necessary when the /LGICMD field contains a full file spec for the procedure. Control of the /DISK qualifier and of the /LGICMD should be available as separate flags.

SIR: F85-55

Abstract: Enhance the SHOW command in AUTHORIZE

Description: Very often a need arises to check all of the users in the UAF for one particular AUTHORIZE parameter, such as who has what privileges, or how many people have an increased WSEXTENT. It would be very handy to be able to enter "SHOW/WSEXTENT [*,*] rather than having to write a program to do this or go through a full record for hundreds or thousands of usernames.

This could be extended also with the ability to specify a range for the parameter within which to choose records.

Another handy item would be to extend the wildcard specifiers for UICs to allow the use of "%" signs. This way an entire range of UIC's can be scanned or listed without

having to include the entire file or enter the command repeatedly.

SIR: F85-56

Abstract: VMS should log the fact that one process was stopped by another

Description: VMS should make an entry in the accounting record of a process if that process is stopped by another process. The entry should show the Process ID of the process causing the termination. This would provide some measure of accountability and aid in tracing malicious users.

Commercial

SIR: F85-57

Abstract: Provide for dependency networks of print and batch jobs.

Description: Large production shops often have the need to specify the interdependencies between the running of related jobs. For example, job A should run after job B and job C have run; Jobs D and E should be printed after job F has finished, etc. The SYNCHRONIZE command provides a limited form of this capability. It is desirable to have a simple mechanism to specify larger networks of dependencies for both print and batch.

SIR: F85-58

Abstract: Enhance handling of TAB characters in SORT

Description: With the present SORT software, it is very difficult to sort input files which contain the horizontal TAB character. Such a file usually is produced by entering data from a terminal. It is very difficult to determine the sort key position. This is because the records are displayed (via TYPE for example) with TABs expanded while SORT treats the TAB as a single character. If "spacing" in the file is created with a combination of TABs and blanks, it is even more difficult to properly sort the file. The SORT facility should have an option to expand TAB characters to blanks prior to isolation of the SORT keys.

SIR: F85-59

Abstract: Support Asynch printers through a terminal server.

Description: A large source of terminal I/Os on many systems is the use of asynchronous terminal printers (LA50s, LA120s, etc.). It would be desirable to offload some of this I/O overhead by connecting these printers to an Ethernet terminal server. The use of a terminal server also provides additional flexibility and redundancy.

The DEC terminal servers should be enhanced to fully support such printers. Since these devices might be remotely located and may not have keyboards, it must be possible to configure "dedicated" connections for these devices in the server. Since VMS print queues must be defined for these devices, they must appear to the VAX with fixed, known device names.

SIR: F85-60

Abstract: VMS should implement tape automatic volume recognition and provide the security normally associated with volume labeling.

Description: VMS should provide a complete implementation of automatic volume recognition for tapes, that may be enabled/disabled by the operator on a per drive basis. This means that (with AVR enabled), when a tape is mounted, the system checks possible labels and honors mount requests without operator intervention, if possible. If a job needs 4 tapes, the operator can mount them all if enough drives are available and then forget about them until somebody else needs the tape drives. It should also be possible for a user to request a tape mount based solely on the tape's label and density. The user should not be required to know what physical devices implement a particular tape density on a particular system. VMS should also support a "visual id" or "slot number" which is displayed in all operator messages related to the mount.

It should be possible to operate a VMS system in a mode where all tapes are under system/operator control. This means that they are pre-initialized and users are not allowed to change the labeling on the tape without special privilege. The BACKUP utility must also conform to such labeling restrictions, thereby insuring that the BACKUP data is written onto the proper reels. VMS should require explicit operator intervention for unlabeled tapes. It is not acceptable that an unlabeled tape which happens to be on a drive be automatically assigned.

VAX-64

SIR: F85-61

Abstract: Support RMS segmented keys of different types

Description: RMS should allow segmented keys where the segments are of different types. For example, it might be desirable to have one segment be a string and another an integer. This would be particularly useful since the VAX convention for storing the bytes of an integer does not support sorting an integer as if it were a string.

SIR: F85-62

Abstract: Add support for descending keys in RMS indexed files

Description: It should be possible to access RMS files using both ascending and descending keys. There are many query applications where such facility provides the most natural solution.

Languages and Tools

SIR: F85-63

Abstract: VMS data structure definitions and entry point definitions should be provided for ALL DEC languages to the fullest extent that they can be used by that language.

Description: When using high-level languages, it is extremely inefficient for each user to code and maintain the definitions of VMS data structures and entry points (RTL and system services). Some languages currently lack even the system service and STARLET data structures. Most languages lack the VMSRTL and LIB definitions. In the case of Pascal, the STARLET data structure definitions provided are not strongly typed to the degree which would be expected by someone who has chosen that language to take advantage of its strong typing. While the "escape hatches" to avoid strong typing may be an appropriate option to give users, they should not be used by DEC as a cheap way out of properly defining the entry points and data structures for ALL languages.

SIR: F85-64

VAX-65

Abstract: Provide a /D_LINES (debug lines) compile-time switch on the VAX-11 C compiler

Description: VAX-11 C should provide an external mechanism to conditionally compile code intended for use while debugging C programs like the /D_LINES switch implemented in VAX-11 FORTRAN.

SIR: F85-65

Abstract: The DEBUGGER should permit MACRO language syntax when debugging high-level-language source code.

Description: There are occasions when it is necessary to examine data structures, such as stacks and argument lists. This cannot usually be done from the high-level-language syntax, so it becomes necessary to perform a "SET LANGUAGE MACRO" command. Later, it may be necessary to revert to the original high-level-language syntax. It is awkward to have to "toggle" between MACRO language syntax and a high-level-language syntax while using the debugger. When parsing a debugger command, it would be convenient if the debugger could first apply the high-level-language syntax rules. If the command could not be properly parsed, then the debugger would revert to the MACRO syntax.

IR: F85-66

Abstract: All DEC supplied editors for VMS should follow a common protocol for protecting against simultaneous edits to the same file.

Description: While source code control software is normally used to synchronize source code changes, there are situations where it is not appropriate (e.g. changing the system startup command procedure). All DEC supplied editors for VMS should use a consistent mechanism to insure that two processes cannot be simultaneously editing the same file. In addition, this mechanism must allow other readers to access the file. Under VMS V4.1, EDT and TECO both handle this differently, so as not to be interlocked against each other. TECO, additionally does not allow for readers to get the old version of the file by default.

Perhaps one implementation mechanism would be to have writers of a replacement version lock the PREVIOUS version for write (allowing other readers). A prime criterion for evaluating this scheme would be to ensure that DCL can execute a command procedure which is open for write.

SIR: F85-67

Abstract: Enhance the pattern matching capabilities of the EDT search command to include column bounds and "negative" searches.

Description: EDT would be more useful if it allowed more elaborate specifications of the pattern to be searched for. It should be possible to restrict the column positions to be searched. For example: FIND "ABC"/POS=1:15 would search for the first occurrence of the string "ABC" located in columns 1 through 15. It would also be helpful to have a "negative" search specification. For example: FIND -"Mr."/POS=1 would find the next line which did NOT begin with the string "Mr.".

SIR: F85-68

Abstract: The select region in EDT should remain active after the region is written to a file, so that further operations can be performed on it.

Description: In EDT a range of code can be selected in KEYPAD mode, and this range can be written to a file. This action causes the code to be de-selected, and no further operations can be performed unless the range is selected again. There needs to be an option which controls whether a selected region is de-selected after an operation is performed on it.

SIR: F85-69

Abstract: EDT should leave an extra space at the end of each line when word-wrapping is performed.

Description: It would be useful if there was an option that would cause EDT to leave a space at the end of each line when performing word-wrapping. This action would greatly simplify the transfer of text files by DECdx to DECmates. Currently, when DEXdx inserts a soft return, the last word on one line will be concatenated with the first word on the following line, after DECmate re-wraps the text. The current work-around is to manually insert end-of-line blanks.

SIR: F85-70

Abstract: Provide unnumbered FORMAT statements to improve the readability and appearance of FORTRAN code.

Description: Many FORTRAN FORMAT statements are used just once, and placing the FORMAT statements immediately after the I/O statements makes it easier to follow the code. When these FORMAT statements are in an indented section of code, the statement numbers destroy the appearance. Eliminating unnecessary statement numbers also makes it easier to locate statement numbers which are targets of branches.

SIR: F85-71

Abstract: Provide a method to improve the readability of RUNOFF input files.

Description: It is difficult to keep track of elements in nested lists, when the list depth exceeds two. If indentation were permitted on lines that contained list directives, readability could be greatly improved.

Security

SIR: F85-72

Abstract: User-controllable mechanism needed to allow other users to access a file only via a user-defined image.

Description: Non-privileged users sometimes need to give other non-privileged users controlled access to data files through a program. Through this facility any user would be able to control who could access his data files and what kind of access they may have. In the current system, in order to allow another user to add a record to a file, that user must be given WRITE access to that file, which means he could alter existing data or delete records from the file.

Presently this requires the system manager to install the program with privilege, which is both an administrative nuisance and a security problem, as the privileged image would also have access to other system data files as well as the intended files. This mechanism should be under user control, i.e., the user should be able to specify which images could access a file. For example, the UIC of the image and data file should probably be required to match before the access would be permitted. This could be accomplished by an option on the compiler or linker when the image was being created. It could also be implemented by allowing the system manager to install an image with a particular identifier (VMS 4.0) and then setting up the access control list for that file to permit access by that

VAX-68

identifier. This would be less flexible but would permit a user to allow access from images other than his own, e.g., a data base manager.

SIR: F85-73

Abstract: Implement government standard security classifications for files (i.e., unclassified, confidential, secret, and top secret) and control access to files based on individual user's security clearance.

Description: Many VAX systems are being operated by government agencies or contractors and either are processing or need to process classified data. Such a facility would make system management much easier for existing systems and would encourage more VAXes to be used for classified processing. The system manager should be able to specify a security level for each user account using the AUTHORIZE utility. When a file is created, it should be given the security level of the creating process. If the file is edited or copied, it should retain its classification. A utility should be provided to allow a person with a (SECURITY) privilege to change the classification of a file.

SIR: F85-74

Abstract: End-to-end DES encryption should be provided within DECnet-VAX.

Description: The VAX/VMS system should support end-to-end DES encryption within DECnet-VAX with a separate DES key being used for each DECnet logical connection. This should be implemented at the NSP level, so that it is transparent to the user. The system manager should be able to activate or deactivate DES encryption between his VAX and any other VAX (that supports that feature). Privileged users on intermediate nodes can read and/or modify data being routed through their nodes or observe data in transit across an Ethernet, and not all nodes in a large DECnet network are equally trustworthy. End-to-end DES encryption would serve to protect this data in transit. Also, DES might be used to protect need-to-know access to classified data in a network.

SIR: F85-75

Abstract: Better node authentication should be provided by DECnet-VAX.

VAX-69

Description: The node authentication provided by DECnet-VAX in the form of node transmit and receive passwords is woefully inadequate. Not only is it easily circumvented, but it is only applicable to adjacent nodes. A better node authentication capability is desired, perhaps one that is encryption based, so that a system can have a high degree of certainty that the nodes with which it is communicating are who they claim to be. Note that it is not necessarily desired that all data be encrypted, as that may entail a high overhead. Also, although this request pertains to DECnet rather than simply to VMS, it is necessary to have adequate node authentication within DECnet in order to ensure the security of a VAX within a DECnet network.

Improvements to the SIR Process

The purpose of this section is to obtain your input on how the SIR process can be improved. The following include a number of suggestions which have been received at past DECUS symposia. Please let us know how you feel on each of these items. To simplify the tally, use the following scale to describe your preference:

- 10 Strongly agree
- + 5 Somewhat agree
- 0 Neutral or no opinion
- 5 Somewhat disagree
- 10 Strongly disagree

Enter your response on the specially numbered lines on the ballot. This section will be summarized separately from the rest of the ballot.

SIR: F85-100

Abstract: The SIR ballot is important enough to be conducted by a separate mailing.

Description: The SIR ballot has traditionally been conducted through the Pageswapper. The primary advantage of this mechanism has been the low cost. However, since the advent of paid newsletter subscriptions, not all SIG members receive the Pageswapper. If a wider distribution of the ballot is really important, the SIG could pay to conduct one or both ballots by directly mailing to the SIG membership. This would, however, reduce the funds available for projects in other areas.

SIR: F85-101

Abstract: The process of filling out an SIR ballot should be simplified.

Description: It has been suggested that people are discouraged from participating in the SIR ballot because of the time it takes to do the voting. The voting scheme used on the SIR ballot has traditionally involved dividing 100 points among the available SIRs, with a 10 vote per SIR limit. An alternative format might list all of the SIRs and ask for a 1 to 5 rating for each SIR. The goal of the original scheme was to force voters to strongly distinguish between the most important SIRs and those of only moderate interest. The alternative format might encourage people to mark "5" for every SIR, rather than carefully considering each one.

SIR: F85-102

Abstract: The quality of the SIR items should be improved.

Description: One possible reason that people don't participate in the SIR ballot is that they don't find anything worth voting on. This may be due to lack of interesting SIRs, or it may be due to unclear or poorly worded ballot items. The ballot is currently reviewed by the VAX SIG working groups to generate clear, comprehensive, technically correct items. Emphasis could be placed on improving this review process.

SIR: F85-103

Abstract: Some form of electronic voting on SIRs should be available.

Description: With electronic mail becoming increasingly familiar to many people, some form of electronic voting might become feasible as an adjunct to the paper ballot. The large size of the ballot text (and the time required to analyze it) might discourage online reading of the entire ballot. However, some mechanism of electronic vote collection is more feasible, and could be investigated, if there is sufficient interest.

SIR: F85-104

Abstract: Some percentage of the SIRs not making top 10 should be automatically included on the next ballot.

Description: The typical SIR ballot contains 50 to 75 items, of which DEC responds to the top ten. Most good ideas tend to be submitted repeatedly if they fail to make the cutoff. To help insure that good ideas are not lost, the second 10 or 15 items could automatically be added to the next ballot. A possible objection to this policy is that, over the course of a year, action by DEC or users might alter the importance or direction of a previously submitted SIR.

SIR: F85-105

Abstract: The SIR process should try to encourage the inclusion of hardware improvements.

Description: While the SIR ballot has always been open to all VAX issues (software, hardware, service, etc.) almost all SIRs received pertain to software. An effort could be made to encourage the inclusion of hardware items on the ballot. It may be that hardware requests would not show strong popularity on the ballot. The various groups within the VAX community might not share common hardware interests to the same degree as software interests. Traditionally, it has been harder to get hardware issues addressed by Digital. The SIR ballot might lend extra credibility to such efforts.

SIR: F85-106

Abstract: The SIR process should narrow its scope to concentrate on issues which ONLY Digital can resolve.

Description: It has been suggested that too many ballot items are "wasted" on requests which can be resolved by user written programs. These tend to obscure requests which can only be solved by DEC (changes to the operating system or major features). However, some users do not have the resources to develop their own solutions and cannot depend on unsupported "public domain" software for critical applications.

SIR: F85-107

Abstract: Several SIR's requesting small enhancements to a particular program should be combined on the ballot.

Description: Very often a number of SIR's are received requesting very minor additions to a particular program (e.g. BACKUP). When they are combined together, there is more chance for that combined ballot item to make top ten status. This does make it impossible to oppose one particular change out of a group of desired ones.

VAX System SIG Committee List

As of July 18, 1985

Osman K. Ahmad - TOPS-VAX
Association of American Railroads
Technical Center, Research and Test Department
3140 South Federal Street
Chicago, IL 60616

Joe Angelico - System Management
US Coast Guard CCGD8(DT)
Hale Boggs Federal Building
500 Camp Street, New Orleans, LA. 70130

June Baker - Planning

Elizabeth Bailey - Volunteer Coordinator
222 CEB
Tennessee Valley Authority
Muscle Shoals, AL 35660

Joe L. Bingham - Librarian
Mantech International
2320 Mill Road
Alexandria, VA 22314

C. Doug Brown - Security
Sandia Labs
P.O. Box 2644
Albuquerque, NM 87185

Jack Cundiff - Assistant Symposium Coordinator
Muskingum College
New Concord, OH 43762

James R. Cutler - Hardware
Software Results Corporation
2887 Silver Drive
Columbus, OH 43211

Tom Danforth - Handout Editor
Woods Hole Oceanographic Institute
Woods Hole, MA 02543

Doug Dickey - Data Management SIG Interface
CTEC, Inc.
6862 Elm Street
McLean, VA 22101

Jim Downward - Migration and Host Development
KMS Fusion Inc.
3621 South State Road, P.O. Box 1567
Ann Arbor MI 48106

Dan Fleury - Education
University of Hartford
West Hartford, CT 06117

Dennis Frayne - Real Time/Process Control
McDonnell Douglas
5301 Bolsa Avenue
Huntington Beach, CA 92646

Carl E. Friedberg - Internals
In House Systems
165 William Street
New York, NY 10038

Bob Boyd - Commercial
GE Microelectronics Center
MS 2P-04
Post Office Box 13409
Research Triangle Park, NC 27709

Don Golden - Overseas Coordinator / Publications Coordinator
c/o Shell Development Company, D-2132
Westhollow Research Center
Post Office Box 1380
Houston, TX 77001

Gary Grebus - System Improvement Request
Battelle Columbus Labs
Room 11-6011
505 King Avenue
Columbus, OH 43201-2693

B. Hancock - Network
Sohio Petroleum Company
Two Lincoln Center
5420 LBJ Freeway, Suite 900/LB 03
Dallas, TX 75240

R. Haydt - Foreign Devices, Hardware/Software
Information Consultants International Incorporated
P. O. Box 2014, E. V. STA
Ormond Beach, FL 32074

Jeffrey S. Jalbert - Symposium Coordinator
J C C
Post Office Box 381
Granville, OH 43023
614-587-0157

Ken Johnson - VAXcluster Working Group Chair
Meridian Technology Corporation
Post Office Box 2006
St. Louis, MO 63011

Lawrence J. Kilgallen - Newsletter Editor
Box 81, MIT Station
Cambridge, MA 02139-0901

Margaret Knox - Chair
Computation Center
University of Texas
Austin, Texas 78712

Ross W. Miller - Vice Chair and Working Group Coordinator
Online Data Processing, Inc.
N 637 Hamilton
Spokane, WA 99202

Bob Robbins - VAXelan
Array Computer Consultants
5364 Woodvale Drive
Sarasota, FL 33582

Larry Robertson - Real Time/Process Control
Bear Computer Systems Inc.
5651 Case Avenue
North Hollywood, CA

P. Sandwell - Graphics
Seismograph Service Corporation
P. O. Box 1590
Tulsa, OK 74102

David Schmidt - LUG Coordinator
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
Mantech International
2320 Mill Road
Alexandria, VA 22314

Louise Wholey - Languages and Tools
Measorex Corporation
One Results Way
Cupertino, CA 95014

Douglas J. Wilson - Office Automation
MIT Joint Computer Facility
Room 5-137, 22 Massachusetts Avenue
Cambridge, MA 02139

INPUT/OUTPUT

A SIG Information Interchange

A form for INPUT/OUTPUT submissions is available at the back of the issue.

indent (5) blanks before (1) test page (2)

INPUT/OUTPUT 433

Caption: Hewlett-Packard 7550A Plotter -- Reply to I/O # 371

Message: We use the following for PLOT QUEUE definitions for hp 7585:

```
$ SET TERM
TTAL:/DEVICE_TYPE=UNKNOWN/SPEED=9600/PERM/NOWRAP/-
NOECHO/NOFORM/HARDCOPY/NOBROADCAST/WIDTH=511
! Plotter TDV
$ SET DEV/SPOOLED TTAL:
$!
$! INIT/START QUEUES
$ DEFINE/FORM/WIDTH=511 PLOTTER 1
$ INIT/QUEUE/START/TERM/-
DEFAULT=(NOFLAG,NOBURST,NOFEED,NOTRAILER)/-
SEPARATE=(NOBURST,NOFLAG,NOTRAILER,NORESET)/-
FORM=PLOTTER/SCHED=NOSIZE TTAL:
$ INIT/QUE/START/GENERIC=55A1: HP7585
```

We then send HP-GL-Plotfiles with record length 511 to the device by

```
$ PRINT/QUE=HP7585/FORM=PLOTTER/NOHEAD file-spec
```

Under VMS 4.1, DEF/FORM/WIDTH is necessary instead of SET TERM/WIDTH under VMS 3.7.

Contact: Bernd Grothkopp
Robert Bosch GmbH/ZWI
Postfach 50
D 7000 Stuttgart 1
West Germany
Telephone 0711-8116538

Date: July 3, 1985

INPUT/OUTPUT 434

Caption: Disable/Enable CTRL/Y in FORTRAN code -- Reply to I/O # 409

Message: Use the run-time library routines:

LIB\$DISABLE_CTRL (Volume 5B page RTL-79)

LIB\$ENABLE_CTRL (Volume 5B page RTL-90)

These routines are available under VMS version 3 and 4.

Contact: Glenn Zorn
Bunker Ramo
95 Merritt Boulevard
Trumbull, CT 06611
Telephone (203) 386-2098

Date: May 23, 1985

INPUT/OUTPUT 435

Caption: Disable CTRL/Y -- Reply to I/O # 409

Message: Refer to the Run-Time Library Routines (LIB\$ENABLE_CTRL) and (LIB\$DISABLE_CTRL). If you disable control Y in a program, you need to enable it within the program after you are done. If you don't, CTRL/Y will be disabled for the process completely until you enable it or logoff. Refer to V4.0 documentation. It gives a better explanation than V3.n documentation.

Contact: Patrick J. Holmay
Academic Computer Services
Science Hall, Room 018
St. John's University
Collegeville, MN 56321
Telephone (612) 363-2706

Date: May 29, 1985

VAX-78

INPUT/OUTPUT 436

Caption: Disable CTRL/Y -- Reply to I/O # 409

Message: One way to disable control/Y in a FORTRAN program is to use the system service LIB\$M_CLI_CTRL_Y, as in the following example:

```

external lib$m_cli_ctrl_y
mask=%loc(lib$m_cli_ctrl_y)
status=lib$disable_ctrl(mask,oldmask)
!(disables control Y)
c
c *****
c Body of program where control Y is disabled
c *****
c
status=lib$enable_ctrl(mask,oldmask) !(enables
control Y)
end

```

Contact: Christopher M. Gordon
Hughes Research Labs RL86
3011 Malibu Canyon Road
Malibu, CA 90265

Date: May 30, 1985

INPUT/OUTPUT 437

Caption: How to Disable CTRL/Y -- Reply to I/O # 409

Message: The Run-Time Library allows enabling or disabling CTRL-Y from any high level language. In FORTRAN:

```

IOSTAT = LIB$DISABLE_CTRL('02000000'X)

or

IOSTAT = LIB$ENABLE_CTRL('02000000'X)

```

Contact: Ron Williams
Southwest Research Institute
Div15
6220 Culebra Road
San Antonio, TX 78285
Telephone (512) 684-5111 ext 3490

Date: Jun 4, 1985

VAX-79

INPUT/OUTPUT 438

Caption: VAX to Prime/Prime to VAX file transfer

Message: We are looking for a fast (>= 9600 baud), easy to use, file transfer program between VMS and Primos, possibly using X.25 interface. Other than KERMIT and BLAST, which use asynchronous transmission, is anything available?

Contact: Jack Patteeuw
Ford Motor Company
Diversified Products Technical Center - C309
17000 Rotunda Drive
Dearborn, MI 48121
Telephone (313) 323-8643

Date: May 31, 1985

INPUT/OUTPUT 439

Caption: Ditto Foreign Tape Copy

Message: Can anyone suggest a method for doing a bit-wise copy of a tape with protection for bad spots on the target tape? This comes up in a requirement for making copies of IBM, UNIVAC, PRIME and DEC20 tapes on a VAX. IBM has a DITTO program but we couldn't use that to copy VAX BACKUP tapes (tested prior to /INTERCHANGE mode) and there may be similar problems going the other way.

Contact: Edward E. L. Mitchell
Mitchell and Gauthier Associates
290 Baker Avenue
Concord, MA 01742
Telephone (617) 369-5115

Date: May 30, 1985

VAX-80

INPUT/OUTPUT 440

Caption: VAX to LSI via DAV11

Message: I would like to talk to anyone using a DAV11 to communicate between a VAX and an LSI (PDP11/03). I am interested in sharing information about drivers, diagnostics, etc.

Contact: Pete Ruch
SPACECMD/INDPI
Cheyenne Mountain Complex, CO 80914-5601
Telephone (303) 473-4010 x3529

Date: June 24, 1985

INPUT/OUTPUT 441

Caption: Backup to tape via DECnet -- Reply to I/O # 418

Message: Sourcnode command :
\$ BACKUP XYZ TAPENODE::"0=NETTAPE"/SAVE/BLOC=4096

nettape.com (use proxy access):
\$ ALLO MTA0:
\$ MOUNT MTA0: LABEL /BLOC=4096
\$ CONVERT SYS\$NET:/FDL=NETI
MTA0:SRCDAT.BCK/FDL=MTA0OUT

netin.fdl:
ORG=SEQ; CAR-CON=NONE; FORMAT=VARIABLE; SIZE=4096

netout.fdl:
ORG=SEQ; CAR-CON=NONE; FORMAT=FIXED; SIZE=4096

Contact: Allen P. Rueter
510 South Kingshighway Boulevard Bx8131
Department of Radiology
Washington University
St. Louis, MO 63110
Telephone (314) 362-7133

Date: May 28, 1985

VAX-81

INPUT/OUTPUT 442

Caption: VMS <--> RSX DECnet

Message: We have a MicroVAX I, with a DHV11 running MicroVMS 4.0 and end-node DECnet. We are trying to communicate, using asynchronous DECnet, to a PDP-11/23+ with a DLV11E, running RSX-11M Version 4.1. We are able to get each end to work in loop-back, but can not get the two systems to talk to each other. If we connect each system to Ethernet, they will communicate. Although DEC claims (in the VMS software product description) that this is not supported, has anyone been successful in getting VMS and RSX to communicate asynchronously?

Contact: James R. Bennett
 Manager of Engineering Services
 Geniographics Corporation
 Post Office Box 591
 Liverpool, NY 13088
 Telephone (315) 451-6600 ext. 226

Pageswapper Editor's Response

I believe I recall people at the Spring US Symposium in New Orleans saying that there had been a protocol implementation error in the VMS async DECnet which has since been corrected to allow the two systems to communicate.

INPUT/OUTPUT 443

Caption: SUPREM Semiconductor Processing Simulator

Message: I need to run SUPREM II or SUPREM III on our 11/750 under VMS. SUPREM is a program which originated at Stanford University and which simulates the processing involved in semiconductor manufacture. Please contact me if you have this program running under VMS or if you know of anyone who has it.

Contact: Bill Jones
 Datalinear Corporation
 11211 Prosperity Farms Road
 Palm Beach Gardens, FL 33410
 Telephone (305) 694-8050

Date: May 30, 1985

INPUT/OUTPUT 444

Caption: Exquota from Print Symbionts driving Versatec

Message: Versatec device drivers check process quotas by calling EXE\$BUFFERQUOTA after tab expansion. Tab expansion causes increase in record size making buffer larger than the SYSGEN parameter MAXBUF. To fix: change driver to call EXE\$BUFQUOPRC which bypasses comparison with IOC\$GW_MAXBUF.

Contact: Tom Moog
 Building 203
 Argonne National Lab
 9700 South Cass Avenue
 Argonne, IL 60439
 Telephone (312) 972-3073

Date: June 6, 1985

INPUT/OUTPUT 445

Caption: Disable CTRL/Y -- Reply to I/O # 409

Message: This is something I use a lot when I have to write code which requires the process be installed with privileges not given to normal users, however, it has to be used by them. It is described in detail in the Run-Time Library Manual.

```
include '($libclidef)'
```

```
!+
! Disable control Y recognition
!-
    istatus=lib$disable_ctrl(lib$m_cli_ctrly)
    if(.not. istatus) call lib$stop(%val(istatus))
```

```
!+
! Enable control Y recognition
!-
    istat=lib$enable_ctrl(lib$m_cli_ctrly)
    if(.not. istat) call lib$stop(%val(istatus))
```

Contact: T.W. Heim Jr. (Tom)
Simpact Associates, Incorporated
5520 Ruffin Road
San Diego, CA 92123-1390

Date: June 6, 1985

INPUT/OUTPUT 446

Caption: Operator Interface Program -- Reply to I/O # 410

Message: The company I used to work for sells a product they developed called SADI which sounds like the menu interface you described. SADI is an inexpensive, easy-to-use menu system which transfer control to and from a spawned subprocess. Contact:
Mark Crego
c/o ManTech Services Corporation
2320 Mill Road
Alexandria, VA 22314

Contact: Herbert J. Matthews IV
E.R.C.
5686 Medallion Court
Alexandria, VA 22303
Telephone (703) 663-2177

Date: June 6, 1985

INPUT/OUTPUT 447

Caption: Disable CTRL/Y -- Reply to I/O # 409

Message: The disabling and enabling of CTRL/Y can be accomplished from a Fortran program by calling the run-time library procedures LIB\$ENABLE_CTRL and LIB\$DISABLE_CTRL. Check the RTL manual for the usage. Instead of disabling CTRL/y, you might want to detect it, trap it, set a flag, and handle it in your program with a normal exit. Read about out-of-band AST's under QIO (IO\$SETMODE!IO\$M_OUTBAND) in the system services manual.

Contact: Herbert J. Matthews IV
E.R.C.
5686 Medallion Court
Alexandria, VA 22303
Telephone (703) 663-2177

VAX-84

Date: June 6, 1985

INPUT/OUTPUT 448

Caption: Return Receipt in VAXmail

Message: I am interested in a VMS program that will provide a return receipt for VAXmail under Version 4.1.

Contact: Charles Sheets
Walter Reed Army Institute of Research
6825 16th Street, North West
Building 83, Division of Biometrics
Washington, DC 20307-5100
Telephone (202) 576-3146

Date: June 18, 1985

INPUT/OUTPUT 449

Caption: Seeking information on dual 780 on one SBI using MS780 memory

Message: I have a vague memory of reading in the Pageswapper about a user who put two 780 CPU's together on one backplane thus making it a 782 without the multiport memory. We have a 782 with MA780 multiport memory but would like to expand beyond its capacity. Using MS780 memory seems to be the only option. None of my back issues seem to contain this article. Has anyone actually made this conversion?

Contact: David Ruhoff
Digital Productions
3416 South LaCienega Boulevard
Los Angeles, CA 90016
Telephone (213) 938-1111

Date: June 23, 1985

Pageswapper Editor's Response

The Pageswapper article was "LUG Meeting Reports" (essentially a paraphrase from a LUG newsletter) in Volume 6 Number 1 (July 1984) referring to work done with VMS. Those who did that work referred to an article in the preprint volume of the 9th Annual Symposium on Computer Architecture regarding a prior effort

VAX-85

of the same nature under Unix.

INPUT/OUTPUT 450

Caption: P and WHAT for VMS 4.x

Message: I'm still on version 3.7 of VMS on a VAX 11/750. I will be moving to version 4.x soon, and I will be losing two valuable but unsupported tools - P and WHAT.

P is, as explained in the notes I have, "... a program for looking at other jobs, somewhat similar to 'J' of 'K' on the pdp-10." The sources are P.MAR, DPY.MAR, DPYDEF.MAR, P.COM, the system symbol table, and P.HLP. I could modify it if I were a guru, but... Does anyone have a version of P that runs on v4.x?

For WHAT, I have no notes or source. It comes up showing a dynamic display of the processes. You can SHOW MAIL, SHOW MUTEXES, ADD columns to various displays, etc. This is a big help to watch messages going from process to process, identify MWAIT problems, etc. Same question, does anybody know someone who has this goodie?

Contact: Frank W. Croft
Post Office Box 2042
Wilmington, NC 28403

Date: June 26, 1985

INPUT/OUTPUT 451

Caption: SNOBOL for VMS

Message: Does anyone have or know of a native mode SNOBOL interpreter and/or compiler for VMS?

Contact: Clyde T. Poole
University of New Orleans
Computer Research Center
New Orleans, LA 70148
Telephone (504) 286-6760

Date: June 26, 1985

VAX-86

INPUT/OUTPUT 452

Caption: VAX Pascal Pretty Printer -- Reply to I/O # 409

Message: We have a so-called Pascal Formatter which indents the lines in dependence of the Pascal keywords, translation of the key words to upper/lower case. It is simple to use. The documentation is in the German language.

Contact: Rudolf, Reusch
AEG
Hafenstr
D 2000 Wedel
West Germany
Telephone 04103/200-565

Date: July 4, 1985

INPUT/OUTPUT 453

Caption: Reading IBM 3470 exchange Floppies -- Reply to I/O # 379

Message: We have a program for reading IBM Exchange Format Floppies (8 inch single density, 128 bytes per sector) which runs under VMS (MCR). The software and source are:

FLXIBM
ERIK A. Rosdol
District Software Support
AUEA/VIENNA Project VMB 707013

Contact: Alfonso Garcia Cores
System Manager
Banco Pastor
Paseo de Recoletos, 19
28004 - Madrid
Spain
Telephone 1-4335800 exte. 264

Date: July 8, 1985

VAX-87

INPUT/OUTPUT 454

Caption: Disable CTRL/Y -- Reply to I/O # 409

Message: Fortran Routines are provided to disable and enable CTRL/Y or C:

```
SUBROUTINE DISABLE
J=0 ! INTEGER*4
J=JIBSET(J,25) ! SETBIT 25
II=LIB$DISABLE_CTRL(%REF(J),)
RETURN END
call DISABLE to disable CTRL/Y

SUBROUTINE ENABLE
J=0
J=JIBSET(J,25) ! SETBIT 25
II=LIB$ENABLE_CTRL(%REF(J),)
RETURN END
call ENABLE to enable CTRL/Y
```

Contact: Paul Paskaran
Planning and Transportation Department
Westminster City Council
Victoria Street
London SW1
UK
Telephone 01-798-2626

Date: July 8, 1985

INPUT/OUTPUT 455

Caption: Looking for Route Accounting Package

Message: We are in need of a route accounting package for laundries. This is to run under VMS on an 11/750.

Contact: Jim Hatlelid
Midco Data System, Incorporated
Box 5009
Minot, ND 58702
Telephone (701) 857-1155

Date: July 15, 1985

Steering Committee Nominations

It's that time again! Every two years the VAX SIG elects seven (7) people to be the steering committee. These seven people have the primary responsibility for the operations of the SIG. Prior to each Fall DECUS Symposium the steering committee meets to elect its chair and to make job assignments. A few additional non-voting members may be added to this core group in order to fill major areas. Currently the major areas are: chair, vice-chair, secretary, symposia (2 positions), volunteers, publisher, newsletter editor, special projects, SIRs, local user groups, library, tape copy, and working group coordination.

Who can be nominated? Any US Chapter DECUS member of the VAX SIG who has five DECUS members sign a nomination form.

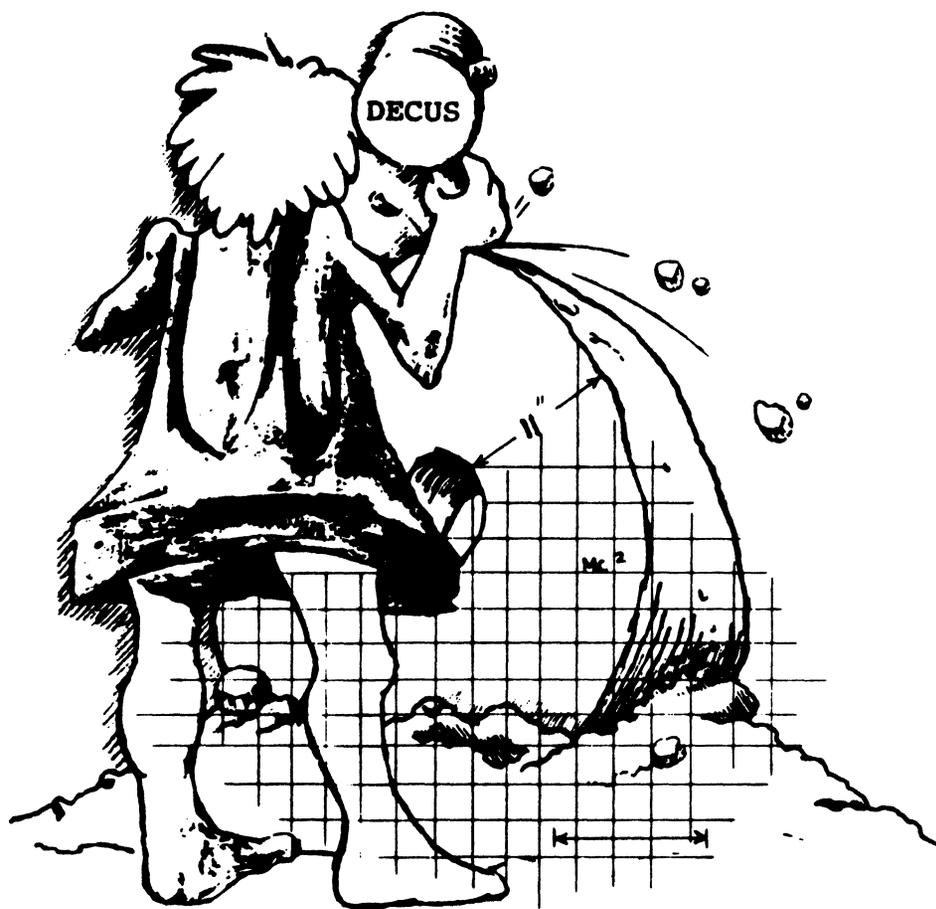
What are the time commitments? The time commitment varies during the year and with the job(s) to be done. A rough average is 2-10 hours per week. This includes weekly (minimum) electronic communications with other DECUS leaders via DECUS's DCS system, and attending each symposia including the extra 1-2 leadership days. The steering committee conducts one or two "woods meetings" per year -- these planning meetings are usually 2 1/2 days each. There may be other out-of-town meetings depending on your responsibilities. And finally, there is the actual work to be done! DECUS will pay for non-symposia travel and expenses incurred.

What are the benefits? This level of involvement in DECUS does require a substantial commitment from you and support from your employer is really important. What you and your employer get in return is a closer working relationship with DEC, especially in product development. Your contacts within the VAX community are also a big benefit. And DECUS provides on-the-job training in areas such as management, public speaking, group dynamics, politics, and organization.

What is the schedule? The nomination forms are due to the DECUS office no later than 15 September. And yes, you may nominate yourself! You just need a total of five supporting signatures of DECUS members. The nominee's resume is also due into the office at the same time. The balloting will be done in late September-early October and the results announced at Fall DECUS.



LIBRARY



DECUS PROGRAM LIBRARY LIBRARY HINTS AND NOTES

What does the DECUS Library offer for the Professional-300 Series?

The DECUS Library has many interesting programs which run on the PRO-300 series of computers. They run the range from text editing/word processing programs such as TECO and RUNOFF to language and database systems such as the C LANGUAGE SYSTEM and DBMS (Data Base Management System).

What about software for UNIX?

A new program named "RSX-P/OS Tar for Floppy Diskettes", DECUS Program Order Number: PRO-134, will be of great interest to UNIX users. This program makes it possible to read/write to UNIX TAR format floppies on RX01, RX02, or RX50 drives under RSX or P/OS operating systems.

How does the DECUS Library distribute programs for the PRO?

All of the PRO programs we currently offer are for either the P/OS or PRO/RT-11 operating systems. The programs come in the Digital standard 5 1/4 inch floppy diskette (RX50K).

Which DECUS offerings can I use on my PRO?

There is often confusion among users about which programs they can use on their PRO. The P/OS operating system is a subset of Digital's RSX-11 operating system, while PRO/RT-11 was derived from Digital's PDP-11 RT-11 operating system. P/OS can access floppies which are listed in the DECUS Program Library Catalog as being in "FILES-11" format and cannot access floppies in "RT-11" format. The reverse holds true also, PRO/RT-11 can access floppies which are listed as being in "RT-11" format and cannot access floppies in "FILES-11" format. Take care to note this when ordering PRO programs from the DECUS Program Library.

DECUS Program Library CHANGES:

- A new media code (JE), for the 5 1/4" Floppy Diskettes is being announced with DECUS Order No: PRO-139. The U.S. Chapter service charge will be \$83.00 for a DECUS member and \$99.00 for a non-member.

- For DECUS Order Number: 11-SP-67, Symposium Tape from RSX SIG, Spring 1984, Cincinnati, the abstract cross references DECUS Order No: V-SP-27 as the same package but in VMS/BACKUP Format. This is incorrect, the package in VMS/BACKUP Format is DECUS Order No: V-SP-28. Please note this change in your catalog.

ORDER NUMBER: 11-48

TITLE: TRACE Debugging Program, Version: November 1971

MEDIA CODES: (AA), (GC)

STATUS: NO LONGER AVAILABLE

DATE: APRIL 1985

COMMENT: The DECUS Library no longer distributes paper tape.

ORDER NUMBER: 11-243

TITLE: LISTER: Listing Utility Program, Version: December 1975

MEDIA CODES: (DA)

STATUS: NO LONGER AVAILABLE

DATE: MAY 1985

COMMENT: Unable to reproduce a legible copy of the offered media.

ORDER NUMBER: VAX-57

TITLE: PAM: Package for Analogue Modelling, Version: V3.0, October 1982

MEDIA CODES: (EB), (MA)

STATUS: TEMPORARILY ON HOLD - UNDER EVALUATION

DATE: JULY 1985

COMMENT: User reported problems. Awaiting input from the Author.

NEW AND REVISED LIBRARY PROGRAMS

NEW LIBRARY PROGRAMS AVAILABLE FOR THE PROFESSIONAL-300 SERIES OF COMPUTERS

DECUS PROGRAM ORDER NUMBER: PRO-134
Title: RSX-P/OS Tar for Floppy Diskettes,
Version: V1, April 1985

Author: R. Gaughan and G. Everhart
Submitted By: Glenn C. Everhart, Ph.D.
Operating System: IAS, RSX-11 D, RSX-11 M, RSX-11 M-PLUS, P/S,
Source Language: MACRO-11,
Special Hardware Required: RX01, RX02, or RX50 drives,
Keywords: Utilities - P/OS, Utilities - RSX-11

Abstract: This program allows read/write to UNIX (tm) TAR format floppies on RX01, RX02, or RX50 drives under the RSX or P/OS operating systems. All sources, including a version of SUPERMAC that will work with them are included, plus objects and a P/OS task image. Necessary adjuncts including a task to mount RX50's foreign under P/OS are presented.

Also on the disk are a P/OS version of the RSX SRD Working Group SRD utility and R. Kirkman's image mode RX50 copier for P/OS, and an inspect-only file lister from the Fall 1984 RSX SIG tape with some local enhancements.

Using the TAR program it is possible to move files between P/OS and various flavors of UNIX on floppy. This can be handy where communications utilities are unavailable on one end or the other, or where faster throughput is needed than is possible on even a very high speed communications line.

UNIX is a trademark of AT&T Bell Laboratories.

Documentation on magnetic media.

Media (Service Charge Code): 5 1/4" Floppy Diskette (JA), Format: FILES-11

DECUS PROGRAM ORDER NUMBER: PRO-135
Title: Easycom/PRO for the Professional - 350/380 Series, Version: V1.0-06, April 1985

Author: Lee Knoch
Submitted By: Digital Equipment Corporation
Operating System: P/OS V1.7, 2.0, 2.0A,
Source Language: FORTRAN 77,
Memory Required: 256KB,
Other Software Required: PRO/COMM V1.8, 2.0,
Special Hardware Required: A hard disk is required.
Keywords: Data Communications, Emulators

Abstract: Easycom/PRO is a program for the PRO-350 and 380 series of computers running under P/OS which is patterned after (and enhanced over) the DECmate Easycomm Application. Easycom/PRO automatically logs you into a computer system or database after which you enter PRO/Comm Terminal Emulation. You simply create a login (or script) file describing what you manually do (often in a lot of steps) to log in. After this is done, you run Easycom/PRO and select the proper script file. Additional features of Easycom/PRO include the ability to work with either the PRO Communications port or the TMS modem and to define a "default" script file which you select merely by pressing "return". An editor is built right in so you need not worry about how to create the script files either. Several examples as well as the User's Guide (manual) are contained in the kit.

A brief User's Guide (manual) may be found on the diskette in [USERFILES] and is named EASYCOM.DOC. Print it out for the Easycom/PRO command syntax. You may want to print out the example Easycom/PRO files on the kit too. These all end with ".EZC" and are in the [USERFILES] directory on the floppy.

Sources not included. Documentation on magnetic media. Media (Service Charge Code): Manual (EA), 5 1/4" Floppy Diskette (JA), Format: FILES-11

DECUS PROGRAM ORDER NUMBER: PRO-136
Title: PRO/VLINK for the Professional - 350/380 Series, Version: V1.0-06, April 1985

Author: Lee Knoch
Submitted By: Digital Equipment Corporation
Operating System: P/OS V2.0, 2.0A
Source Language: FORTRAN 77,
Memory Required: 256KB,
Other Software Required: PRO/Tool Kit V2.0,
Special Hardware Required: A hard disk is required.
Keywords: Tools - Applications Development

Abstract: To create a running program on the Professional 350 or 380, the program must be compiled and LINKed. Before the program can be LINKed, the user must create a task builder command file and an overlay descriptor file.

Creating these files can prove to be a formidable obstacle to the initial P/OS user, often taking many times longer than the application development time itself.

PRO/VLINK creates these complex files (as well as the P/OS Hard Disk Application Installation file) for the application developer and allows the application development cycle to be simply create (program), compile, link and run.

PRO/VLINK supports the following source languages: FORTRAN, PASCAL, BASIC, DIBOL and MACRO. Task builder files for COBOL and DECUS C are created but may need additional editing.

Subroutine calls from the following facilities are supported: P/OS, POSRES, POSSUM, COMLIB, CGL, FMS, PRTIL and DECnet.

Sources not included. Documentation on magnetic media. Media (Service Charge Code): Manual (EA), 5 1/4" Floppy Diskette (JA), **Format:** FILES-11

DECUS PROGRAM ORDER NUMBER: PRO-137 Title: Adventure for the Professional - 300 Series, **Version:** V1, January 1984

Submitted By: Glenn C. Everhart, Ph.D. **Operating System:** RSX-11M, RSX-11M-PLUS, P/OS **Source Language:** FORTRAN IV, **Keywords:** Games

Abstract: Adventure is a magical, unpredictable and often addicting computer game that has caught on in the United States in now epidemic proportions. It is a treasure hunt with all the trimmings, mysteries and challenges that grow more and more complex as the game unravels. Adventure is more of a puzzle than a game. Once solved, it's mastered! The mastering, however, often takes months of drawing maps and planning strategy. Adventures sweeping popularity lies in the power to enchant. Players are projected into a world of fantasy, one that blends the heart-pounding suspense of Treasure Island with the magic of Alice in Wonderland.

Documentation on magnetic media. Media (Service Charge Code): 5 1/4" Floppy Diskette (JA), **Format:** FILES-11

DECUS PROGRAM ORDER NUMBER: PRO-138 Title: Airplane Lander for the Professional-300 Series, **Version:** V1, May 1985

Submitted By: Glenn C. Everhart, Ph.D. **Operating System:** P/OS, **Keywords:** Games

Abstract: This program is an airplane landing simulation game. It provides a pseudo-graphic display of an aircraft instrument panel with real time updates at one second intervals. The program simulates a real instrument landing approach from an altitude of 25000 feet to the runway, with instructions from ground radar control. Aircraft climbs, dives and stalls are properly simulated. An off airport landing as well as go-around for a missed approach are both possible.

Source code is supplied for both VT100 compatible and VT52 compatible terminals, and command files are supplied to enable versions to be produced for background, foreground and system job.

Restrictions: Needs tailoring to change systems dependent calls to P/OS. Tools for doing so are supplied.

Documentation on magnetic media. Media (Service Charge Code): 5 1/4" Floppy Diskette (JA), **Format:** FILES-11

DECUS PROGRAM ORDER NUMBER: PRO-139 Title: DBMS: Data Base Management Package for the Professional-300 Series, **Version:** V1, February 1984

Author: R. DiMarco **Submitted By:** Glenn C. Everhart, Ph.D. **Operating System:** P/OS, **Source Language:** MACRO-11, **Keywords:** Data Base Management

Abstract: The database package was designed to allow small, homogeneous databases to be quickly set up and manipulated. The package provides the user with the following facilities:

1. An extremely simple method of defining the structure of the records which make up the database.

2. A screen oriented database editor, which allows records to be created, edited and deleted. The editor supports protected and database managed fields. The editor was designed so that persons with little computer experience (i.e. typists, clerks, etc.) can use it.

3. A report generation package, which allows specific records to be selected, and reports generated. The records to be included in the report can be sorted by any field within the record, and the report format can be easily modified.

4. A FORTRAN interface which allows records in the database to be readily accessed from a FORTRAN mainline program. Fields within a record can be accessed via their logical names. A FORTRAN interface is also provided to the sorting module, in the event that the user needs to implement a special application program. The FORTRAN interface also allows the user to access two or more databases simultaneously.

5. A menu program is included to allow the user to change between the various application programs without the knowledge of their computer interfacing.

This version (V2) of the Data Base Management Package includes an easier method of defining and establishing a database, and a much larger manual. The procedures discussed in the manual are illustrated in a pre-built database for managing telephone information which is included.

Note: A hard copy of the manual is available under DECUS No. 11-529.

Restrictions: Reformatted for P/OS. Needs work to actually run under P/OS (RSX) but conversion aids are included in the package.

Documentation on magnetic media. Media (Service Charge Code): 5 1/4" Floppy Diskette (JE), **Format:** FILES-11

DECUS PROGRAM ORDER NUMBER: PRO-140 Title: CGS: Common Graphics System for the Professional-300 Series, **Version:** May 1985

Submitted By: John F. Davis **Operating System:** RT-11 V5.1, **Source Language:** FORTRAN IV, MACRO-11, **Memory Required:** Applications dependent, **Other Software Required:** FORTRAN, MACRO, RUNOFF (to generate documentation).

Keywords: Graphics, Libraries - RT-11, Professional-300 Series - RT-11

Abstract: The Common Graphics System (CGS) is a library of FORTRAN callable subroutines that provide general purpose graphics primitives across a variety of main frame computers. RT CGS supports the same set of user-visible primitives on the PRO-300 series.

The RT CGS library includes a metafile drive. Metafile format is fully documented. Other device drivers, including use-written drivers, can be linked but this capability has not yet been implemented.

Examples of utilities to plot metafiles on the PRO screen and on a pen plotter are provided.

Complete sources and documentation are provided for the RT CGS Library, but only sources are given for the main utility routines. Object modules are furnished so that the utility examples can be modified and relinked.

Restrictions: 2-D primitives only. I/O via metafiles and translator utilities.

Complete sources not included. Documentation on magnetic media. Media (Service Charge Code): 5 1/4" Floppy Diskette (JB), **Format:** RT-11

DECUS PROGRAM ORDER NUMBER: PRO-141 Title: TTLIB: VT100 Library Sources for the Professional-300 Series, **Version:** V1, May 1985

Submitted By: Glenn C. Everhart, Ph.D. **Operating System:** P/OS, **Source Language:** MACRO-11, **Keywords:** VT100 Routines

Abstract: TTLIB is a library of programs to conveniently control a VT100 type terminal in ANSI mode. Routines allow drawing boxes and lines, cursor positioning, screen appearance, video attributes, screen and line clearing, screen and keyboard behavior, graphics facilities, assorted heights and widths, tab settings and clearings, and reporting cursor position.

Restrictions: Needs some tailoring to run under P/OS. Tools to do so are provided but not instantly usable as is.

Documentation on magnetic media. Media (Service Charge Code): 5 1/4" Floppy Diskette (JA), **Format:** FILES-11

NEW LIBRARY PROGRAMS AVAILABLE FOR CP/M

DECUS PROGRAM ORDER NUMBER: CPM-259 Title: Bar Graph Generator, Version: V6.02, October 1984

Author: William A. Seacrist, South Central Power, Lancaster, OH, **Operating System:** CP/M V2.2, **Source Language:** MBASIC, **Memory Required:** 64K Bytes, **Other Software Required:** MBASIC, **Special Hardware Required:** LA50 Printer or equivalent, **Keywords:** Graphics

Abstract: Graph is a BASIC program that uses the VT180(ROBIN) and the LA50 printer to their full capacity. This program has internal documentation and is packed full of whistles and

bells. Laymen need not fear, this is a menu-driven, user-friendly program. Between two and twelve graphic bars may be printed. The width of the bars change depending upon how many are going to be printed. The average of all bars may be displayed with a triple graphics line across the bars. The latest update of this program includes a routine that allows users to save their graph data so it can be loaded and modified at a later time.

Note: Works best with LA50 printer
Documentation on magnetic media.
Media (Service Charge Code): 5/4" Floppy Diskette (JA)

NEW LIBRARY PROGRAMS AVAILABLE FOR THE PDP-11 COMPUTER FAMILY

DECUS PROGRAM ORDER NUMBER: 11-794 Title: WHO for RSX-11M Without KMS Support, Version: V3.0, May 1985

Author: Raymond Tai and Larry Tai, University of California, Davis, CA **Operating System:** RSX-11M V4.1, **Source Language:** MACRO-11, **Memory Required:** 128K, **Keywords:** Utilities - RSX-11

Abstract: This WHO differs from the James G. Downward's WHO-KMS Fusion Inc. in that it does not require any KMS Fusion routines or hooks built into the operating system. This WHO performs what the old WHO does plus it displays the logged-in and current UICs, flags privileged users and the user that's invoking WHO. Furthermore, it displays the users' default SY:, CLI and the last logged-in date and time.

Note: This program has not been tested under any other version of RSX-11M.

Restrictions: A maximum of 200 accounts and a maximum of 4 active tasks displayed.

Documentation on magnetic media.
Media (Service Charge Code): Write-Up(AA), Floppy Diskette, (KA), 600' Magtape (MA), **Format:** FILES-11

DECUS PROGRAM ORDER NUMBER: 11-795 Title: GRAPHKIT: Graphics Routines for the HP-7221 C Plotter, Version: V1, April 1984

Author: 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, **Other Software Required:** Hewlett-Packard Plot/21 software library, **Special Hardware Required:** Hewlett-Packard 7221 C 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.

Documentation on magnetic media. **Media (Service Charge Code):** Write-Up (AA), Floppy Diskette (KA), 600' Magtape (MA), **Format:** FILES-11

DECUS PROGRAM ORDER NUMBER: 11-796 Title: FDIR: Fast Directory Program for RSTS/E, Version: V2.2, April 1985

Author: Andreas Luik, Esslingen, West Germany **Operating System:** RSTS/E V7.0 or later because of UUD call 25 (Wildcard PPN lookup) **Source Language:** PASCAL(OMSI V2.0), **Memory Required:** 28KW, **Keywords:** Sort, System Management - RSTS/E, Utilities - RSTS/E

Abstract: The FDIR-Package consists of the programs FDIR(Fast Directory), SDIR(Sort Directory) and some other files (e.g., command files, documentation). FDIR will create a directory listing of the accounts existing on any RSTS/E V7 (V8 has not been tested, but should also work) disk device (for example RL01/Q2, RP05). Directories of DECTapes or magtapes are not possible. FDIR reads the directory data directly from the User-File-Directory (UFD) of each account.

The functions of FDIR are similar to those of \$DIRECT, but you should recognize, that FDIR is about three times faster than DIRECT and about six times than PIP and has more features (e.g., date specifiers, matching protection codes, VT100 options are supported and last but not least, sorting is available).

SDIR is used to reorder the directory listings produced by FDIR, PIP or DIRECT. Sorting is possible by several keys, for example filename, extension, creation date or position on disk, in forward or reverse order.

Restrictions: No file attributes supported, directories of magtapes or DECTapes are not possible.

Documentation on magnetic media. **Media (Service Charge Code):** 600' Magtape (MA), **Format:** DOS-11

DECUS PROGRAM ORDER NUMBER: 11-797 Title: LPV07: Lineprinter Handler for HT-11/RT-11 V02C, Version: V07/11, May 1985

Author: Anthony P. Cruz, Roseville, MI **Operating System:** RT-11 V2C(HT-11 11/79) **Source Language:** MACRO-11, **Memory Required:** 432-550 Words (option dep.), **Special Hardware Required:** Any ASCII printer connected to a DLV-11 or DLV-11 "like" interface addressed at 177510 and vectored at 200., **Keywords:** Device Handlers

Abstract: LPV07.MAC is the culmination of a long effort to develop a functional and truly useful "LP" device driver. The major underlying goal was to develop a driver that would PROPERLY support very modest printers such as DEC'S LA35 Lacking Forms Control Option.

Such drivers have been around for some time. However, systems lacking adequate operating system support were often furnished with only the binary versions of the LP driver. Moreover, such versions often times were capable of handling only advanced lineprinters equipped with forms control hardware, automatic perforation skipping and hardware handshaking. One operating system typical of those lacking the appropriate LP driver is HEATH'S HT-11 system. This operating system which was available exclusively to owners of HEATH'S H/WH-11 mini-computer (an LSI-11-based product), is actually a somewhat "diluted" version of DEC'S RT-11 V02C. As the need arose, and/or as hardware improvements were made at my installation, new features were added to the existing driver. Following the purchase of a HEATHKIT H-125 Lineprinter, I decided to develop a final "no-holds-barred" driver, capable of handling the H-125 AND anything inferior to it by simply using the appropriate conditional assembly file. This driver is the result of that effort and should be a boon to users of HT-11 or RT-11 V02C.

Documentation on magnetic media. **Media (Service Charge Code):** Write-Up and Listing (DC), Floppy Diskette (KA), 600' Magtape (MA), **Format:** RT-11

DECUS PROGRAM ORDER NUMBER: 11-798
Title: ANOVA1: A Routine for Analysis of Variance.
Version: November 1984 **Author:** Bob Melino, Xerox Corp., Webster, NY **Operating System:** RT-11 V5.1. **Source Language:** FORTRAN IV. **Memory Required:** 64 KB. **Keywords:** Statistics

Abstract: Information to be entered is irep, items and the data. Irep = number of replicates and items = number of items. Irep can be in the range of 2 to 20. Items can be up to 15. You may continue

NEW LIBRARY PROGRAMS AVAILABLE FOR VAX/VMS

DECUS PROGRAM ORDER NUMBER: VAX-123
Title: VSH: A Shell (Command Interpreter) for VAX/VMS, **Version:** V1.0, October 1984

Author: Camillo Bongiovanni, CSZ, Torino, Italy
Operating System: VAX/VMS V3.5 **Source Language:** C. **Keywords:** Language Interpreters, Tools - Software Development

Abstract: A shell is a command language interpreter. VSH is the name of a particular command interpreter on VAX/VMS. The primary purpose of VSH is to translate command lines typed at a terminal into system actions, such as invocation of other programs. VSH is a user program, just like any one might write. It incorporates all the features of DCL and a history mechanism; most of the features of VSH are designed mainly for interactive VMS users. Hopefully, VSH will be a very useful program for everyone in interacting with the VAX/VMS Operating System. In addition, there are some useful utility programs that can be used in order to facilitate software development; their source files were taken from the UNIX Operating System, and adapted for VAX/VMS. They are not fully tested, thus questions and/or problems encountered are invited.

Note: Release notes distributed with each order. **Restrictions:** "C" I/O functions are quite different between UNIX and VMS, thus complete compatibility between UNIX-VMS is not easily attained.

Documentation on magnetic media. Media (Service Charge Code): 600' Magtape (MA), **Format:** VMS/BACKUP (Blocked at 8192)

the analysis by using the NEWMAN-KEULS RANGE TEST.

This program will allow you to enter new data or use an old or merged data files. It will write the data file on the disk in either the default name of ANOVA1.DAT or a user selected .DAT file.

Documentation on magnetic media. Media (Service Charge Code): Floppy Diskette (KA), 600' Magtape (MA), **Format:** RT-11

DECUS PROGRAM ORDER NUMBER: VAX-131
Title: Orthotron Testing and Data Storage Program, **Version:** April 1985

Author: David R. Kraay and Jonathan D. VanOss, Hope College, Holland, MI **Submitted By:** William K. Anderson **Operating System:** VAX/VMS V3.7 **Source Language:** VAX-11 COBOL V3.0, **Memory Required:** 4,096,000 Bytes, **Other Software Required:** SPSS-X 2.0, **Special Hardware Required:** VT101 or VT220 Terminal, **Keywords:** Sports

Abstract: This program has been designed for use with the Orthotron II R (manufactured by Lumex Inc.). It allows the user to store test and daily exercise data for individual subjects using the machine. Subject groups have been divided into three sections:

1. Health Enhancement Program Subjects - Pre and post test data storage. Allows for before and after program comparison.
2. Pre-season Athletes - Allows the user to enter up to three sets of test data for up to three sports for any one particular athlete.
3. Rehabilitation Subjects - Allows the user to enter exercise sessions for any individual performing rehabilitation on the machine. Table and graphs are used to allow the user to compare exercise sessions.

The Health Enhancement Program and Pre-season screening sections of the program compute testing statistics for each subject. Computed statistics include comparison ratios, percent weaker ratios, and strength to body weight

ratios. In addition, support programs have been included to allow the user to do statistical analysis using the SPSS-X statistical software package. Statistical analysis includes descriptive data on sample groups, comparison of pre and post test data, and frequency tables.

The system comes with its own self-starting login file. It is recommended that this file be automatically executed by the user for each use of the system.

Note: Screen printouts for user instruction are not included on magnetic media.

Documentation on magnetic media. Media (Service Charge Code): Manual (EB), 600' Magtape (MA), **Format:** VMS/BACKUP (Blocked at 8192)

DECUS PROGRAM ORDER NUMBER: VAX-132
Title: MASSGRAF: A Program that Generates Graphics Images, **Version:** V1.0, March 1985

Author: Judi Cleary, Sohio Research Center, Cleveland, OH **Operating System:** VAX/VMS V3.7, **Source Language:** VAX-11 FORTRAN, **Memory Required:** 200-300 blocks depending on which device driver is used., **Other Software Required:** DI-3000(tm) subroutine library, licensed by Precision Visuals, Inc., **Special Hardware Required:** Graphics device to display graphics - consult Precision Visuals, Inc. for list of devices., **Keywords:** Graphics

Abstract: MASSGRAF is a graphics program that generates graphics images which can later be included into a word-processing document. These images, consisting of basic geometric shapes and variations on boxes, arrows, circles, etc. can enhance textual information and aid communication. Various line widths and line styles are available.

Using a command-driven interface, the user generates a graphics file, one page at a time. This "page" of graphics can be displayed on a graphics terminal, but was designed to output to a laser printer. Once created, a graphics file can be "included" in a MASS-11(tm) document and be printed as consolidated pages.

Editing of the graphics image can be done by using MASSGRAF or the EDT editor and editing the command file. DI-3000 is a trademark of Precision Visuals, Inc.

Restrictions: This software is based on device-independent (DI-3000) graphics subroutine lib-

rary. An executable file must be built for each device driver. It is also designed to be used with a word processing package called MASS-11. However, it can be used separately. Associated Documentation: Additional documentation on DI-3000 library can be obtained from Precision Visuals, Inc.

Documentation on magnetic media. Media (Service Charge Code): 600' Magtape (MA), **Format:** VMS/BACKUP (Blocked at 8192)

DECUS PROGRAM ORDER NUMBER: VAX-133
Title: GRAF11: A Package to Graph Scientific Data, **Version:** V1.2, March 1985

Author: Judi Cleary, Sohio Research Center, Cleveland, OH **Operating System:** VAX/VMS V3.7, **Source Language:** VAX-11 FORTRAN, **Memory Required:** 400-600 blocks depending on which device driver is used., **Other Software Required:** DI-3000 (tm) subroutine library, licensed by Precision Visuals Inc., **Special Hardware Required:** Graphics device to display graphics - Consult Precision Visuals Inc., for list of devices., **Keywords:** Graphics, Scientific Applications

Abstract: GRAF11 is a graphics package that provides an easy way to graph scientific data. GRAF11 uses an interactive, command-driven interface. Many commands have default values which can be easily overridden.

Commands and data can be entered from the keyboard or from a file. Graph formats include linegraphs, barcharts and scattergraphs, using linear, log, calendar or probability axes. Curve fitting and smoothing can also be done.

GRAF-11 was written using DI-3000 (tm) graphics library and is therefore quite "device independent" regarding graphics display. Graphs from GRAF-11 can also be merged with MASS-11 (tm) word-processing documents.

DI3000 is a trademark of Precision Visuals Inc.

Restrictions: This software is based on device-independent (DI-3000) graphics subroutine library. An executable file must be built for each device driver.

Associated Documentation: Additional documentation on DI-3000 (tm) library can be obtained from Precision Visuals, Inc.

Documentation on magnetic media. Media (Service Charge Code): 600' Magtape (MA), **Format:** VMS/BACKUP (Blocked at 8192)

NEW LIBRARY PROGRAMS AVAILABLE FOR THE DECSYSTEM-20 FAMILY OF COMPUTERS

DECUS PROGRAM ORDER NUMBER: 20-SP-8
Title: Symposium Tape from the DECSYSTEM-20 SIG, Fall 1984, Anaheim, **Version:** Fall 1984

Author: Various **Submitted By:** Betsy Ramsey, American Mathematical Society, Providence, RI **Operating System:** TOPS-20 release V5.1, **Source Language:** BLISS-36, MACRO-20, GNOSIS, **Keywords:** Symposia Tapes - TOPS-20, Kermit, Utilities - TOPS-20

Abstract: The TOPS-20 Symposium Tape from Fall 1984 (Anaheim) contains TAPE11 an ANSI-standard tape utility and other programs from Emerson Electric; GNOSIS CAI programs from

University of Vermont; SYSLIB, a set of callable routines and USR, a multi-system username program from Energy Enterprises; GTJFN enhancements and ANAL crash dump analysis program from SUMEX; and the November 1984 release of KERMIT from Bernie Eiben.

No guarantees are made as to the completeness, usability or quality of the programs on this tape and the material has not been checked or reviewed.

Documentation on magnetic media. Media (Service Charge Code): 2400' Magtape (PB)

REVISIONS TO LIBRARY PROGRAMS

DECUS PROGRAM ORDER NUMBER: PRO-102
Title: BFGUSER, **Version:** V1.2, May 1985

Author: Jack Wenrick, et al, BF Goodrich R&D, Breckville, OH, **Operating System:** P/OS V2.0, **Source Language:** FORTRAN 77, PRO/BASIC, **Other Software Required:** PRO/BASIC for BASIC programs, **Special Hardware Required:** LA50 or LA100 for hardcopy of plots, LVP16 plotter, hard disk required for PROPLOT, **Keywords:** Graphics, Runoff, Text Formatting

Abstract: This diskette contains software for the PRO-350 developed at BFGoodrich R&D. [USERFILES] This directory contains several programs written in PRO/BASIC. All of these programs will run on any PRO-325 or 350 which has PRO/BASIC.

USELA50 - A menu driven printing program which allows selection of vertical and horizontal pitch and other attributes on the LA50 printer and print documents or data files.

MERGEFILES - A program to merge two or more files together.

AMORTIZE - Calculates amortized loan schedule based on the amount borrowed, interest rate, and number of payments.

MOLWT - Calculates molecular weight from a molecular formula and weight percent of each of the atoms.

TYPRINT - Prints directly to LA50 from keyboard.

TEXTMOD - Finds character strings in a BASIC program and can do substitution for all occurrences of the string, list lines that contain the string, and list line numbers that contain the string.

[PROPLOT] and [PROPLC] - This is a data graphing and polynomial curve fitting program. It is written in FORTRAN and was run through the PRO/Toolkit to create the Native version. It can fit up to six curves per set of axes to a 1 to 5th degree polynomial and selecting linear or Log scales for the X and Y axis. Monochrome and color monitor versions provided.

[ZZRNO] This is a PRO version of BONNER Labs RUNOFF, the best version of RUNOFF I have used.

[DTR] This directory contains a PRO/DATATRIEVE initialization file to declare Global definitions in DATATRIEVE to allow control of printer (LA50 or LA100) and screen attributes.

Note: Release notes distributed with each order. Changes and Improvements: PROPLOT is now a PRO-350 native program which originates large hardcopy graphics on a printer. PROPLOT is an installable application.

Documentation on magnetic media. Media (Service Charge Code): 5 1/4" Floppy Diskette (JB), **Format:** FILES-11

DECUS PROGRAM ORDER NUMBER: 11-352
Title: DR11-A/C and DRV11 Loadable Driver, **Version:** April 1984

Author: R. E. Beverly III, Ph.D., R. E. Beverly III and Associates, Columbus, OH **Operating System:** RSX-11M V3.0 or later **Source Language:** FORTRAN 77, MACRO-11, **Memory Required:** Approx 1600 Words, **Special Hardware Required:** DR11-A or -L or DRV11 parallel I/O interface, **Keywords:** Device Handlers

Abstract: The DR11 Loadable Driver will allow a user to access up to sixteen (16) DR11-A/C or DRV11 general device interfaces using standard QIO procedures under an RSX-11M Version 4 mapped system. The driver supports read, write, attach, detach, and interrupt requests. Interrupts are realized by the setting of event flags in the task, thereby eliminating the need for the user to handle the interrupt himself.

A build package, consisting of all sources for the driver, plus a command build file, and a set of sources for FORTRAN-77 callable routines to control the driver, as well as full documentation, are included in this release version.

Changes and Improvements: Bugs fixed, compatible with RSX-11M Version 4 and FORTRAN 77, operation verified with DRV11 parallel I/O interface. **Restrictions:** Loadable driver support and user-written driver support (\$GTWRD and \$PTWRD) must be selected at RSX-11M sysgen time.

Documentation on magnetic media. Media (Service Charge Code): Write-Up (AA), Floppy Diskette (KA), 600' Magtape (MA), **Format:** FILES-11

DECUS PROGRAM ORDER NUMBER: 11-594
Title: CPU Usage Monitor Display Facility for RSX-11M. **Version:** X02.05, August 1984

Author: Y. N. Miles, TRIUMF, U.B.C. Vancouver, B.C. Canada, **Operating System:** RSX-11M V4.1, **Source Language:** MACRO-11, **Memory Required:** 1K (+ 256 words of POOL !), **Special Hardware Required:** VT52 / VT100 type CRT terminal. Extended instruction set in CPU, **Keywords:** System Management - RSX-11

Abstract: USE is a CPU-usage display facility which shows on a video terminal a bar graph of the eight most CPU-intensive processes. This is achieved by loading a histogram driver into pool, and calling this driver directly from the (KW11) clock interrupt vector. The histogram driver checks sign bit on saved PSL (minus if task, + if system process), and then stores the process name (task: \$TKTCB -- cur TCB), I/O state (saved PSL has PR4 set), idle state (low order bit set in \$IDLFL), else \$FORK state..

USE requires a CRT with Digital Equipment VT52 escape sequences. It needs to have privilege, and it needs executive global symbols contained in LB:[1,54]RSX11M.STB. All fields, documentation, generation, and source are contained in one universal library USE.ULB. To generate USE, type @USE.ULB/LB:USEGEN.

Changes and Improvements: Shows time spent in \$FORK and \$INTERRUPT states; less system overhead than previous version. **Restrictions:** Must be linked with LB:[1,54]RSX-11M.STB, Events sampled on system clock, requires 256 words of POOL (gives it back when exit).

Documentation on magnetic media. Media (Service Charge Code): 600' Magtape (MA), **Format:** BRU V4.1

DECUS PROGRAM ORDER NUMBER: 11-765
Title: RSX SIG Tapes Evaluation, **Version:** V3, December 1984

Author: A. Szentgali **Submitted By:** Klaus Centmayer, TU Muenchen, Munich, West Germany **Operating System:** IAS, RSX-11M, **Keywords:** Symposia Tapes -RSX-11

Abstract: This collection of reports is a review of programs from the DECUS RSX Symposium Tapes. Its goal is to evaluate the programs and their building procedures and to help users in choosing and installing software according to their actual needs and configuration. Testing

includes building and installation procedure and, as far as possible, a brief run test. This report contains the US-RSX-SIG-Tapes Spring and Fall '82.

The tape includes a SIG-Tape Road Map Summary as a quick reference. It contains:

RSX-IAS US Fall'77 ... Spring'84, Europe'79...'83; PASCAL Spring'80 ... Fall'81, RT-11 Fall'79 ... Fall'81; Lars Palmer + IAS-ICR collections.

Changes and Improvements: Testing of additional programs.

This tape contains documentation only. **Media (Service Charge Code):** 600' Magtape (MA). **Format:** FILES-11

DECUS PROGRAM ORDER NUMBER:V-SP-14 Title: Finger System: Network and Local Server, **Version:** V41.1.10, May 1985

Author: Dr. Richard Garland, Columbia University, New York, NY **Operating System:** VAX/VMS V4 or later, **Source Language:** VAX-11 FORTRAN, MACRO-32, **Keywords:** System Management - VMS, Networking

Abstract: This program serves three main functions:

- Identifies users of the systems, where they are, what program etc. For this function it serves as a personalized SHOW SYSTEM.

- Finds a specific user, gives the above information if he/she is logged on and in addition gives information about his/her mail and an optional information file he/she may supply.

- Serves as an in-bound DECnet server. Remote users can finger the local system as above and local users can finger remote systems that support the function. As part of the network "finger protocol" it performs explicit route-through. This is valuable in an internet situation such as going from DECnet to ARPAnet etc. This program can communicate with DECSYSTEM-20's and other VAXes running the program over DECnet, and through DECnet/ARPAnet gateways to ARPAnet hosts.

In a network situation where users are spread over many nodes and where there is large mail traffic, it can be an invaluable tool in finding people, ascertaining if they got your mail, etc.

Changes and Improvements: Updated for VMS V4.X

Documentation on magnetic media. Media (Service Charge Code): 600' Magtape (MC). **Format:** VMS/BACKUP (Blocked at 8192)

DECUS PROGRAM ORDER NUMBER:V-SP-30 Title: NOTIFY, **Version:** V1.1, April 1985

Author: T.J.F. Steele, Peter Steele & Partners, Solihull, W. Midlands, UK **Operating System:** VAX/VMS V3.7 **Source Language:** VAX-11 BASIC, **Keywords:** Mail

Abstract: The NOTIFY utility adds a new command verb to DCL. This sends a single line of text to another user on the system, without requiring any special privilege. For example, if user ALPHA were to type:

```
$ NOTIFY BRAVO "Seen CHARLIE lately?"
```

then user BRAVO would see:

```
*** From ALPHA: Seen CHARLIE lately?
```

on his terminal. By default, one bell is sent, but this may be increased or suppressed with the /BELL qualifier. The message usually starts on a new line, but this may be turned off with /NOCLRF. The utility also allows privileged users to broadcast to an ambiguously specified username of all logged-on users (*), suppress the "*** From ALPHA: " tag (/NOTAG), or suppress the automatic truncation to 57 characters (/NOTRUNC). Installation is very straightforward, using VMSINSTAL.

On line help is provided in the main library.

Changes and Improvements: Minor bugfix - protection of SYSSYSTEM:NOTIFY.EXE now set to W:E for world access; additional qualifiers /BELL, /CRLF, /TRUNC, /TAG, can not be included easily on the distribution tape with other programs. Sources not required for use as command verb - modification may compromise system security.

Release notes distributed with each order. Complete sources not included. Documentation on magnetic media. Media (Service Charge Code): 600' Magtape(MC). **Format:** VMS/BACKUP (Blocked at 8192)

DECUS PROGRAM ORDER NUMBER:VAX-99 Title: INDEX:FORTRAN Cross - Referencer and Flow Chart Generator, **Version:** V3.03, May 1985

Author: Michael N. Levine, Naval Weapons Center, China Lake, CA **Operating System:** VAX/VMS V3.7 or later, **Source Language:** MACRO-32, **Keywords:** Cross - ReferencersTools - Applications Development, Utilities - VMS

Abstract: INDEX is a FORTRAN source cross-referencing and flow charting utility that allows the user to look at individual source files (optionally saying on what lines and how they are used). Furthermore the user can select for display/save for SUPER INDEX only those variables or COMMON blocks with the characteristics that he is interested in-global/local, assigned value/not assigned value, used/unused, imported/exported, etc. in any combination. Also available is the optional ability to show up to three additional items of information for display during the regular and SUPER INDEX:

The variable storage location information (local, in COMMON, passed by argument, etc).

Usage in FUNCTION/SUBROUTINE calls (routine used in and argument number).

A user selected tag of up to 31 characters.

The data saved for a SUPER INDEX listing (consisting of 2 to 5 data items as outlined above) can be displayed with a great deal of flexibility as to the data item order and format (or saved in an ISAM data file for the user to work on directly). The resulting information supplied allows the user to follow the flow of data throughout a program or find the usage of any selected data variable as required.

If selected, the user may at the same time generate a flow chart of the source file currently being cross-referenced.

If wanted, the user can generate in place of the SUPER INDEX, an entry point cross reference

listing showing who calls who and is called by who (with optional graphical tree output).

Changes and Improvements: Bug fixes - made compatible with VMS V4 FORTRAN. **Restrictions:** Does not handle the new CDD "Dictionary" command.

Documentation on magnetic media. Media (Service Charge Code): 600' Magtape(MA) **Format:** VAX/ANSI (Blocked at 4096)

DECUS PROGRAM ORDER NUMBER:VAX-127 Title: AKCOUNT: A VMS System Accounting Package, **Version:** April 1985

Author: Kenneth J. Trumbley and Martin Serrer, National Research Council, Ottawa, Ontario, Canada **Submitted By:** Martin Serrer **Operating System:** VAX/VMS V4.1, **Source Language:** VAX-11 FORTRAN, MACRO-32, **Keywords:** System Accounting - VMS

Abstract: The AKCOUNT software has been designed to provide a VAX computer installation running V4.x of VMS operating system software with all the necessary accounting tools to change users for resources used. The package includes all the source code plus various command procedures as well as installation notes.

The software in SYSTEMS LABORATORY of NRC has been set up as a batch job to execute every Friday night. When the job runs, the information from the system accounting file, plus listing files from DISKQUOTA and AUTHORIZE are merged together and written to a file "SY\$ACCOUNT:AKCOUNT.TOT". A report generator reads this file and creates detailed or summary type printouts.

Changes and Improvements: Updated to work with VMS V4.1.

Documentation on magnetic media. Media (Service Charge Code): 600' Magtape (MA). **Format:** VMS/BACKUP (Blocked at 8192)

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
P.O. Box 53
Spring House, PA 19477
(215) 542-7008

BUSINESS APPLICATIONS

Thomas Byrne
L. Karp & Sons
1301 Estes
Elk Grove, IL 60007
(312) 593-5705

DATA MANAGEMENT SYSTEMS

Steve Pacheco
Ship Analytics
P.O. Box 410
North Stonington, CT 06359
(203) 535-3092

DAARC

Ellen Reilly
William H. Rorer
500 Virginia Drive
Ft. Washington, PA 19034
(215) 628-6547

GRAPHICS APPLICATION

Michael Anton
P.O. Box 591293
Houston, TX 77259-1293
(713) 928-4838

IAS

John Ross Roman
McDonnell Douglas
600 McDonnell Blvd.
Hazelwood, MO 63042
(314) 234-0984

APL

Doug Bohrer
Bohrer & Company
903 Ridge Road, Suite 3
Wilmette, IL 60091
(312) 251-9449

COMMERCIAL LANGUAGES

Jim Wilson
Pfizer Inc. QC Division
P.O. Box 88
Terre Haute, IN 47808
(812) 299-2121 x 271

DATATRIEVE

Joe H. Gallagher
Cleveland Clinic Foundation
9500 Euclid Avenue
Cleveland, OH 44106
(216) 444-2551

EDUSIG

Fred Bell
Taft College
Taft, CA

HMS

William Walker
Monsanto Research Corp.
P.O. Box 32 A-152
Miamisburg, OH 45342
(513) 865-3557

LANGUAGES & TOOLS

Alan Folsom Jr.
Fischer & Porter Company
E. County Line Road
Warminster, PA 18974
(215) 674-7154

LARGE SYSTEMS

Michael Joy
1st Church of Christ
Scientist
Boston, MA 02115
(617) 262-2300 x 3903

NETWORKS

Vicki Hancock
2510 Limestone Lane
Garland, TX 75040
(214) 495-7353

PERSONAL COMPUTER

Caroline Mack
6415 Adelphi Road
University Park, MD 20782
(301) 927-0108

RSX

Dominic DiNollo
Loral Electronics
Engineering Computer Center
Ridge Hill
Yonkers, NY 10710
(914) 968-2500 x2210

SITE MANAGEMENT & TRAINING

Gregory Brooks
Washington University
Behavior Research Lab.
1420 Graton St.
St. Louis, MO 63104
(314) 241-7600 x 257

VAX SYSTEMS

Larry Kilgallen
c/o DECUS Office
219 Boston Post Road. (BP02)
Marlboro, MA 01752

MUMPS

Janet Berryman
2405 N. Bush
Santa Ana, CA 92706
(714) 953-1025

OFFICE AUTOMATION

Margaret Drake
Univ. of TX Health Science Ctr.
7703 Floyd Curl Drive
San Antonio, TX 78284
(512) 691-6105

RSTS

Bill Hobbs
ComManD, Inc.
6535 E. 82nd St., Suite 102
Indianapolis, IN 46250
(317) 842-5320

RT

Bill Leroy
The Software House, Inc.
470 E. Paces Ferry Road Park
NE 1020
P.O. Box 52661
Atlanta, GA 30355
(404) 231-1484

UNISIG

William Toth
Harvard-Smithsonian Ctr. for
Astrophysics
60 Garden Street P353
Cambridge, MA 02138
(617) 495-7181

Bruce Bergman
UserWare International
2235 Meyer Avenue
Escondido, CA 92025-1070
(619) 741-8825

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	_____
Spring '85 Proceedings (SP5)	_____	12.00	_____
Fall '85 Proceedings (FA5)	_____	12.00	_____
Spring '86 Proceedings (SP6)	_____	12.00	_____
Fall '86 Proceedings (FA6)	_____	12.00	_____
TOTAL COST OF SUBSCRIPTION			\$ _____

MASTERCARD VISA _____ 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 Rd, Marlboro, MA 01752 (617) 480-3418.

SS860107

TYPE OF BUSINESS (ENVIRONMENT)/COMPUTER APPLICATIONS

Please check that which best describes your business/application

- | | | |
|--|---|---|
| 21 <input type="checkbox"/> ACCOUNTANCY | 1 <input type="checkbox"/> EDUCATION/PRIMARY | 73 <input type="checkbox"/> NUMERICAL CONTROL |
| 7 <input type="checkbox"/> BANK | 2 <input type="checkbox"/> EDUCATION/SECONDARY | 68 <input type="checkbox"/> OEM-COMMERCIAL |
| 64 <input type="checkbox"/> BUSINESS/COMMERCIAL | 61 <input type="checkbox"/> EDUCATION-TECHNOLOGY | 78 <input type="checkbox"/> OEM-TECHNICAL |
| 74 <input type="checkbox"/> BUSINESS/INFORMATION SYSTEMS | 3 <input type="checkbox"/> EDUCATION/UNIVERSITY | 56 <input type="checkbox"/> PHYSICAL SCIENCES |
| 57 <input type="checkbox"/> CHEMISTRY | 67 <input type="checkbox"/> ENGINEERING | 20 <input type="checkbox"/> RESEARCH/DEVELOPMENT |
| 54 <input type="checkbox"/> CLINICAL LABORATORY | 65 <input type="checkbox"/> FINANCE/ACCOUNTING | 10 <input type="checkbox"/> RETAIL |
| 63 <input type="checkbox"/> COMPUTATION | 77 <input type="checkbox"/> GOVERNMENT | 76 <input type="checkbox"/> SOFTWARE DEVELOPMENT |
| 11 <input type="checkbox"/> CONSUMER ELECTRONICS | 75 <input type="checkbox"/> GRAPHICS | 53 <input type="checkbox"/> TELECOMMUNICATIONS |
| 18 <input type="checkbox"/> CONSULTANT | 4 <input type="checkbox"/> HOSPITAL | 19 <input type="checkbox"/> TELEPHONE/UTILITIES |
| 72 <input type="checkbox"/> DATA ACQUISITION | 62 <input type="checkbox"/> INDUSTRIAL | 51 <input type="checkbox"/> TIMESHARING |
| 52 <input type="checkbox"/> DATA COMMUNICATIONS | 55 <input type="checkbox"/> LABORATORY/SCIENTIFIC | 80 <input type="checkbox"/> TRAINING/INSTRUCTION |
| 13 <input type="checkbox"/> DATA PROCESSING SERVICES | 14 <input type="checkbox"/> LIBRARY | 66 <input type="checkbox"/> TYPESETTING/PUBLICATION |
| 71 <input type="checkbox"/> DATA REDUCTION | 58 <input type="checkbox"/> LIFE SCIENCES | |
| 17 <input type="checkbox"/> DIGITAL EMPLOYEE-ENGINEERING | 70 <input type="checkbox"/> MANUFACTURING | |
| 15 <input type="checkbox"/> DIGITAL EMPLOYEE-MARKETING | 79 <input type="checkbox"/> MARKETING | |
| 16 <input type="checkbox"/> DIGITAL EMPLOYEE-SERVICE GROUP | 59 <input type="checkbox"/> MEDICAL RESEARCH | |
| 60 <input type="checkbox"/> EDUCATIONAL ADMINISTRATION | 6 <input type="checkbox"/> MILITARY INSTALLATION | |

SPECIAL INTEREST GROUP (SIGs) ENROLLMENT

I wish to participate in the following DECUS U.S. Chapter Special Interest Groups.

- | | | |
|---|--|---|
| 33 <input type="checkbox"/> APL SIG | 11 <input type="checkbox"/> HARDWARE AND MICRO | 36 <input type="checkbox"/> PERSONAL COMPUTER |
| 2 <input type="checkbox"/> COMMERCIAL LANGUAGES | 35 <input type="checkbox"/> IAS | 18 <input type="checkbox"/> RSTS/E |
| 6 <input type="checkbox"/> DATA MGMT.SYS. | 31 <input type="checkbox"/> DAARC(LABS) | 17 <input type="checkbox"/> RSX |
| 5 <input type="checkbox"/> DATATRIEVE | 27 <input type="checkbox"/> LARGE SYSTEMS | 19 <input type="checkbox"/> RT-11 |
| 7 <input type="checkbox"/> BUSINESS APPL. | 16 <input type="checkbox"/> LANG. AND TOOLS | 32 <input type="checkbox"/> SITE MGMT.& TRNG |
| 8 <input type="checkbox"/> EDUSIG | 14 <input type="checkbox"/> MUMPS | 21 <input type="checkbox"/> UNISIG |
| 10 <input type="checkbox"/> GRAPHICS APPL | 15 <input type="checkbox"/> NETWORKS | 26 <input type="checkbox"/> VAX SYSTEMS |
| | 34 <input type="checkbox"/> OFFICE AUTOMATION | |

JOB TITLE/POSITION - Please check:

- | | |
|---|---|
| 1 <input type="checkbox"/> CORPORATE STAFF | 101 <input type="checkbox"/> CORPORATE DIRECTOR OF DP/MIS |
| 2 <input type="checkbox"/> DIVISION OR DEPARTMENT STAFF | 102 <input type="checkbox"/> ADMINISTRATIVE ASSISTANT |
| 3 <input type="checkbox"/> SYSTEMS ANALYSIS | 103 <input type="checkbox"/> TECHNICAL ASSISTANT |
| 4 <input type="checkbox"/> APPLICATIONS PROGRAMMING | 104 <input type="checkbox"/> SERVICES COORDINATOR |
| 5 <input type="checkbox"/> SYSTEMS ANALYSIS/PROGRAMMING | 105 <input type="checkbox"/> MANAGER |
| 6 <input type="checkbox"/> OPERATING SYSTEM PROGRAMMING | 106 <input type="checkbox"/> ANALYST |
| 7 <input type="checkbox"/> DATABASE ADMINISTRATION | 107 <input type="checkbox"/> PROGRAMMER |
| 8 <input type="checkbox"/> DATA COMMUNICATIONS/TELECOMMUNICATIONS | 108 <input type="checkbox"/> DATABASE MANAGER |
| 9 <input type="checkbox"/> COMPUTER OPERATIONS | 109 <input type="checkbox"/> DATABASE ADMINISTRATOR |
| 10 <input type="checkbox"/> PRODUCTION CONTROL | 110 <input type="checkbox"/> 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

WE NEED YOUR COMMENTS!!!

In order to meet the needs of DECUS members who are interested in AI, we have prepared an AI SIC Survey And Questionnaire. Please answer the questions (it shouldn't take more than five minutes of your time) on the following two pages, then return the completed questionnaire to:

Don Rosenthal, AI SIC
Space Telescope Science Institute
Homewood Campus
Baltimore, MD.
21218

AI SIC SURVEY AND QUESTIONNAIRE

Name (optional) _____

Address (optional) _____

In which of the following areas of AI are you interested?

AI Languages and Development Environments ___ Engineering Applications ___

Expert Systems ___ Machine Learning ___ Medical Applications ___

Natural Language Understanding ___ Planning and Problem Solving ___

Robotics ___ Vision ___ Other _____

What is your current involvement with AI?

Present AI project ___ Present AI feasibility study ___

Future project (within 6 months to a year) ___

Professional development ___ General interest ___

Other _____

What would you like to see from an AI special interest group?

DECUS Symposia Sessions:

Technical ___ Management Oriented ___ Other ___

Examples of sessions you'd like to see given or repeated:

DECUS Presymposium Seminars:

Technical ___ Management Oriented ___ Other ___

Examples of seminars you'd like to see given or repeated:

Newsletter Editorial Content:

Book reviews ___ Case studies ___ Language, tool and product reviews ___
Q and A ___ Tutorial articles ___ Other _____

Communication with DEC:

Wishlist coordination ___ Arranging meetings with DEC personnel at DECUS ___
Other _____

Other SIG Activities: _____

What would you like to see from DEC in terms of AI?

Languages ___ Which? _____
Courses ___ What topics? _____
Products ___ What kind? _____
Other _____

Are you satisfied with user AI participation/representation at symposia? ___

If not, what changes would you suggest? _____

Are you satisfied with DEC AI participation/representation at symposia? ___

If not, what changes would you suggest? _____

Are you willing to volunteer for the AI SIC? ___ In what capacity?

Presenting or chairing AI sessions at DECUS symposia ___
Contributing to the newsletter ___ Working on the steering committee ___
Other: _____

Additional comments: _____

Thank you for completing this questionnaire. Your input is appreciated!

TO SUBMIT
REMOVE FORM AND
RETURN TO:

HMS Editor
DECUS
BP02
249 Northboro Road
Marlboro, MA 01752

HARDWARE SUBMISSION FORM
A SIG INFORMATION INTERCHANGE

MESSAGE _____

Contact
Name _____

Address _____

Telephone _____

Type of equipment: _____

SUBMIT ANY TYPE HARDWARE TROUBLES AND/OR FIXES.

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

DECUS PERSONAL COMPUTER SIG QUESTIONNAIRE

General:

I would like information on _____

I would like to see an article in the newsletter on _____

I would like to see a symposium session on _____

I am willing to be contacted by PC SIG members by telephone for expert assistance/advice on: _____

Phone number to call: Area Code (____) # _____

I attend DECUS Symposiums: ___always ___sometimes ___never

I use/own: ___Rainbow(s) ___PRO(s) ___DECmate(s)

I use the machine(s) checked above:

___at work ___at home ___both

If at work, total number of DEC PC's at your site: _____

I also use: ___VAX ___IBM or other mainframe ___IBM/other PC

Type of use: ___business ___educational ___government

Primary Operating System: ___MS-DOS ___CP/M ___both equally

___other _____

I belong to a local DEC PC User Group: ___yes ___no

There is a user group in my geographic area: ___yes ___no

I would like information on starting a user group: ___yes

I use a modem: ___often ___reluctantly ___never

___for work ___for pleasure ___both

Here is a DEC PC User Group not on your list:

Name of Group _____

Name of Contact Person _____

Address _____

Telephone (____) _____

Here is a DEC oriented bulletin board not on your list, or new information on a listed board:

Name of Board _____

Full name of Sysop _____

Address if known _____

City and State _____

Telephone Number _____

Other Info: _____

I am willing to write an article on: _____

The subjects of greatest interest to me are:

___word processing ___project management

___spreadsheets ___specialized vertical software

___graphics (type) _____

___communications ___Rainbow

___programming ___PRO

___software reviews ___DECmate

___technical articles ___Robin

___DEC Gossip and News ___Other: _____

If I had it to do over again, I:

___would buy another DEC Rainbow/PRO (circle one)

___might buy another Rainbow/PRO if it was a bargain (circle one)

___would not buy another Rainbow/PRO (circle one)

This newsletter is going to be folded into one large monthly publication (but will remain quarterly.) Will you continue to subscribe at the new price of \$35/year? ___yes ___no

Feel free to enclose another page(s) with comments!

Do you feel that leaving the prices out of the newsletter:

- is appropriate
- is very annoying
- makes the articles less useful

Do you feel that Decus should revise its (anti)commercialism policy?

- yes
- no

Name _____
Company _____
Address _____
City/ST/ZIP _____
Work Phone () _____
Home Phone () _____

Return to:
Caroline M. Mack
6415 Adelphi Road
University Park, MD
20782

fold here, flap under

stamp

Caroline M. Mack
6415 Adelphi Road
University Park, MD 20782

fold here, flap over

PRO 300 SERIES WISH LIST BALLOT

Use this ballot to show which items on the Wish List are most important to you. Put the number of the most important item on the list in space 1, the next most in space 2, etc.

1	11	21	31	41	51	61
2	12	22	32	42	52	62
3	13	23	33	43	53	63
4	14	24	34	44	54	64
5	15	25	35	45	55	
6	16	26	36	46	56	
7	17	27	37	47	57	
8	18	28	38	48	58	
9	19	29	39	49	59	
10	20	30	40	50	60	

Please add the following to the wish list:

Comments:

RETURN BALLOTS TO:

Thomas R. Hintz
University of Florida
IFAS Computer Network
Building 810
Gainesville, FL 32611

(fold here)

stamp

Thomas R. Hintz
University of Florida
IFAS Computer Network
Building 810
Gainesville, FL 32611

(fold here)

RSX Questionnaire

The following questionnaire is designed to provide information concerning the current user makeup of the RSX SIG and the future directions the SIG should take. The RSX Steering Committee will use the results of this questionnaire to ascertain how we can best meet the needs of the SIG membership in the future. Therefore, this is an important opportunity for you to help the SIG be more responsive to your needs by filling out a questionnaire and sending it to:

Elizabeth Bailey
222 CEB
Tennessee Valley Authority
Muscle Shoals, Ala., 35660

(Note: do NOT turn in another questionnaire if you turned one in at New Orleans). Thank you for your help.

How do you classify yourself?

- System Manager
- System Programmer
- Applications Programmer
- End User
- Sales/OEM
- Other (please explain) _____

What do you use your RSX system for? (Check all that apply).

- Real time
- Business applications
- Scientific applications
- Office automation
- Turnkey applications
- Program development
- Production work

What RSX systems do you have?

- RSX-11M
- RSX-11M Plus
- microRSX
- P/OS
- AME

What CPU's do you run RSX on?

What other operating systems do you use?

What layered products do you use with RSX?

- Networks
- Fortran
- SORT
- Datatrieve
- Basic
- Pascal
- FMS
- A to Z
- Cobol
- Dibol
- RPG

Is your RSX system part of a network?

- Yes
- No

Is your RSX system under maintenance?

- Yes
- No

How did you hear about DECUS?

What is your biggest complaint about RSX?

What do you like most about RSX?

How did you hear about the RSX SIG?

What do you expect of the RSX SIG?

List SIG activities in order of importance and evaluate how successfully

the RSX SIG is performing these activities.

What new services should the RSX SIG provide?

Why do you belong to DECUS?

How often do you attend DECUS symposia?

- Always
- Once a year
- Occasionally
- Once
- Never

If you do not attend symposia on a regular basis, why not?

How could you best justify attending a DECUS symposium?

What types of sessions at DECUS symposia are beneficial to you?

- Panels
- Q and A sessions
- Technical presentations
- Clinics/workshops

What specific topics would you like to see presented?

Do you subscribe to the Multi-Tasker?

- Yes
- No

RSX MULTITASKER

If you do not subscribe, do you have access to the Multi-Tasker?

Yes

No

What information do you want from the Multi-Tasker? (Please be relatively specific.)

What RSX-oriented pre-symposium seminars would you be interested in?

VAX SIG Steering Committee Nomination Form

(please print or type all except signature)

Nominee name _____
address _____

phone _____ DECUS # _____

The following five DECUS members sponsor the above nomination:
Name DECUS# date signature

- 1.
- 2.
- 3.
- 4.
- 5.

Please return form by 15 September 1985 to:
DECUS
VAX SIG Nominations
249 Northboro Road
Marlboro, MA 01752

PAGESWAPPER - September 1985 - Volume 7 Number 2
VAX SIG Steering Committee Nomination Form

Tear out to submit a nomination

DECUS
VAX SIG Nominations
249 Northboro Road
Marlboro, MA 01752

VAX Systems SIG Fall 1985 SIR Ballot

Questionnaire

DECUS membership number _____ (six digits)

My VAX experience level (check one):

Wizard _____ Expert _____ Knowledgeable _____
Normal _____ Novice _____

Our site uses the following VAX models (check all that apply)

8600 _____ 11/782 _____ 11/780,11/785 _____ 11/750 _____
11/730,11/725 _____ MicroVAX _____

Our site uses the following operating systems (check all that apply)

VAX/VMS _____ VAXelan _____ ULTRIX _____ UNIX (tm) _____
RSX-11 _____ RSTS _____ RT-11 _____ IAS _____
TOPS-10 _____ TOPS-20 _____

Other _____

We are currently running VMS Version _____ (if applicable)

We use VAX's in the following applications (Check all that apply)

Business EDP _____ Software Development _____
Education _____ Computer Science Research _____
Data Acquisition/Control _____ CAD/CAM _____
Service Bureau _____ Hardware Development _____
Scientific/Engineering _____ Office Automation _____
Telecommunications _____

Other _____

Our principal programming language is _____

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

PAGESWAPPER - September 1985 - Volume 7 Number 2
INPUT/OUTPUT Submission Form

Tear out 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 - September 1985 - Volume 7 Number 2
System Improvement Request Submission Form

Tear out to submit an SIR

Gary L. Grebus
Battelle Columbus Laboratories
Room 11-6011
505 King Avenue
Columbus, Ohio 43201-2693
USA

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	_____
Spring '85 Proceedings (SP5)	_____	12.00	_____
Fall '85 Proceedings (FA5)	_____	12.00	_____
Spring '86 Proceedings (SP6)	_____	12.00	_____
Fall '86 Proceedings (FA6)	_____	12.00	_____
TOTAL COST OF SUBSCRIPTION			\$ _____

MASTERCARD VISA _____ 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 Rd, Marlboro, MA 01752 (617) 480-3418.



Printed in the U.S.A.

“The Following are Trademarks of Digital Equipment Corporation”

ALL-IN-1	Digital logo	RSTS
DEC	EduSystem	RSX
DECnet	IAS	RT
DECmate	MASSBUS	UNIBUS
DECsystem-10	PDP	VAX
DECSYSTEM-20	PDT	VMS
DECUS	P/OS	VT
DECwriter	Professional	Work Processor
DIBOL	Rainbow	

Copyright ©DECUS and Digital Equipment Corporation 1985
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.

POLICY NOTICE TO ALL ATTENDEES OR CONTRIBUTORS “DECUS PRESENTATIONS, PUBLICATIONS, PROGRAMS, OR ANY OTHER PRODUCT WILL NOT CONTAIN TECHNICAL DATA/INFORMATION THAT IS PROPRIETARY, CLASSIFIED UNDER U.S. GOVERNED BY THE U.S. DEPARTMENT OF STATE’S INTERNATIONAL TRAFFIC IN ARMS REGULATIONS (ITAR).”

DECUS and Digital Equipment Corporation make no representation that in the interconnection of products in the manner described herein will not infringe on any existing or future patent rights nor do the descriptions contained herein imply the granting of licenses to utilize any software so described or to make, use or sell equipment constructed in accordance with these descriptions.

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.

Ada is a trademark of the U.S. Government, XEROX is a trademark of Xerox Corporation, IB M, PROFFS are trademarks of International Business Machines Corporation, UNIX is a trademark of AT&T Bell Laboratories, CP/M, PL/I are trademarks of Digital Research, Inc., MS-DOS is a trademark of Microsoft Corporation, TSX-PLUS is a trademark of S&H Computer Systems, Inc.





DECUS SUBSCRIPTION SERVICE
DIGITAL EQUIPMENT COMPUTER SOCIETY
219 BOSTON POST ROAD, (BP02)
MARLBORO, MA 01752

Bound Printed Matter
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 USA

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