DECUS U.S. CHAPTER
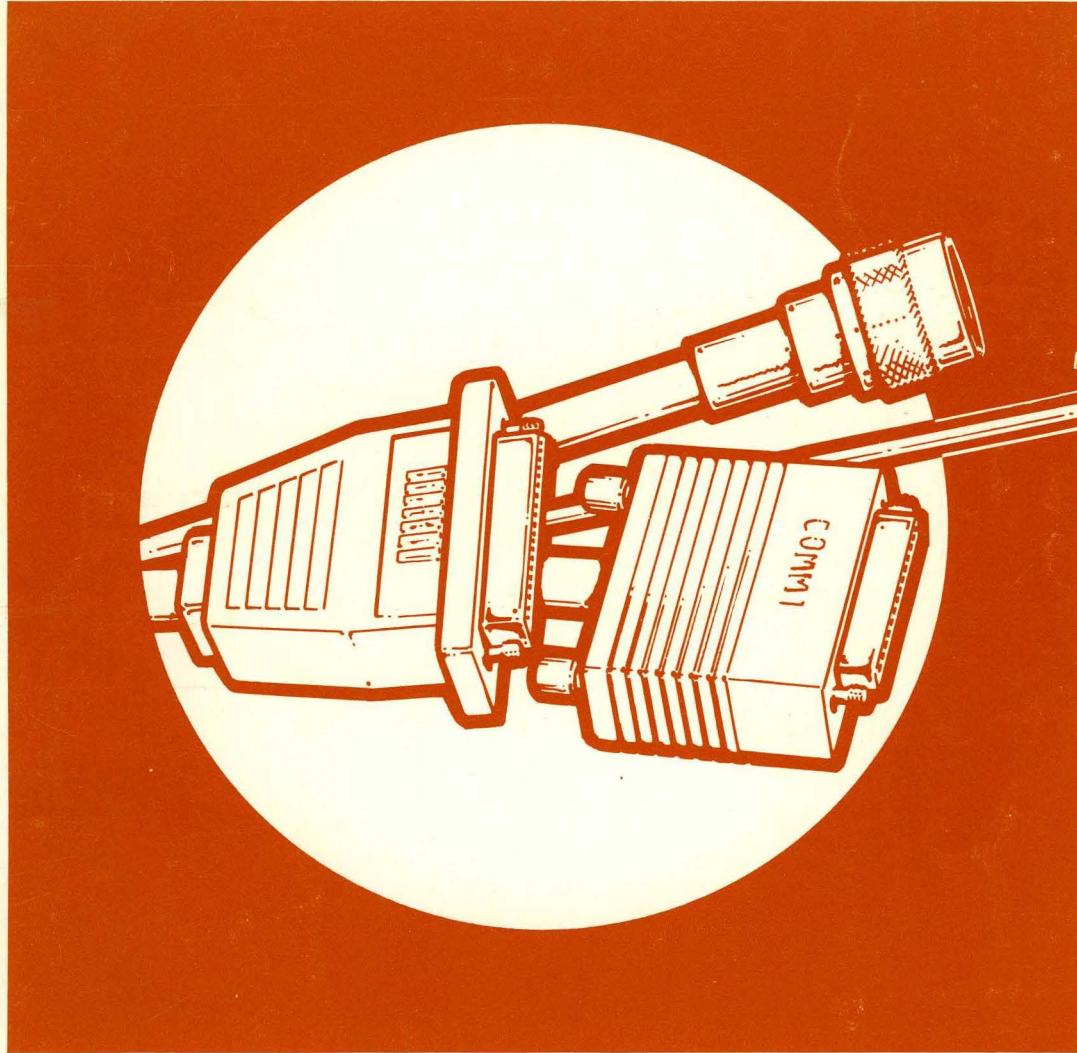
SIGs

NEWSLETTER

JANUARY

FEBRUARY

MARCH

APRIL

MAY

JUNE

JULY

AUGUST

SEPTEMBER

OCTOBER

NOVEMBER

DECEMBER

Your best connection for:

peer-to-peer interaction

technical exchange

VOLUME 4

NUMBER 2

DECUS

Printed in the U.S.A.

"The Following are Trademarks of Digital Equipment Corporation"

ALL-IN-1	FORTRAN	RSX
DATATRIEVE	HSC	RSX-11M-PLUS
DATATRIEVE-11	IAS (et.al)	RT-11
DECalc	LN03	RX02 (et.al)
DECdx	MACRO	SPM
DECmate	MicroVAX	TEAMDATA
DECmux	MUXserver	ULTRIX
DECnet	PDP-11	VAX
DECreporter	PDP-11/23 (et.al)	VAX DATATRIEVE
DECtalk	P/OS	VAXmate
DECUS	Pro-350	VAXstation
Digital	PRO	VAX/VMS
Eve	RALLY	VAX 8600 (et.al)
FMS	ReGIS	VMS
	RSTS	VT240
		WPS-PLUS

GENERAL TABLE OF CONTENTS

SUBSCRIPTION & MEMBERSHIP FORMS

Dear DECUS Member,

As you can see from my byline photo, I spent most of the last couple weeks in the Wisconsin woods, outside of even long-rangepager range from Chicago. I had enough time to sit back, drink a cup of coffee, and admire Mother Nature at her pace, (somewhat slower than normal big city speed.) While gazing at a large oak, I spent some time ruminating on the future of this newsletter.

A lot of people have spent a lot of time recently in committees and taskforces coming up with suggestions as to what to do with the SIG's newsletters. Many of the suggestions were avowed blue-sky approach. Some were obvious shots from the hip.

Some of the ideas were good, simple ones that could be easily done and gave immediate results... "Change the cover to something more exciting." Some sounded good at the beginning, but met with less than enthusiastic response from the troops in the trenches... "How about putting the editors' pictures on the section pages?" (Actually, I thought that was a nice idea, and I may even continue the practice.)

What I was contemplating in my mini woods meeting was the recently completed SIG Council Publications Task Force Report. I salute the members of the task force for a stellar achievement, that of actually following the dictate; "Do as I meant, not as I said." They were given a nebulous command: "Do something about low Newsletter subscriptions, and look at DECUS communications in general." They came up with a clear, simple proposal.

How they did it is also interesting. Instead of sitting around and batting ideas back and forth, they first attempted to really study the product. They did an analysis of back issues to analyze contents of the letter. They polled the SIG chairs and the Newsletter Editors. They analyzed the fall '87 readers' survey.

When they were done, they decided that there was nothing fundamentally wrong with the Newsletters, nothing could be done to substantially increase subscriptions by "tweaking" with the newsletters. The committee did come up with a far simpler, (in concept at least,) direct solution to a problem that cuts across all DECUS activities, not just the newsletter. A summary of their report follows:

GOAL: We believe that general DECUS communications must reach all interested DECUS members.
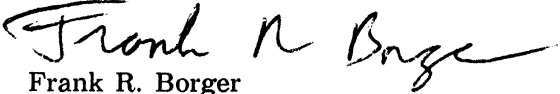
STRATEGY: To achieve this goal, we recommend that the SIGs Newsletters, DECUScope, Symposium Preliminary Program/SAG, Call for Participation, annual membership audit, library updates, seminar and other communications intended to reach all DECUS members be combined into a single monthly publication distributed to the audited membership at no charge.

It's simple in concept. Instead of the "DECUS U.S. Chapter SIGs Newsletter" just call it the "DECUS U.S. Newsletter". One publication a month instead of up to 6 per month for the office to worry about. At the reader's end, one mag a month with all the news about all the products. Us editors can finally forget about all the wailing and gnashing of teeth over "declining subscriptions." Maybe we could even include a catalog page, and sell DECUS and SIG stuff to members who aren't lucky enough to attend symposia. After a lot of woodsing, I think it can really work!

But the key to it all is that one word... AUDITED. DECUS is finally in the process of a full membership audit. (By the time you read this, you should have received your own audit.) If we include a yearly audit card that has to be filled out and sent back, so that we really are only talking to people who still have an active interest in DECUS, we solve a lot of problems. Our SIGs now know how many active members they have. Our members can update their profiles if their job changes. But if it is going to work, we must, (as Jim Sims suggested in a recent letter to the DECUScope editor,) "Ask The User."

Even the cost factor doesn't look too bad if our upcoming first audit trims our "real" membership by a factor of 2 or 3. Economics of printing and mailing really favor one missive a month, so the total cost of this idea would not be equal to $35 times the number of active members.

So I'm asking you, the DECUS user, what do you think? Drop me a line.

Frank R. Borger
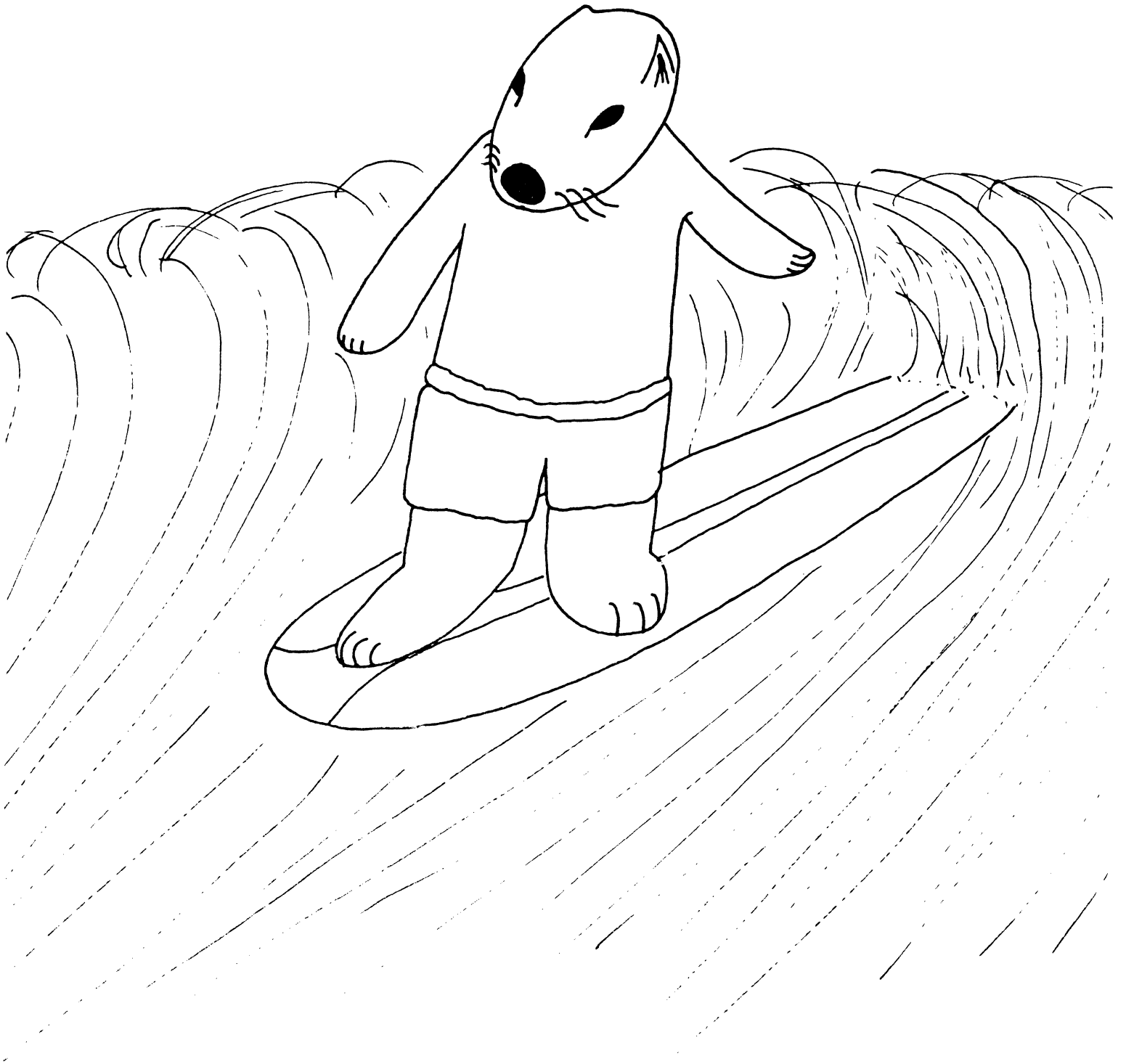
Contributions

This newsletter is a volunteer activity. There are no compensations given to any author or editor. Articles and letters for publication are encouraged from anyone. They may include helpful hints, inquiries to other users, reports on DECUS and SIG business, summaries of SPRs submitted to Digital, or any information of interest to users of either DATATRIEVE or 4th Generation Languages. However, this newsletter is not a forum for job and/or head hunting, nor is commercialism appropriate.

Machine readable input is highly desirable and machine-to-machine transfer of material is preferred, but most anything legible will be considered for publication.

Please send contributions, or for further information please contact either:

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

Joe H. Gallagher, Ph.D.
4GL Solutions
10308 Metcalf, Suite 109
Overland Park, KS 66212

Editorials and letters to the editor within the Wombat Examiner and 4GL Dispatch are solely the opinion of the author and do not necessarily reflect the views of the Digital Equipment Computer Users Society, Digital Equipment Corporation, or the author's employer. All editorials are marked as "An Editorial"; letters to the editor always begin "Dear Editor".

Table of Contents

DECUS U. S. Chapter

DTR/4GL Activities at the 1988 Fall Symposium in Anaheim

There are many activities sponsored by the DTR/4GL SIG which are not on the regular schedule. As soon as you get to Anaheim, check the weekend edition of the Update.daily, the Symposium newspaper, for any changes and check in the lobby of the Marriott Hotel for the room number of the DTR/4GL SIG Suite.

Sun 16th	9:00AM	(various)	Pre-symposium seminars
Sun 16th	5:00PM	SIG Suite	Volunteer's "sign-up" meeting
Sun 16th	6:00PM	SIG Suite	SIG Steering Committee Meeting
Sun 16th	9:00PM	Marriott	Welcoming Reception
Mon 17th	9:00AM	Northwest	DTR/4GL SIG Opening Session
Mon 17th	4:00PM	SIG Suite	RALLY/TEAMDATA Clinic
Mon 17th	6:00PM	SIG Suite	Working Group Chairs/Counterparts (C)
Mon 17th*	TBA	TBA	Application Factory Working Group
Tue 18th	12:30PM	TBA	Newsletter Staff Meeting (over lunch)
Tue 18th	4:00PM	SIG Suite	DATATRIEVE Clinic
Tue 18th*	TBA	TBA	FOCUS Working Group
Tue 18th	7:00PM		Night at Disneyland
Wed 19th*	TBA	TBA	RALLY Working Group
Wed 19th*	TBA	TBA	Oracle Working Group
Wed 19th	4:00PM	TBA	Accent-R Working Group
Thu 20th*	TBA	TBA	SmartStar Working Group
Thu 20th*	TBA	TBA	Powerhouse Clinic
Thu 20th*	TBA	TBA	Powerhouse Working Group
Thu 20th	7:00PM	Salon F	Wombat Magic
Thu 20th	9:00PM	SIG Suite	Reception
Fri 21st	3:00PM	Northwest	SIG Closing Session
Fri 21st	3:30PM	SIG Suite	SIG Steering Committee Meeting

*	Tentative schedule; check weekend edition of Update.daily
TBA	To Be Announced; time/place not finalized at this time; check week-end edition of Update.daily
(C)	An organizational meeting of Working Group Chairs, Working Group Vice-Chairs, Digital and non-Digital Counterparts, and the SIG Working Group Coordinator's SubCommittee. This meeting is not of interest to general Symposium attendees or other Steering Committee members.

Volunteers:

Enhance your enjoyment of the Anaheim Symposium by participating in volunteer SIG activities. Session chairs and suite hosts/hostesses are needed to assist with SIG activities. Volunteers receive an appreciation gift of a much sought after jacket! To participate, attend a drop-in meeting of volunteers between 5:00PM and 6:00PM on Sunday, October 16, in the DTR/4GL SIG Suite in the Marriott Hotel (check in the lobby

for the room number) or see Harry Miller, Volunteer Coordinator, at the Sunday evening Welcoming Reception (9:00 to 10:00). You may also contact Harry Miller by phone at 714-988-6481 extension 7798 at the Ontario California Police Department during the week before the Symposia if you would like to reserve a particular session to chair.

Session chairs have the best seat in the room – right up front! They introduce the speaker, control the question and answer session at the end of the talk, evaluate the presentation, enforce the DECUS commercialism policy, and assist the speaker with the lights and audio-visuals.

Suite hosts/hostesses welcome attendees, help direct attendees to Digital engineers and experiences users to get their questions answered, and make sure the computers don't sprout legs.

SIG volunteers will receive an appreciation gift of a jacket! The SIG will also send "thank you" letters to the volunteer's boss if the volunteer request it.

Seminars:

The DTR/4GL SIG is sponsoring three seminars in Anaheim on Sunday, October 16th. The seminars are

S10	Focus Reporting: Using Your FOCUS and Non–FOCUS Databases
S11	RALLY as a Programmer Productivity Tool
S12	Remote Data Access Alternatives

Although the deadline for pre-registration is passed, walk-ins may still register on a space-available basis at the registration area in Anaheim for a slightly higher fee. Seminars are still your best training value.

Steering Committee Meetings:

The DTR/4GL SIG Steering Committee will meet on Sunday, October 16th, at 6:00PM in the SIG Suite in the Marriott Hotel. Items to be covered at the meeting include:

- election of Seminars Rep, Library Rep, and Symposium Rep
- action on a new awards program
- reports of functional area representatives
- assignments to cover last-minute changes in sessions
- other matters that may arise.

The Steering Committee will also meet in the Suite at 3:30PM on Friday, October 21st, immediately following the Closing Session to make plans for the Atlanta Symposium. All those interested in participating in the SIG leadership activities are requested to attend this important meeting.

DTR/4GL SIG Suite:

The DTR/4GL SIG will have a suite in the Marriott Hotel. The suite is where attendees can relax a little, have a soft drink, meet other attendees interested in DATATRIEVE and 4GLs, talk with expert users and Digital engineers, work on problems or test solutions on a microVAX, check technical issues in the latest documentation (VAX-DATATRIEVE, RALLY, TEAMDATA, and DECReporter) or in one of the back issues of the Wombat Examiner and 4GL Dispatch, submit a product improvement request (PIR), work on your presentation for Wombat Magic or the two contest problems, or just pick up a souvenir button of the Anaheim Symposium. The Suite will be open

Monday	9:30AM–4:00PM
Tuesday	9:00AM–4:00PM
Wednesday	9:00AM–4:00PM
Thursday	9:00AM–5:00PM
Friday	9:00AM–3:00PM

Clinics:

DATATRIEVE, RALLY/TEAMDATA, and Powerhouse Clinics have been scheduled so that users may get their technical questions answered. Bring your problems (with printouts if available) to get help from experts! The consulting and problems solving at the clinics could pay for your trip to DECUS Symposia.

Working Group Meetings:

Working Group Meetings for Accent-R, Application Factory, FOCUS, Oracle, Powerhouse, RALLY, and SmartStar are planned for Anaheim; check the Update.daily or the Wombat Sessions-at-a-Glance for the exact time and place of each working group meeting. Attendees who are interested in one or more of these products should attend the Working Group Meeting to influence the future activities of these groups.

Accentuating ACCENT:

The working group has plans for Anaheim. We are expecting a status report on the Bulletin Board project. Our Working Group editor will be working with SIG Newsletter editor to publish this report in the SIG Newsletter. And we might have participants from Europe! We will also have some discussion about the agenda of a proposed one day Accent meeting at a future date. There will be a status report on the progress of the development of the user list. See you at the Accent-R Working Group meeting.

In FOCUS:

A working group meeting will be held at the symposium. In addition, a pre-symposium seminar on "FOCUS Reporting: Using Your FOCUS and Non-FOCUS Databases" will be presented. We expect IBI will be supplying a MicroVAX 2000 (at best) or a VAXMate (at worst) for demonstrations of FOCUS in the SIG Suite.

RALLY:

DEC has announced RALLY V2.0; come to the RALLY Clinic and Working Group meeting to learn about the new features. The RALLY Working Group is the forum for those interested in the fourth generation language (4GL) RALLY to meet and discuss topics of mutual concern. Both problems and expertise are welcome.

Wombat Magic, Problem Contests, and Reception:

At a Symposia when its Thursday night at 7:00PM, its Wombat Magic time! Expect some clever and unusual DATATRIEVE Magic, a few humorous stories, prizes and drawings, a visit from the Great Wombat, and much more! This time in Anaheim in addition to the regular contest for the best magic, there are two special contests for the best solution to the "Sorted Extremum of a Class" problem and the "DATATRIEVE-11 date format conversion" problem. See the September issue of the Newsletter for specification of the problems and contest details. After Wombat Magic at 9:00PM, there will be a reception in the SIG Suite for all SIG members and attendees interested in DATATRIEVE and 4GLs.

Wombat Magic, Spring 1988 – Part 3

Session Chair: Dick Azzi, Motorola, Phoenix, AZ
Session Editor: Joe H. Gallagher, Ph. D., 4GL Solutions, Overland Park, KS

Editor's note: The following is Part 3 of a highly edited transcription of the audio tape of the Wombat Magic Session at the 1988 Spring DECUS Symposium in Cincinnati, Ohio, which occurred on May 19, 1988. Parts 1 and 2 appeared in two previous month's issue. Material which was presented on transparencies has been merged into the oral presentation. An attempt has been made to convey both the technical content of the Magic Session as well as the humor, covert intellectual swaggering, and the spirited interchange of the presentations. Material which appears in the text with square brackets [] has been added by the editor in an attempt to improved the understandability of this very exciting Magic Session. Special acknowledgment is given to Mary Gallagher who assisted with the transcription of the audio tapes.

Richard Copeland, Corning Glass Works, Corning, NY

We had a situation awhile ago where someone wanted to make-up a date string and center it at the top of every report page. Ok, so they do something like this and declare a date field, DAY; prompt the user to enter the report

date; then declare a field called DT; and give it a real long format string, format DAY using the day of the week, the month of the year, the day, and the four digit year. You get something like DT equals "Monday, January 10, 1988", with some spaces in there. This works out very well, but of course this is a variable length string. Some day names are longer than others; some month names are longer than others; it varies left and right. Now, how are you going to get this centered on top of a report? You have a variable, you want it centered. You can not; DATATRIEVE centers it automatically if you put hard coded text into your SET REPORT_NAME equals something, it centers it. The other way you can try is to take control yourself and print out the DT string at column 10. But again that 10 is a hard coded number not a variable. If you try to put a variable in there [COL variable] it won't let you. So how to get that centered with a variable? Well ... Here's a procedure I put together to demonstrate how to do this.

```
DEFINE PROCEDURE DATED_REPORT
! a procedure for a centered report date
DECLARE DAY USAGE IS DATE.
DECLARE DT PIC IS X(80).
DECLARE DX COMPUTED BY CHOICE OF
    FN$DAY(DAY) LT 10 THEN FORMAT DAY USING W(9)||","|||
        FORMAT DAY USING M(8)|||FORMAT DAY USING D,BYYYY
    ELSE FORMAT DAY USING W(9)||","|||
        FORMAT DAY USING M(8)|||FORMAT DAY USING DD,BYYYY
END_CHOICE.
    DAY = *."report date"
    DT = DX
!
DECLARE CENTER COMPUTED BY 40 - ((FN$STR_LOC(DT||"~","~") -1)/2).
DECLARE CENTERED_DATE PIC IS X(80).
    QUERY_HEADER IS -.
    CENTERED_DATE = DT
    REPEAT CENTER BEGIN
        CENTERED_DATE = " "|CENTERED_DATE
    END
READY domain
    REPORT domain WITH date-field = DAY
        AT TOP OF PAGE PRINT COL 31, "Centered Report For", SKIP,
            CENTERED_DATE, SKIP, COLUMN_HEADER
        . . .
    END_REPORT
END-PROCEDURE
```

It declares a variable called CENTER COMPUTED BY assuming an 80 column wide screen. The number of spaces you have to insert in front of this to make it centered is 40 minus half the length of the string. You take DT; append a tilde or some other character that's not used [within the string] to the end of DT; so the tilde appears out here; search for the tilde and it tells you how long that string is. You then subtract; find that string location; subtract one and divided by two; subtract that from 40. That tells you the number of spaces you have to move that string over to get it centered. All right, now you declare CENTERED_DATE as PIC X(80); you set the CENTERED_DATE equal to this long format string and then repeat CENTER, repeat this number of times, CENTERED_DATE equals a space, vertical bar with CENTERED_DATE and that prepends that many spaces onto that text string. You can then report your domain with date fields equals DAY centered up here and AT TOP OF PAGE PRINT "Center Report For", SKIP, CENTERED_DATE, SKIP, COLUMN_HEADER, ... END_REPORT, and it comes out a centered report for Tuesday, December 8, 1987, or whatever that date works out to be.

```
                      Centered Report For
                   Tuesday, December 8, 1987
        _____        _____        _____        _____        _____
```

We've also used this in other situations. Say you will have someone who's searching for report on a customer name and the customer's name is a variable length field as well, it prompts 'em [for the] customer name you would like to look up; it finds the report for that customer name; centers up that report and prints it out. I'm still researching how to get this to print centered AT TOP OF CUSTOMER_NAME, but this has solved about 90% of our needs and works well for us. [Applause]

Lew Lasher, Digital Equipment Corporation, Nashua, NH

John Henning said I had to do this to get some RALLY Magic here. I also came here late, but I don't think there's any other RALLY Magic so you can't throw anything at me for that reason. People have opaque things. I see everybody covering up part of their slides. You have special opaque things for that? (Yes, it's called paper.) OK, well I brought some with me here. Opacity is one of our most important skills. RALLY Magic – windows – windowing here. We do windows. OK, like everything in RALLY starts with a FORM/REPORT and the magic is the user hits the Help key and then you get a HELP MENU with various things here, help on this, help on that, more help, still more help and help yourself.

```
+-----------------+              +-----------------------+
|                 |              |         MENU          |
|      FORM       |    Help      |                       |
|                 |              | 1.  Help on this      |
|                 |  ------->    | 2.  Help on that      |
|     REPORT      |              | 3.  More help         |
|                 |    Key       | 4.  Still more help   |
|                 |              | 5.  Help yourself     |
|                 |              |                       |
|                 |              |                       |
+-----------------+              +-----------------------+
```
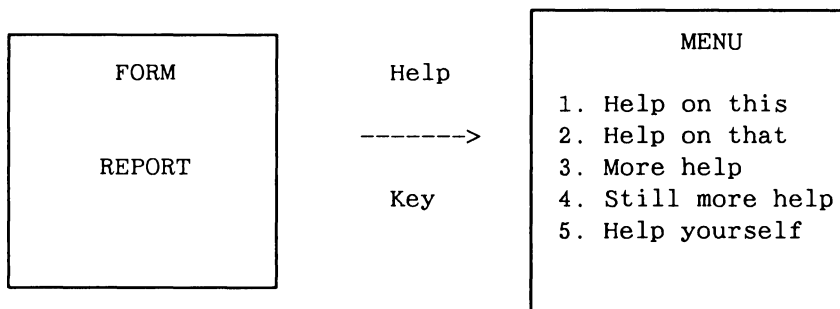
[For those of you who are] half–way familiar with this technique of obscuring things, DEC teaches other techniques of obscuring things. OK. Now I got to tell you how to do it here. Start with the formal report packet and you put in the Help number, just to make it more concrete than it already is. I put in a sample Help number, in this case I put in 500 and then the second step is you create the menu. You actually have to put in the menu choices that calls the Help messages, but those are just one line things that say help this, help that, help other things. Then we get to take the obscure RALLY feature which makes this RALLY Magic, which is create Library AFILE entry whatever that is, and we have (as usual in RALLY) an obscure long menu path, in this case 55431, that's how you create this thing and you just say entry 500 is the menu and there you have it, a HELP MENU. That's it.

Andy Schneider, Digital Equipment Corporation, Nashua, NH

I work with the VAX–DTR Development Team, formerly from the DTR–11 Development Team, and I learned a neat trick a few years ago. I presented it at Magic a few years ago, but since they're so many new timers I figured you would get a kick out of it and I spiced it up a little. The title of this is "How To Lose Your Boss", kind of like how Bert [Roseberry] approaches things, how to fake you boss out, only I figured how to lose your boss. First thing you do is go out to your local Digital sales person and you say "Give me a high speed line printer, the fastest one you got." So you go buy that. Then you go to a junk yard and you steal an old PDP–11/70 or go to Bedford, some of the old DEC facilities, [and take] the ones they've been training off of for years. Next you install either RSX, or RSTS, or heaven forbid, IAS, or TSX, no, no, it doesn't work on TSX, sorry. ... Then you install DTR–11, that fantastic product. Then you go get your boss; your already for it now. You ask him to burst the line printer listings. While he's in the paper path, you know in front of the printer, enter the DTR command AT TOP OF PAGE, PRINT NEW_PAGE. For those of you who have not experienced this, it kind of launches the paper across the room. The final step in all of this is pack your office while your boss crawls out from under the pile of paper. [Laughter]

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

Some of the best magic is the simplest magic. This was presented by the former SIG Chair, my predecessor, Larry Jasmann, who is with the U. S. Coast Guard. [Suppose an] application requires you to go through and change all the file names on the domains because some logical was pointed somewhere else and you wanted to move them somewhere else and you have 450 domain definitions and you need to change them. So all you do is say "EDIT ALL". It pulls the entire contents of your dictionary into the editor buffer and you do a mass substitution, changing everything you want all at one fell swoop. Exit from the editor and it changes everything. The simplest magic is the best magic.

Sue Hall, Rhodes College, Memphis, TN

I think I'm going to have to hold this [the lavaliere microphone] because I don't thing I've got anything to clip it too. [Laughter] I work for Rhodes College in Memphis, which some of you may know is kind of down in the

corner of the State of Tennessee. We border on Alabama, Arkansas, and Mississippi and even Missouri is not too far away. First thing I'd like to show you is how to clutter up your dictionary. [Laughter] Our alumni office has this notion that they have a hundred cities that alumni might live in. Now some of 'em are perhaps uncooperative enough to live in some other place, but most of them tend to live in these places. Further they decided, well you know Memphis has certain ZIPs. Arkansas just outside Memphis has certain ZIPs, across in Mississippi there are certain ZIPs and then they throw in a few others. So another thing you have to remember how to type all these. You know, they get lazy; they don't want to look it up. We defined a procedure called GREATER_MEMPHIS and we just put a little–biddy piece of code in it, but mostly it has those darn zip code numbers,

```
DEFINE PROCEDURE GREATER_MEMPHIS
    ZIP BT 38100 AND 38138 OR
    ZIP BT 39001 AND 39010 OR
    ZIP BT 72600 AND 72619
END_PROCEDURE
```

so when they get ready to find the collection they just say

```
FIND PEOPLE IN ALUMNI WITH :GREATER_MEMPHIS
```

(or the procedure for some other city) and then they can add on whatever other constraints they want, but boy when I go looking for a procedure in their dictionary it takes me a while to find what I want cause there's a hundred cities in there.

Now another thing I have, is that we have students that live in the dorm, in fact about 90% of our students live in the dorm. They have to go pay a room deposit in the cashier's office and then they go up to the Dean of Students office and they register.

```
DEFINE DOMAIN ROOMS ...
. . .
05 DORM PIC X(n).
05 ROOM_NUM PIC 9(m).
05 SS_NUM PIC 9(9).
...;
DEFINE DOMAIN STUDENTS ...
. . .
05 SS_NUMBER PIC 9(9).
05 NAME ...
...;
```

Well, the Dean of Students goes in and he takes his domain that is keyed by room, cause its dorm and room [that they are interested in] and he puts [in] the students social security number and in the meantime the cashier's office has been recording the fact that they have a deposit on this student, and they've got a code for what kind of room it is so they know what to charge him and then they come along in September and they said, well the Dean of Students office has 900 students and I've got 897. How am I going to find out who is in the dorm according to the Dean of Students but is not in the cashier's file? Or sometimes its vice versus. They say we're gonna print out a list, each of us, and compare them name by name. I said no you don't want to do that. And this does it,

```
FOR STUDENTS
    IF NOT ANY ROOMS WITH SS_NUM = SS_NUMBER THEN
        PRINT SS_NUMBER, NAME
```

You can turn it from either direction; it gets a little more complicated if you consider the fact that most of the rooms have at least two occupants, so that there's a list. You get a little more complicated but I can't remember on the fly where you stick the ANY for the hierarchy. [Applause]

Larry D. Roduner, Tremco, Inc., Beachwood, OH

I don't know if any of you have run into this problem but I do. I have some users who like to have columns, solid lines in the middle of their report. These usually happen with variable length lists, which doesn't help any. The way I have done this is to define a variable Z with a PIC X(5), and put in escape sequences to control the printer; this is coming out on a LN03 printer [or other printer supporting eight–bit characters]. So what I've done is, since the solid line ["|"] always leaves a gap between lines, to print a solid line, backed up with the back space character, do a partial line down, print the solid line again and then do a partial line up. And then print the rest of the line. The value of these special characters in octal and decimal are

Character	Description	Octal	Decimal
<BS>	back space	(010)	008
<PLD>	partial line down	(213)	139
<PLU>	partial line up	(214)	140

You have to set the columns per page to a large number because there will be many characters which will give you only one printed space. A report procedure would look like:

```
DECLARE Z PIC X(5).
    Z = "|<BS><PLD>|<PLU>"
REPORT domain
    SET COLUMNS-PAGE = 1000
    . . .
    PRINT . . ., COL N, field1|Z|field2|Z|field3 ... |Z
    . . .
END-REPORT
```

Alan H. Beer, Compusult, San Jose, CA

I'm sure that most people in here frequently find themselves working with databases. And validation can always be something of a nuisance. So you've got some sort of interactive procedure where people have to enter a social security number and then you get some sort of screen presented that's got all the student information that goes with it. Well, if you want to validate the social security number, rather than have the screen procedure chock when it gets one that it doesn't find, its real simple. You create a domain table from the student master list.

```
DEFINE TABLE SSNO_TABLE FROM STUDENT_MASTER USING
STUDENT_SSNO : STUDENT_NAME
END_TABLE
```

The trick is we're never going to use the second field – we're only going to use the first. So when it comes time to run something, we can build ourselves a little loop to keep reporting student after student in the form

```
WHILE 1 EQ 1 BEGIN
    IF *."SS Number" IN SSNO_TABLE THEN BEGIN
        :STUDENT_SCREEN_PROCEDURE
    END
END
```

If the social security number is not in the SSNO_TABLE it will prompt for the next number. When you want to get out of the procedure, you hit CONTROL Z or CONTROL C. [Applause]

Awards

[Chris Wool announced the awards.] We have a some of honorable mention awards. Our honorable mention awards go to Alan Beer for converting dates from the off brand format to the standard DATATRIEVE format and the comment searcher, to Ray Ferrara and Doug Cropper for creating the FMS forms and the Wombat cursor, and Ann Kah for making DATATRIEVE-11 actually work.

For the pen and pencil sets, Joe Mei for the elapsed week days and business days functions and Bart Lederman for vector plots.

The top prize, a VAX DATATRIEVE doc set, goes to Richard Copeland for his centered printing.

Evelyn Garcia announced the dishonorable mention awards to Dana Schwartz and Bert Roseberry for their top 10 lists.

Bert Roseberry, U. S. Coast Guard, Washington, DC

Although Bert did not present his bit of plotting magic at the session in Cincinnati, he did turn in his transparencies. Create a file, whose name is ENSIGN.UPD, which contains the following

```
DELETE ENSIGN;
REDEFINE PLOT ENSIGN
-31,31
PRINT "p[116,700]w(sOri3)t(AOiOs2)'Coast Guard Wombat En-
sign't(sl)s(a[0,0][767,479])p[,500]"
PRINT "p[280,150]t(S2,w(r))'Semper'"
PRINT "p[280,180]t(S2,w(r))'Paratus'"
PRINT "p[176,260]v(w(ri2))[250,280]"
PRINT "p[176,261]v(w(ri2))[250,281]"
PRINT "p[176,262]v(w(ri2))[250,282]"
PRINT "p[176,263]v(w(ri2))[250,283]"
PRINT "p[176,264]v(w(ri2))[250,284]"
PRINT "p[176,265]v(w(ri2))[250,285]"
PRINT "p[176,266]v(w(ri2))[250,286]"
PRINT "p[176,267]v(w(ri2))[250,287]"
PRINT "p[176,268]v(w(ri2))[250,288]"
PRINT "p[176,269]v(w(ri2))[250,289]"
PRINT "p[176,270]v(w(ri2))[250,290]"
PRINT "p[176,271]v(w(ri2))[250,291]"
/
```

Then, in DATATRIEVE extract the WOMBAT plot by

```
DTR> set dictionary cdd$top.dtr$lib.plots
DTR> extract on wombat.plt wombat
```

Then, create a new plot with the SUMSLP batch editor with the DCL command

```
$ EDIT/SUMSLP/OUTPUT=ENSIGN.PLT WOMBAT.PLT/UPDATE=ENSIGN.UPD
```

Then, back in DATATRIEVE, insert the new plot and plot it by:

```
DTR> set dictionary cdd$top.dtr$lib.plots
DTR> @ensign.plt
DTR> plot ensign
```

which gives

Parsing Text Using DATATRIEVE

Leonard E. Herzmark, P. E., Maricopa County Health Dept., Phoenix, AZ

Here at the Environmental Services of Maricopa County Health Department (Phoenix and vicinity) we monitor the air quality through a number of remote stations. When I say remote, I mean it. The county, part of which is miles and miles of nothing but miles and miles, encompasses an area somewhat larger than the state of Massachusetts. It's a long drive between some stations. The pollutants monitored are ozone, carbon monoxide, oxides of nitrogen and sulfur dioxide. In addition to these, we monitor other parameters including wind speed and direction, inversion intensity, and air conditioner cooling function in the instrument rooms. Particulates (as total) and particulates (10 microns and less) are collected in devices reminiscent of a vacuum cleaner and analyzed for various chemical components (these samples are picked up manually after a twenty–four hour run of the collection machines). The chemist does his determinations gravimetrically for total particulates and total extractable organic matter and uses a visible spectrophotometer to determine sulfates and an atomic adsorption spectrophotometer for metals, including lead, copper, iron, nickel and manganese. It should be of interest to note that over the last decade we have seen a drop in lead in the air by an order of magnitude. Sulfur dioxide is low, generally below the level of detectability. Just about everything else has gotten much worse.

We wanted to drive several DECTALK (synthesized voice) phone lines for the media and public on which we would report the air pollution on a continual basis. Our original data logging system was purchased as a package when we acquired our VAX, and the software was all written in FORTRAN. No one here is FORTRAN literate. We are all engineers and scientists not professional programmers; we promptly forgot anything we learned in computer science as soon as the grades came out. Since I wrote almost everything in use here, in DATATRIEVE, I was called upon to see what could be done.

First of all, let me tell you that the output of the acquisition program is an hourly report in the form of a .LST file. The format of each succeeding .LST file is exactly the same, only the date, hour, and data for each parameter changes. The files do not have a standard DATATRIEVE RMS data file structure. A typical hourly report looks like:

```
1-HOUR  AVERAGE  REPORT
DATE:  03/23/87                     HOUR:  12:00
SITE      DCN  +  WSPD +  WDIR +  OZON +  CO   +  DELT
          ID      (MPH)   (DEGA)  (PPB)   (PPM)   (DEGT)

          ------------------------------------------------
CPHX      A       3.2     245.    60.9    4.1     -0.3
SPHX      B       1.4     233.    52.3    5.0
WISR      C       1.7     222.    40.1    5.1     0.1
WPHX      E       3.5     200.    37.1    4.1
```

Looking at the print out of this report, I pictured a series of fields which contained the information needed for the DATATRIEVE (RMS) files. I should point out that for ease of explanation, the original reports and files have been simplified to remove several fields which were not really needed to illustrate the point of this article.

By the way, SITE is the location of the monitor (in this example CPHX is Central Phoenix), WSPD is the wind speed, WDIR its direction, OZON is ozone, CO is carbon monoxide, and DELT is the temperature difference between two sensors located at different elevations on the same tower. A positive value in the DELT field indicates that there is a temperature inversion; that is, the upper sensor has a higher temperature than the lower one.

During the initial attempts to read the data, DATATRIEVE advised that the record was defined as 55 bytes but the 'data' file was 255 bytes long. When printing out the data through DATATRIEVE, the data was also shifted to the right by one character; it was clear that there was an extra byte at the beginning of the record which was not visible when the files were directly printed. Analyzing the data file by

```
$ ANALYZE/RMS AVG1HR01.LST

    . . .

RMS FILE ATTRIBUTES
    File Organization: sequential
    Record Format: variable
    Record Attributes:    fortran
```

```
Maximum Record Size: 255
Longest Record: 58
        . . .
```

showed that the data file has FORTRAN record attributes and thus needs a FILLER field at the beginning of the record for the FORTRAN carriage control character. The maximum record size is 255 bytes so a FILLER field was added to the end with a PIC X(199). The record size of 55 bytes, 1 byte FILLER at the beginning, and a 199 byte FILLER at the end make a total of 255 bytes. The longest record is actually 58 bytes since the line of minus signs between the header and the data actually extends three characters past the end of the data.

The mapping of the record definition to the data looks like FIGURE 1.

1–HOUR AVERAGE REPORT

```
         DATE:  03/23/87                          HOUR   13:00
 SITE    DCH  +   WSPD  +  WDIR  +  OZON  +   CO   +   DELT
          ID      (MPH)    (DEGA)   (PPB)    (PPM)     (DEGT)
 ------------------------------------------------------------------
 CPHX    A        3.2      233.     63.8      2.0      -2.2
 SPHX    B        2.7      266.     44.6      0.4       0.6
 WISR    C        5.4      237.     42.6      1.1       0.6
 WPHX    E        4.6      247.     40.8      0.6


 RECORD   LEH_REC
 01       AVG1HR01.
 05       FILLER     PIC X.
 05       SITE       PIC X(4).
 05       FILLER     PIC X(6).
 05       DATE1      PIC X(8).
 05       WSPD       PIC X(4).
 05       FILLER     PIC X(4).
 05       WDIR       PIC X(4).
 05       FILLER     PIC X(4).
 05       OZON       PIC X(4).
 05       FILLER     PIC X(4).
 05       COHR       PIC X(5).
 05       SEPERATE_CO REDEFINES COHR.
 10       C0         PIC X(4).
 10       FILLER     PIC X.
 05       SEPERATE_HOUR REDEFINES COHR.
 10       FILLER PIC XXX.
 10       HR         PIC XX.
 05       FILLER     PIC XXX.
 05       DELT       PIC X(5).
 05       FILLER     PIC X(199).
 ;
         DEFINE DOMAIN LEH USING LEH_RECORD ON DISK$DATA:AVG1HR01.LST;
```

Figure 1

The data and record definition above shows how the fields were broken out based on the incoming report structure. The fields were defined to pick up the data, beginning with a 4 byte field to get the site, then a 6 byte filler, followed by the 8 byte date field, etc. I have attempted, by means of shading to show each field as I envisioned them, one from the other. Notice that fields that have data are named, whereas those that just take up space are

the 'filler'. The domain LEH was defined using AVG1HR01.LST as the 'data' file, and the record was built to get the proper spacing. Since there was, in one case, an overlap between two of the fields, the covering field COHR was first defined and then redefined using the 'REDEFINES' clause, broken apart so that CO and HR could be read separately.

Experimenting with various record select expressions gave a feel for what would be necessary to read particular records of the data file. It turns out that DATATRIEVE looks at each line of the file as being a separate record. The line with date and hour is the one where the field CO contains a ":" (or you could check for DATE1 containing a "/"). The data lines can be gotten by checking for the SITE name.

A procedure to read the data and store it in a "real" DATATRIEVE domain is:

```
DEFINE PROCEDURE CONVERT_DATA
READY LEH SHARED
READY HERTZ SHARED WRITE
DECLARE VDATE USAGE DATE.
DECLARE VHR PIC XX.
    FOR LEH BEGIN
        IF (CO CONT ":") THEN BEGIN
            VDATE = DATE1
            VHR   = HR
        END
        IF (SITE="CPHX","SPHX","WISR","SPHX") THEN BEGIN
            STORE HERTZ USING BEGIN
                DATE1 = VDATE
                HR    = VHR
                SITE  = SITE
                WSPD  = WSPD
                WDIR  = WDIR
                OZON  = OZON
                CO    = CO
                DELT  = DELT
            END
        END
    END
END_PROCEDURE
```

where the domain and record definition for HERTZ is

```
DEFINE DOMAIN HERTZ USING HERTZ_RECORD ON DISK$DATA:HERTZ.DAT;
DEFINE RECORD HERTZ_RECORD
01 HERTZ_REC.
    05 SITE       PIC X(4).
    05 DATE1      USAGE DATE.
    05 WSPD       PIC X(4).
    05 WDIR       PIC X(4).
    05 OZON       PIC X(4).
    05 CO         PIC X(4).
    05 HR         PIC XX.
    05 DELT       PIC X(5).
;
```

The print out of the final, desired data in the record is

SITE	DATE1	WSPD	WDIR	OZON	CO	HR	DELT
CPHX	23−Mar−1987	3.2	245.	60.9	4.1	12	−0.3
SPHX	23−Mar−1987	1.4	233.	52.3	5.0	12	
WISR	23−Mar−1987	1.7	222.	40.1	5.1	12	0.1
WPHX	23−Mar−1987	3.5	200.	37.1	4.1	12	

From here, the data manipulation is rather straight forward.

It would appear that any file, even though it is not of the DATATRIEVE RMS structure, could be similarly parsed and rebuilt using these techniques.

Author Index of Wombat Examiner and 4GL Dispatch, Volumes 1 to 9

The Wombat Examiner and 4GL Dispatch is the newsletter of the DATATRIEVE/Fourth Generation Languages Special Interest Group of DECUS. The 1988 August issues is the end of 9 volumes of the publication. Fourteen issues were published between July of 1979 and June of 1985; thirty-six issues, one issue per month, were published between September of 1985 and August of 1988 for a total of 50 issues in the first 9 volumes over a period of 9 years.

Volume Combined	Volume Wombat Examiner	Number	Month	Year
	1	1	July	1979
	2	1	May	1980
	2	2	September	1980
	3	1	February	1981
	4	1	February	1982
	4	2	June	1982
	4	3	November	1982
	5	1	March	1983
	5	2	July	1983
	5	3	January	1984
	5	4	November	1984
	6	1	April	1985
	6	2	April	1985
	6	3	June	1985
1	7	1	September	1985
	. . .			
1	7	12	August	1986
2	8	1	September	1986
	. . .			
2	8	12	August	1987
3	9	1	September	1987
	. . .			
3	9	12	August	1988

Because the publication of the Wombat Examiner (and 4GL Dispatch) preceded the beginning of the DECUS U.S. Chapter SIGs Newsletters, volume 1 of the combined newsletters corresponds to volume 7 of the Wombat Examiner.

A subject index for Volumes 1 to 9 will be published in next month's newsletter. In the future, a yearly subject and author index will be published in the newsletter issue which falls just after the Fall DECUS Symposia.

Author Index of Wombat Examiner and 4GL Dispatch, Volumes 1 to 9

Gallagher, Joe, H. (con't)	V9N6p2, V9N6p10, V9N7p2, V9N7p7, V9N8p28, V9N9p2, V9N9p3, V9N10p3, V9N11p2, V9N11p5, V9N12p2, V9N12p20
Gettman, James E.	V8N5p25
Gey, Fredric	V2N1p36
Glaser, Harold T.	V5N3p1
Golden, Terry	V5N4p44
Goldfield, Michael	V4N1p68
Golston, Susan	V7N5p6
Gray, Barrie	V7N12p32
Greer, Tom	V8N1p47
Guidi, John	V2N1p27, V4N1p53
Hare, Keith	V7N5p8
Harris, Diane	V5N3p54, V6N1p18
Harris, Sue	V7N8p5, V8N2p19, V9N8p2
Harris, Jr., Basil	V6N1p16, V7N9p4
Haskin, Denis	V7N6p9
Hazelwood, Ann	V2N2p14
Heinz, Chris	V5N2p58, V5N4p11, V7N5p4
Henning, John	V8N12p15
Herzmark, Leonard E.	V7N8p25, V9N9p2
Hilton, Joan	V1N1p10, V2N1p28, V4N1p81, V4N1p86, V4N1p97, V4N1p97, V4N2p2
Hoover, Bob	V7N6p11, V8N5p14, V8N12p16
Horn, Larry O.	V4N1p50
Hornback, David R.	V7N10p18
Hornback, Katherine	V7N4p4
Hulse, Les	V7N6p10, V7N7p19, V7N7p21
Hurst, Art	V4N1p102
Ingram, Wayne D.	V2N1p7
Israel, Jan	V5N2p71
Jasmann, Lawrence M.	V4N1p5, V4N1p99, V4N3p2, V5N2p57, V5N3p58, V5N4p1, V5N4p17, V6N2p2, V6N3p2, V7N1p2, V7N2p2, V7N3p3, V7N4p2, V7N4p4, V7N5p2, V7N5p8, V7N7p2, V7N9p2, V7N11p25, V8N5p13
Johnson, Walt	V6N1p21
Jones, John	V5N3p61
Jones, Wayne	V4N1p22, V4N1p102, V5N1p45, V7N6p8, V7N8p23
Kazanfer, Meryem	V5N4p51
Kellerman, Seymour	V5N2p70
Kessler, Judy	V4N1p67
Kimmel, Lorey B.	V8N12p11, V9N11p3
Krug, Paul	V5N4p37
Lambert, Ross W.	V4N1p68
Landau, Rick	V4N1p86, V4N1p96, V4N1p101
Leal, William M.	V8N1p22, V8N6p2, V8N6p13, V8N10p8, V8N10p12
Lederman, Bart Z.	V4N1p82, V4N1p84, V4N1p98, V4N1p103, V4N2p8, V4N3p5, V5N2p32, V5N2p72, V5N4p3, V6N1p2, V6N3p3, V7N1p4, V7N1p30, V7N2p3, V7N3p10, V7N5p3, V7N6p13, V7N7p21, V7N8p12, V7N8p27, V7N11p24, V7N12p22, V8N2p4, V8N2p11, V8N3p15, V8N5p26, V8N8p3, V8N11p19, V8N12p10, V9N1p11, V9N3p15, V9N4p12, V9N6p14, V9N8p25, V9N9p6, V9N10p2, V9N10p6, V9N12p3
Lemmon, Jean	V4N1p86
Leving, Jeff	V8N1p43
Lott, Bob	V5N4p30, V5N4p45, V5N4p49, V5N4p53
Mahaney, Thomas	V5N4p35
Martin, Joanne	V8N12p16
Mathis, Jack	V8N5p10
Mazzoni, Michael E.	V7N3p5
McCormick, Bob	V2N1p14, V2N1p30
McIntyre, James M.	V1N1p23, V3N1p36

McKinzie, Mary	V5N4p38
McMillan, James E.	V6N1p11, V7N8p24, V8N5p11
McQuire, Ed	V8N5p23
McWilliams, Elaine V.	V7N6p4
Megginson, Ernie	V8N5p31
Mei, Joseph	V9N12p12
Meirowitz, Mark J.	V2N2p7
Merkle, Fritz	V4N1p94, V4N3p42
Meyer, Maryhelen	V8N5p25
Morgan, Brad	V4N1p95
Morris, Henry	V5N1p22
Naecker, Philip A.	V4N1p36, V4N2p5, V5N1p70, V5N3p53, V5N3p55, V5N4p21, V7N6p10, V7N6p12, V7N8p27, V7N9p3, V7N10p3, V7N10p7, V7N11p9, V7N11p27, V7N12p12, V8N1p3, V8N2p22, V8N4p12, V8N4p20, V8N5p2, V8N6p5, V8N7p39, V8N8p42, V8N9p21, V8N12p16, V9N1p2, V9N3p11, V9N5p2, V9N6p5
Nicholas, Michael D.	V7N3p9
Nordby, Dave	V1N1p10, V3N1p3, V4N1p97
Noyce, Bill	V5N1p61, V5N1p68
Oberle, Jerry	V9N12p10
Opalka, Bill	V8N12p15
Pacheco, Stephen	V6N1p36
Packer, Bernice	V5N4p14
Palme, Lars	V3N1p46, V3N1p47
Parsons, Dudley	V5N2p64
Pearson, Bob	V6N1p20
Pinney, Diane	V8N4p15, V8N5p16
Pitluck, Samuel	V5N3p8
Porteous, William	V6N1p15
Pratt, Lisa	V8N5p13
Putnam, John F.	V7N11p30
Racel, Peggy	V5N2p45, V5N2p62, V5N3p58, V5N3p63
Reines, Herbert G.	V8N12p9
Rieman, John	V7N7p20
Rock, Bob	V2N2p10
Rogers, Anthony D.	V9N10p9
Roseberry, Bert A.	V5N3p52, V5N3p64, V5N3p66, V5N4p40, V6N1p17, V6N3p8, V6N3p17, V7N6p6, V7N7p10, V7N7p16, V8N11p14, V9N2p2, V9N2p3, V9N8p22
Sadler, Brent	V5N2p58
Saint–Jean, Rafael M.	V3N1p45
Samson, Paul	V4N1p101
Saxer, Gary	V4N3p29, V5N1p67
Scandora, Anthony E.	V6N1p9
Scannell, Wade	V5N4p30, V5N4p48
Schipani, Frank	V8N1p47
Schneider, Andrew W.	V5N3p3, V6N1p21
Schwartz, Dana J.	V7N8p20, V7N11p28, V8N2p3, V8N8p2, V9N12p15
Schweer, Kurt	V4N1p100
Scopelliti, Pasquale F.	V5N4p31, V6N1p12, V7N3p9, V7N7p18, V7N11p27, V7N11p31, V8N5p17, V9N11p4
Shafer, Mary B.	V2N2p16
Snyder, Patricia A.	V3N1p30
Spoor et al, E	V9N5p5
Starkey, Jim	V2N1p10, V2N1p18, V4N1p95, V5N4p27
Stephenson, Glen	V4N3p2

Stern, Jr., Donald E.	V7N1p35, V7N2p12, V7N6p2, V7N6p4, V7N7p3, V7N7p22, V7N8p2, V7N9p2, V7N9p23, V7N10p2, V7N11p3, V7N11p25, V7N11p32, V7N12p2, V7N12p42, V8N1p49, V8N2p2, V8N3p3, V8N3p23, V8N4p3, V8N5p9, V8N6p20, V8N8p20, V8N8p49, V8N8p50, V8N10p2, V8N11p2
Sventek, Virginia	V4N1p1
Swanger, James	V5N2p19
Tabor, William I.	V8N12p14
Tamer, Kathy	V2N2p2, V2N2p40
Thomas, Robert F.	V2N1p31
Toyne, Timothy T.	V5N3p33
Valentine, Pamela A.	V5N3p54, V8N5p16
Van Itallie, Frederick J.	V5N1p65, V6N3p14, V8N12p9
Vander Wall, Jay	V5N4p29
Ward, Steven	V5N3p64
Washburn, Diana	V5N4p22, V5N4p50, V6N1p30
Watson, Al	V5N1p64
Watson, Chuck	V1N1p3, V1N1p15, V1N1p17, V1N1p19, V2N2p3, V2N2p11, V2N2p39, V3N1p23, V4N1p6, V5N2p67
Weathers, Dale	V8N1p45
Wegscheid, Doug	V6N3p15, V8N5p7
Wildrick, Dean E.	V4N1p52
Williams College	V6N2p17
Wilson, Tony L.	V4N1p46
Wood, Bill	V9N11p2
Woodland, R. J.	V5N4p36
Wool, T. Chris	V5N1p3, V5N2p67, V5N4p50, V6N1p7, V6N1p36, V6N2p6, V7N8p3, V8N1p52, V8N4p35, V8N7p25, V8N10p2, V9N6p3
Wrobel, Katherine	V6N1p6, V7N6p11
Zuhr, Ken	V7N2p15

edusig
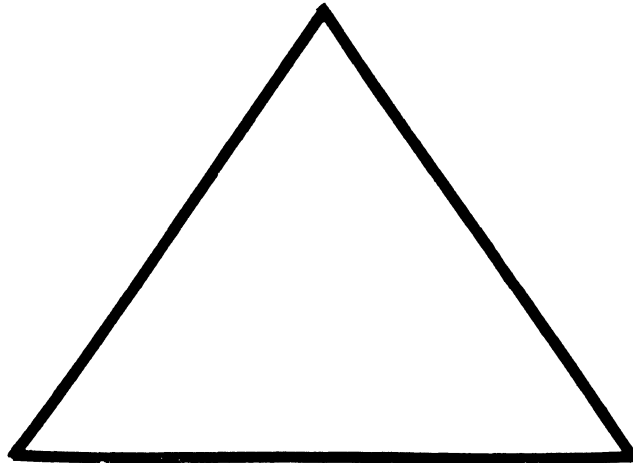
Networking

Over the last few years campus-wide networking has become a much talked about topic in educational computing. Recently, smaller educational institutions have begun to investigate, design, and implement successful campus-wide networks. Recognizing that the computing environment in four-year and two-year colleges depends on the specific goals and direction of the institution, EDUSIG will sponsor a Pre-Symposium Seminar at Anaheim that is geared toward these education environments.

In many cases, small institutions can learn from the successes and failures of earlier efforts in the research-oriented university community. However, because computing has a different focus in smaller institutions, these experiences cannot always be transferred directly. In addition, some issues which are specific to these environments will be discussed.

This seminar will focus on networking in the four year and two year college environment. This seminar is also applicable to school districts who are implementing networking as well as departments in larger universities who are involved in networking.

Topics to be covered in the seminar are:

- Developing a networking vocabulary
- Understanding the differences among various types of networks
- Issues in educational campus-wide networking
- Networking standards and their impact in education
- Dealing with the multi-vendor computing environment in education
- PC networking on campus
- Developing a campus wide network plan
- Considerations in evaluting networks and products
- Inter-campus networking
- National and International networking in the education community

Also, attendees at this seminar will have the opportunity to participate in a case study of campus-wide networking in a small educational institution.

Who Should Attend:

This will be an interactive seminar designed for anyone involved in networking in small educational institutions. The focus is on building a common base of information with which attendees will be better prepared to plan and implement networks within their organization.

Seminar Leader:
Michael Greene is a networking consultant for the Education Industry Marketing group within Digital Equipment Corp. In this position he works with both educational institutions and other

networking groups within Digital to provide focus on networks, network trends, and network requirements for education. For the past eight years he has held a variety of marketing and engineering positions within Education Industry Marketing at Digital. He holds a B.S. degree from St. Andrews Presbyterian College.

Don Fuhr

Graphics
Applications
Special
Interest
Group
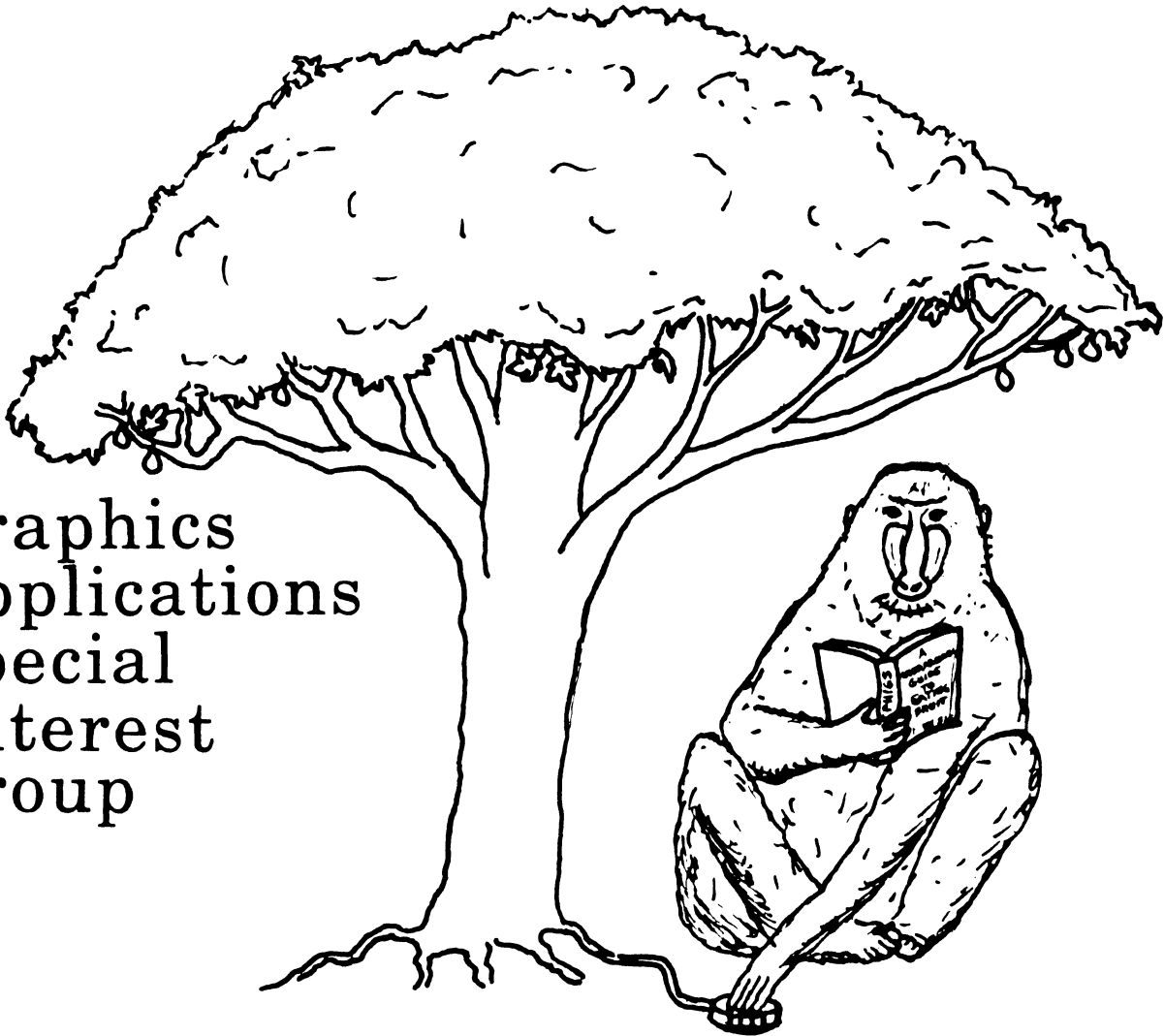
The Graph Paper
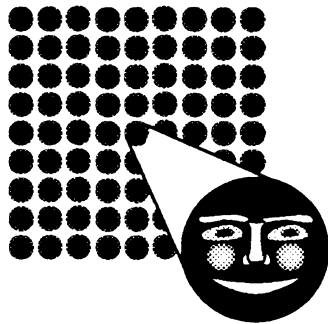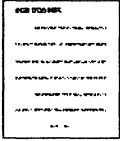
submissions

Robert Hays
3621 South State Road
Ann Arbor, MI 48106

Articles, copies of viewgraphs, tips and tricks, and graphics output are all welcome submissions for the Graphics Applications Special Interest Group (GAPSIG) newsletter; here's how YOU can make submissions:

1) Send in a tape. Tapes can be 1600 or 6250 BPI density. Please provide straight ASCII files or Mass-11(TM) format documents and a letter with your name and address. Please place any charts in separate files.

2) Send in paper. Hey, your editor can type and chew gum at the same time, so don't be afraid to send in hard-copy. And, if all you have is notes, FINE! Send them in!!! We have many folks who can take the ideas and flesh them out with English language extensions. Questions, with or without answers, are desirable, too.

3) Mail the article, etc. to user name HAYS on DCS.

Your editor's address is printed above, so mail your submission today!
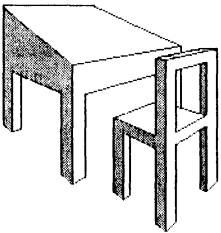
from the chair's desk

Bijoy Misra
GAPSIG Chairperson

The Anaheim Symposium will begin in a couple of weeks and I hope you have made plans to be at Disneyland. The Graphics SIG will have more than fifty sessions spread covering workstations, window environment, image processing, postscript, CAD/CAM, and applications software with many new announcements and updates. The theme for the Symposium for the SIG is "Color Graphics" and we've invited Mr. Judson Rosebush from Judson Associates in New York to be our keynote speaker. The keynote session and other sessions related to the theme have been scheduled for Tuesday, Oct 18th.

While the application of color in graphics is not new, techniques of applying color for visualization is a newly-emerging field. Animation technology and myriads of scientific data have put a new thrust on the way color should be used for display and presentation purposes. The hardcopy reproduction of color on the other hand is a strong function of the medium and many different color schemes have evolved to accomodate the varying needs of color reproduction. Color display, color perception and color hardcopy will be among the topics that we plan to discuss at Anaheim. A competition on hardcopy prints will be hosted in the campground. A Vaxstation 8000 and a Vaxstation II/GPX will also be available to try out. More new products and application software will be exhibited in the Main Exhibit Hall. I'm sure that we'll have a very productive exchange of information in this new area of growth.

special events from the GAPSIG at the Fall Symposium

The theme for the 1988 Fall Symposium for the Graphics Applications SIG is color graphics. The following *special events* are planned:

Monday night, 7:00 to 10:00 PM
Reception at the GAPSIG Campground with a cash bar

Wednesday night, 6:00 to 8:00 PM
Volunteers Reception at the GAPSIG Suite

Thursday night, 7:00 to 10:00 PM
Open House at the GAPSIG Suite

Locations for the Suite and Campground will be announced in the UPDATE.DAILY. Also, don't forget the graphics contest sponsored this year by the GAPSIG. Bring your best graphic, be it a computer manipulated image or a graph or a plot and enter it! You could win!

I also want to touch upon the Open Software Foundation. The Open Software Foundation (OSF) was announced on May 17 as a non-profit, industry-supported research and development organization, sponsored by some of the major vendors of computer hardware and software including Digital and IBM. Among the major goals for the Foundation is the support of application environments that are portable, scalable, and operable on computers of different vendors. With the premise that the Foundation will work in cooperation with the international standard groups, the benefits to the user community could be enormous. The Foundation plans to establish industry standards in areas where no standards exist at the present time and, hopefully, the industry standards will be absorbed as international standards in due course of time.

The level zero application environment specification for OSF includes POSIX, X/OPEN, C, Fortran, Pascal, GKS, PHIGS, TCP/IP, Telnet, SQL, etc. To make the process open, OSF will periodically solicit Requests for Techology, which must meet the required criteria set forth by the Foundation. The current Request for Technology is on the User Environment Component and may be submitted to Open Software Foundation, 20 Ballad Way, Lawrence, MA 01843. The queries on technology submission may be directed to RFT Enquiries Desk, Tele: (508)683-6803. I'll like to see a strong DECUS participation in the OSF activities and I'll advise all of you to write to me about your views on the Charter of the Foundation.

We'll host a few BOF (Bird's of a Feather) sessions on the Foundation's work plan; I invite you to join us in those sessions. The schedule and the room number will be announced in the Update.Daily, and let me hope that we can work together to produce a white paper on behalf of the SIG on issues relating to the OSF. If you get to Anaheim early, see us up in the Graphics suite on Sunday afternoon after the presymposium seminars. See you there!

the GAPSIG
Fall 1988 Symposium schedule

Monday – Windowing and Imaging

9:00 –	9:30	GR050	WORKSTATIONS, WINDOWING AND GRAPHICS ROADMAP	Henry Schneiker
9:30 –	10:00	GR051	GRAPHICS APPLICATIONS SIG BUSINESS MEETING	Bijoy Misra
10:00 –	11:00	GR024	WORKSTATION APPLICATION OVERVIEW	Worksystems Engineering
11:00 –	12:00	GR040	INTRODUCTION TO WORKSTATION-BASED USER INTERFACES	R. Patrick Johnson
12:00 –	1:00	GR005	VMS WORKSTATION SOFTWARE PRODUCT UPDATE	Larry Villandry
1:00 –	2:00	GR046	GRAPHICS MODELS IN WINDOWING	Richard J. Greco
2:00 –	3:00	GR016	DECTERM: THE DECWINDOWS TERMINAL EMULATOR	Edward Parsons
3:00 –	4:00	GR004	INTRODUCTION TO SOLID MODELING	Jim Sims
4:00 –	5:00	GR006	UIS TO X MIGRATION	Irene McCartney
6:00 –	7:00	GR065	DIGITAL'S IMAGE PROGRAM AND PRODUCTS	Joel Arker
7:00 –	8:00	GR066	THE VAX IMAGE SERVICES PRODUCT SET	Joel Arker
8:00 –	9:00	GR067	DEVELOPING APPLICATIONS USING VAX IMAGE SERVICES	Joel Arker
9:00 –	10:00	GR068	MANAGING IMAGES USING ISMS	Joel Arker

Tuesday – Color Graphics

9:00 –	10:00	GR036	COLOR GRAPHICS KEYNOTE	Judson Rosebush
10:00 –	10:30	GR007	AN INTUITIVE COLOR SELECTION MENU FOR THE VAXSTATION	Jeff Johnson
10:30 –	11:00	GR021	QUICK AND DIRTY SCIENTIFIC ANIMATIONS ON VAXSTATIONS	Steve Hankin
11:00 –	12:00	GR020	DATA PRESENTATION IN A TECHNICAL WORLD	Tom Mahowald
12:00 –	1:00	GR012	RULES OF THUMB FOR EFFECTIVE COLOR TERMINAL DISPLAYS	Patricia English-Zemke
1:00 –	2:00	GR057	TRENDS IN MANIPULATION OF COLOR IN GRAPHIC TERMINALS	Gerald Murch
2:00 –	3:00	GR015	THE FUTURE OF COLOR PRINTING	Richard Landau
3:00 –	4:00	GR042	GRAPHICS AND COLOR PRINTING SOLUTIONS FOR THE PROFESSIONAL	Ian Garbutt
4:00 –	5:00	GR010	INTRODUCTION TO IMAGE PROCESSING TECHNIQUES	Stephen L. Schultz
5:00 –	6:00	GR011	IMAGING TECHNIQUES IN THE FOURIER DOMAIN	Stephen L. Schultz

Wednesday – Phigs and GKS

9:00 –	10:00	GR060	INTRODUCTION TO PHIGS	James Flatten
10:00 –	11:00	GR063	VAX PHIGS UPDATE	James Flatten
11:00 –	12:00	GR054	RENDERMAN: A 3D SCENE DESCRIPTION INTERFACE FOR COMPUTER GRAPHICS SYSTEM	Thomas Porter
12:00 –	1:00	GR061	PEX – A THREE DIMENSIONAL EXTENSION FOR X WINDOWS VERSION 11	James Flatten
1:00 –	2:00	GR062	AN INTRODUCTION TO PROGRAMMING WITH VAX GKS	James Flatten
2:00 –	3:00	GR064	VAX GKS (GRAPHICS KERNEL SYSTEM) UPDATE	James Flatten
2:00 –	3:00	GR048	HARD COPY WORKING GROUP MEETING	Henry Schneiker
3:00 –	4:00	GR028	BUILDING A USER APPLICATION WITH THE GKS STANDARD	Stuart Henderson
4:00 –	5:00	GR041	INCREASING REQUIREMENTS FOR GRAPHIC METAFILE SUPPORT	Richard Smith
5:00 –	5:30	GR009	CONVERTING FROM THE REGIS GRAPHICS LIBRARY TO GKS	Steve Henson
5:00 –	6:00	GR038	IMAGING WORKING GROUP	Bob Goldstein
5:30 –	6:00	GR001	BEZIER-BASED MICROCOMPUTER CURVE-FITTING ROUTINE	Teruo Fujii

Thursday – Electronic Publishing

9:00 –	10:00	GR019	HOW TO EVALUATE AND SHOP FOR TECHNICAL WORKSTATIONS	Ken Filcoff
9:00 –	10:00	GR013	FONTOLOGY: THE TERMINOLOGY OF DIGITAL TYPOGRAPHY	Richard Landau
10:00 –	11:00	GR034	THE NETWORKED GRAPHICS TERMINAL: FROM RS-232 TO LAN TO WINDOWING SYSTEMS	Don Hershiser
10:00 –	11:00	GR014	TEXT LAYOUT AND USE OF FONTS: AN OVERVIEW	Richard Landau
11:00 –	12:00	GR043	LATEST IN GRAPHIC TERMINAL ARCHITECTURE	Rick Kamp
11:00 –	12:00	GR017	THE DECWINDOWS FONT LIBRARY	Jim Flowers
12:00 –	12:30	GR003	PENDING TECHNOLOGY BREAKTHROUGHS IN MECHANICAL CAD/CAM/CAE SYSTEMS	Raymond H. Kurland
12:00 –	1:00	GR008	AN INTERACTIVE VAXSTATION FONT EDITOR	Jeff Johnson
1:00 –	2:00	GR058	BEGINNING POSTSCRIPT PROGRAMMING	Henry Schneiker
2:00 –	3:00	GR018	ADVANCED POSTSCRIPT PROGRAMMING	David Gabbe
8:00 –	9:00	GR069	CDA: DIGITAL'S COMPOUND DOCUMENT ARCHITECTURE	Eirikur Halgrimsson
9:00 –	10:00	GR049	GAPSIG COMPUTER GRAPHICS SLIDE SHOW	Bijoy Misra

Friday – Wrap-up

9:00 –	10:00	GR044	GAPSIG WRAP-UP	Bijoy Misra

GKS to RGL conversion

Steve Henson
Woods Petroleum Corporation

We began to define the process for conversion of our applications from RGL to GKS as soon as GKS arrived at Woods Petroleum. We decided to write a set of conversion routines that performed the same functions as the RGL routines that we were using. The conversion was done in the following order:

1) identify routines that required conversion,
2) study the RGL routines to gain insight into programming concepts and methods, and learn how to use GKS,
3) write conversion routines based upon graphics primitives based on GKS,
4) test every routine separately and in combination with the other routines,
5) incorporate the new routines into existing applications programs, and
6) test the converted application programs.

The routines requiring emulation were identified. We have programs that produce cross references and directories of programs and subroutines which were used to determine exactly which RGL routines were in use in our programs. Those routines were targeted for emulation. Here at Woods, our graphics are not terribly complex, consisting of X-Y plots on coordinate, semi-log, and log-log graph paper, and a couple of programs that do some line drawings in the world coordinate system. The conversion steps were tailored to accomodate our use of graphics.

Once the routines requiring modification were identified, we began to study the source listings of those routines, which is provided on the RGL V. 1.1 distribution. This provided insight into the methods and concepts used in writing general purpose graphics subroutines. We also tried to convert the RGL routines directly by simply using the GKS primitives, but this was abandoned because the RGL routines were tightly bound to the ReGIS instruction set.

We began learning how to use GKS during the needs identification phase. Our version of GKS was provided by a third party vendor that offers a course in GKS. The class was very helpful. I felt more comfortable programming with GKS and even began writing conversion routines during my flight back from Oklahoma City, the site of the course!

a guide to migrating UIS applications to X, part 2

·We wrote the conversion routines from the primitives up. This created a software foundation for some of the more specialized routines. It also allowed us to gain some practical experience with GKS on a very basic level. We could test the primitives easily and gain quick satisfaction while we began to understand how all the GKS pieces fit together.

The first step in writing the primitives was the establishment of some common areas. This was necessary for the establishment of values in one routine and the use of those values in another. For instance, LINE draws a line from the current graphic cursor location to the coordinates given. We established variables in common for the current cursor location. The two coordinate pairs are then passed to the GKS routine for generating a poly-line.

Once the primitives (i.e. LINE, MOVE, etc.) were written, the more complex routines were written. Each complex routine was built on top of the primitives. This same concept was used in the RGL routines. As we wrote more complex routines, we added more variables to the common area. Some of the variables were flags that were tested to determine various attributes, such as whether shading was turned on.

Unit testing of each routine and integration testing of the routines in concert was critical to the success of the project. During testing, we found routines that updated common areas other routines needed. The common area continued to grow to accomodate isolation requirements between routines. The code was checked for re-entrancy. Each routine was tested to see that it left windows and viewports in the same state as when the routine was entered. Fortunately, these lessons were learned while still working with the primitive routines.

The work progressed at a good pace until the DRAW_GRAPHPAPER routine. It took some time to develop that routine in its first form. After finishing it, I went on to CLEAR_AREA. It was then that I discovered a difficult problem: GKS has no ability to "undraw" an area. The only method I had to clear an area was to draw a filled box in the background color. Unfortunately, the hardcopy drivers we use have only one drawing color. Consequently, the graphs looked fine on the screen, but the hard copy graphs had a black box in the area that was supposed to be cleared. I had to re-engineer the DRAW_GRAPHPAPER and implement a shielding mechanism. CLEAR_AREA is called early in the application program and defines an area that is shielded. The shielding was then implemented in other routines as necessary to our operation.

It took six months to write the conversion routines. One of the last routines written was INIT_GRAPHICS because all the variables that would need initializing had not been established. Upon finishing the conversion routines, we began to convert application programs. That's when the effort bore fruit: it took less than thirty minutes to convert the first program and less than a week to complete the conversion of all thirty-some-odd graphs that we produce. We've used GKS since September of 1987 and there have been very few problems.

Digital Equipment Corporation

2 COMPARATIVE OVERVIEW OF X AND UIS

This section provides an overview of similarities and differences between the two window systems and will assist in planning a migration strategy from UIS to X. In this section, the following aspects of the window systems will be covered:

Architectural Overview

Coordinate Systems

Windows

Graphics Output

Color

Virtual Displays and Display Lists

Input

Window Manipulation

Data Association

2.1 Architecture Comparison

The architecture of both UIS and X provide window functions required by most applications. The major differences between X and UIS come about because of their design philosophy. UIS was designed to be a more "complete" graphics and window system while X was designed as a low level graphics/window system which expects high level features such as display lists, virtual displays and world coordinates to be implemented in layered libraries. Because of this different design philosophy, many UIS applications which take advantage of these higher level features will be more difficult to port.

One of the most significant design differences between UIS and X is that X is designed as a network based window system, while UIS is designed as a kernel based window and graphics system. A kernel based window system must display all graphics on the local workstation. In contrast, a network based window system allows applications to execute on any computer while displaying output on any workstation in the network. An X application, commonly referred to as a "client", makes window system calls without concern for where the output is presented or where the application is executing. This mechanism is entirely transparent to the user.

The output of a client is sent to the desired workstation for display. A process on the workstation, known as the X "server", is responsible for performing the display operations in the destination window. The server also sends input from workstation devices, such as the mouse and the keyboard, back to the client application.

Although X supplies a rich set of input and output capabilities, it was designed to be device independent. Some graphics hardware may provide highly specialized features that X may not utilize. However, X has an extension mechanism that permits application developers to enhance the window system to take advantage of these unique features.

These additional architectural features in X provide some very powerful capabilities not available in UIS. Many applications will derive significant benefit from this architecture.

2.2 Coordinate Systems

When an application performs output to a window, it must specify the (x,y) location in the window where the output is to be drawn. The interpretation of the (x,y) pair is determined by the coordinate system of the window.

The lowest level coordinate system interprets (x,y) pairs as pixel locations. This is known as the device coordinate system which is supported by both X and UIS. In both systems, the pixel locations are relative to the origin of the window. The origin of an X window is the upper left hand corner with the y-axis increasing downwards. In contrast, the origin of a UIS window is the lower left hand corner with the y-axis increasing upwards. In both systems, the x-axis increases from left to right.

UIS also provides a world coordinate system. In this system, the application defines any convenient unit of measure such as inches, miles or microns. Thus, the application may define its coordinates to represent inches, kilometers, seconds, etc. instead of pixels. The window system is responsible for converting from world coordinates to device coordinates when performing output.

Since X does not support the use of a world coordinate system, an application requiring this capability must provide its own set of routines to perform this world-to-device coordinate conversion.

2.3 Windows

Before an application can perform any input or output on the workstation, it must create a window on the display. The window defines the region into which the application may perform output. The window system insures that no output extends beyond the boundaries of the window. This is known as "clipping." Both UIS and X clip to the window boundaries in the same manner.

Prior to creating a window in X, the application must receive permission from the X server on the desired workstation. To provide system security, each X server determines which clients are permitted access to its resources.

> NOTE: This is implementation dependent.

All X windows are arranged in a hierarchy or tree structure of arbitrary depth. The entire surface of the screen is covered by a single window called the 'root' window. All other windows are descendants of this root window. When a window is created, its 'parent' window must be specified. If this new window (know as a 'child') extends beyond the boundaries of the parent, it is clipped by the parent. Thus, no window may output outside the boundaries of any of its ancestors. A window may have multiple children but a child has only a single parent.
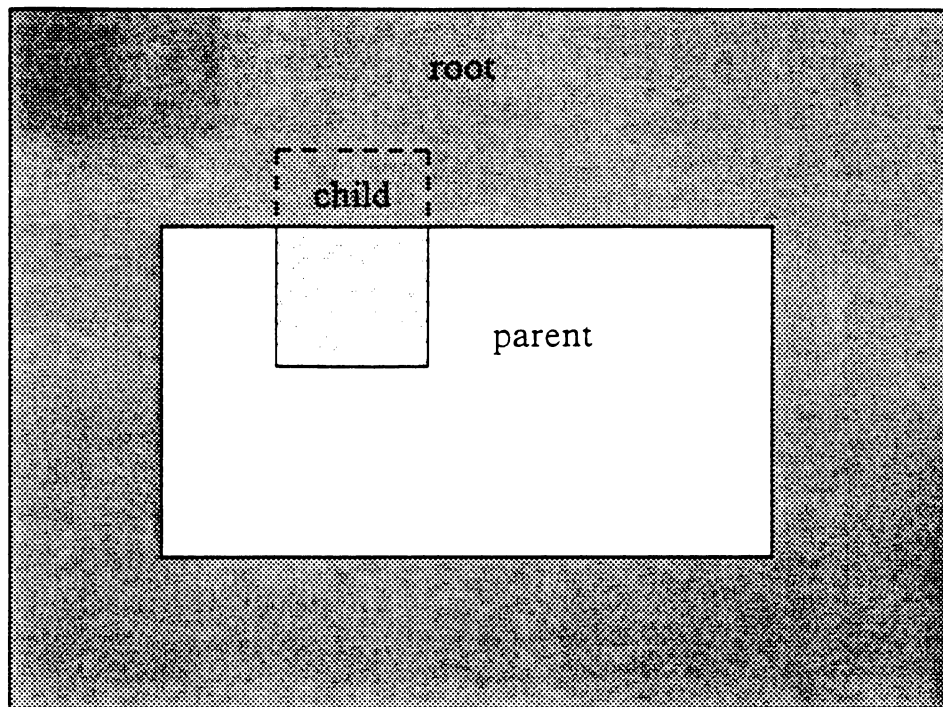
Figure 1
Child is clipped by the parent's boundary

Normally, an X window has a solid background causing it to obscure any window it overlaps. However, a window may be defined without a background, which makes it transparent. Among other things, these windows facilitate temporary overlays such as gridlines.

When any portion of an X window that was obscured by another window becomes visible, the application must be capable of recreating what should be displayed in the newly exposed area. In contrast, UIS automatically restores exposed areas so the application need not be responsible for refreshing the window.

A special characteristic of windows in X is that they require very little overhead. Therefore it is feasible for an application to use a large number of windows without suffering a significant performance impact. This characteristic, in addition to the hierarchical window structure, can provide very convenient methods for implementing various parts of applications such as the human interface.

2.4 Graphics Output

Window systems offer a variety of methods to create graphics output. Most systems provide a number of primitives to draw lines, polygons, and text. All output primitives have various characteristics that affect their appearance. Some window systems also furnish other primitives such as arcs. In addition, applications may group a set of polygons together into a single entity known as a region. Other functions include the ability to manipulate arbitrary rectangular areas. The following sections explore these topics in further detail.

2.4.1 Lines, Polygons, And Arcs - Both X and UIS provide routines to draw a single line, multiple disjoint lines, and multiple connected lines. These lines may be drawn with any specified width, line style, and color. When drawing a sequence of wide, connected lines, there are several methods that can be used to smoothly join them at their endpoints. X allows the application to choose from a number of joint styles.

Writing modes affect the appearance of lines at areas where they intersect. In X, writing modes are called functions. All UIS writing modes, with the exception of complement mode, are available in X.

Lines are drawn as a sequence of pixels. A single pixel may consist of multiple bits in order to produce color or shades of gray. The number of planes in the system determines the number of bits per pixel. Each bit within a pixel resides on a separate plane of display memory. A plane contains all of the bits from the same bit position in every pixel. Thus, if there are four planes in a system, each pixel value is determined by one bit from the same location in each of the four planes.
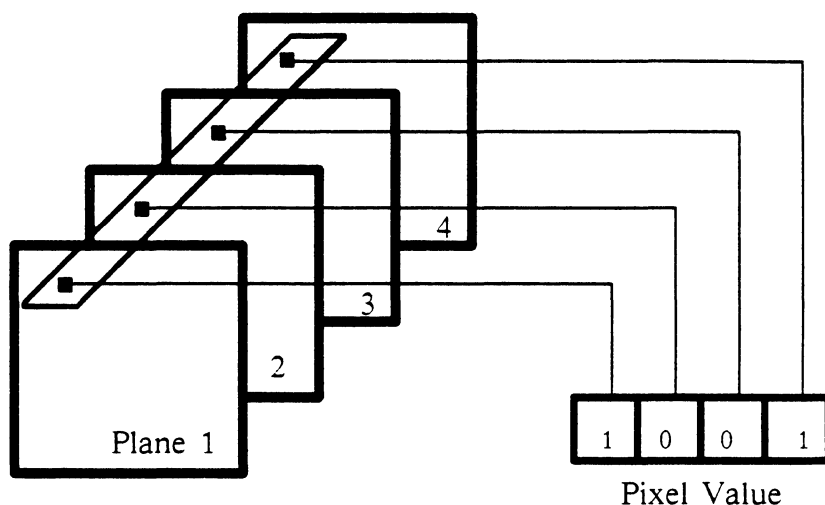
Figure 2
Formation of pixel value from multiple planes

When drawing lines in X, a mask may be used to restrict the operation to a subset of planes. Non-destructive animation is an example of a technique that significantly benefits from this capability. This feature is not available in UIS.

In addition to lines, X and UIS provide methods for drawing polygons. All of the output characteristics for lines are available for polygons. Polygons may optionally be filled. In UIS, fill patterns must be chosen from a predefined set. In X, the application has the flexibility to define the fill patterns, though no predefined patterns are supplied. X also provides the ability to perform filling through a stencil. These stencils are referred to as stipple patterns.

Another primitive supported in both X and UIS is an arc drawing primitive. All of the aforementioned output characteristics are available for arcs. In UIS, the endpoints of an arc may optionally be joined with a chord or to form a pie slice. In X, this option is only provided for filled arcs.

All of the possible output characteristics are grouped into structures called attribute blocks by UIS and graphics contexts by X. In both systems, the applications must specify the structure to use when

drawing a primitive. UIS provides a means of querying the characteristics in an attribute block while X does not.

2.4.2 Text – In a windowing system, text is also a graphics primitive. Both X and UIS support a large variety of fonts. UIS provides a number of text attributes that are not available in X. These attributes include text path, slope, slant, scaling, rotation, formatting, and character spacing. UIS also maintain the notion of current text position for each window.

2.4.3 Regions – It is often convenient to refer to one or more polygons as a single entity. X allows applications to create and manipulate these entities which are known as regions. These regions may be copied, moved, shrunk, and expanded. New regions may be created from the intersection, union, subtraction, and XOR of two other regions. Regions may also be compared. X provides routines to determine if an arbitrary point or rectangle is within a given region. As an example, regions are useful for selecting objects in graphics editors. Regions can also define a clipping region.

2.4.4 Direct Manipulation Of Pixels – Both X and UIS allow the application to manipulate rectangular areas of pixels within a window. Arbitrary areas may be moved to any window location. Images may be read from a window into the application's memory or, conversely, written from memory into a window.

2.5 Color

A significant portion of the cost of graphics systems can be consumed by the memory used to store the pixel values. Limiting the number of planes is one means of reducing the cost of the system. Unfortunately, this affects the number of colors that can displayed. However, the impact of this limitation can be reduced through the use of a color map. A color map is used to convert a pixel value to a color on the display.

An additional benefit of using a color map is that it may be updated by the application at any time. This feature is necessary to implement a number of algorithms in fields such as image processing and CAD.

For the most part, X and UIS handle color maps in a similar fashion. However, minor differences exist. Since X was designed to be device independent, support was included for a wider variety of display types and color maps. Another difference is that X does not provide separate segmentation routines like UIS but the concept can be emulated.

Colors are commonly specified with one of three models: RGB, HLS, or HSV. UIS supplies routines to convert color specifications between models. X does not directly support the HLS and HSV color models. Part III describes the conversion routines.

2.6 Virtual Displays And Display Lists

A virtual display is an imaginary surface onto which objects may be drawn. The virtual display is defined by a range of values in the world coordinate system. The entire virtual display or any subset may be displayed in a window. Multiple windows can share the same virtual display.

A display list is a list of objects in a virtual display. These objects can be scaled, translated, rotated, and edited by an application. A display list may be saved into a disk file called a metafile.

Since X does not support virtual displays and display lists, an application requiring these capabilities must provide its own set of routines.

2.7 Input

One of the responsibilities of a window system is to receive input from workstation devices and deliver the events to the proper processes. This delivery occurs synchronously or asynchronously depending on the window system.

Systems that use a synchronous architecture place events in a queue until the application requests them. This allows the application to determine when input is to be processed. In contrast, asynchronous systems interrupt the application immediately when input is received forcing the application to handle the event at that time. X delivers input synchronously while UIS delivers input asynchronously. Because of this major difference, the input structure of an application may change significantly.

Both X and UIS permit applications to specify which event types they wish to receive. Event types include mouse button change, mouse movement, keyboard key pressed, etc. X guarantees that these events will be delivered to the input queue in the same order in which they were received. In addition, since X uses a queue, the application may extract events from the queue in any order.

2.8 Window Manipulation

Workstation users may manipulate windows on the display using a window manager. These window manipulations in both X and UIS include move, resize, raise (pop), lower (push), iconify, and de-iconify. In addition, the window manager in X allows the user to change the focus of keyboard input to any window. This is similar to attaching the keyboard to a window in UIS.

2.9 Context Management

X applications often use a large number of windows because of the low overhead associated with them. X provides routines to assist the application in managing many windows. These routines associate window system and context information with the application's own information. For instance, they can be used to help the application determine what action to take for a given input event.

2.10 Conclusion

The following tables summarize the issues discussed in this section. Each table lists features specific to X or UIS. UIS features are listed with a rating estimating the amount of effort required to implement the given feature in X. Additional X features that UIS does not provide are listed in Table 2.

FEATURE	DIFFICULTY
Attribute Query	1
Color Segmentation	1
Color System Conversion	1
Predefined Fill Patterns	1
Asynchronous Input	2
Window Damage	2
World Coordinates	2
Display Lists	3
Metafiles	3
Geometry Transformation	3
Text Attributes	3
Virtual Displays	3

Table 1 – UIS Features not implemented in X

Difficulty ratings estimate the amount of effort required to implement this feature in X. The ratings range from 1-3 with 1 requiring the least amount of effort

FEATURE
Cut/Paste Buffers
Context Management
Event Queues
Hierarchical Windows
Low Window Overhead
Plane Access
Property Lists
Regions
Stipple Fill
Tile Fill
User Defined Fill Patterns

Table 2 – Features Unique to X

TABLE OF CONTENTS

EDITOR'S NOTES

It's months like this which make the editor's job worthwhile! This issue of *Leverage* has some of the best articles we've printed, several of which arrived completely unsolicited! My sincere thanks to our contributors this month, for a fine mix of high quality submissions.

I'd like to call your attention specifically to the article on Configuration Management by Mark Kidwell of Texas Instruments, and to the article on Patching code with VAX DEBUG, by Jerry Oberle. Either one is worth the cost of yur subscription by itself.

Also in this issue is the current L & T SIG wishlist. This is quite extensive, but is labeled as to products and/or topics. Please take the time to become familiar with this, the items will be discussed at the Anaheim symposium, and there will be an opportunity to express your feelings regarding them in a forthcoming issue of *Leverage*.

On another note, DECUS is (again) considering modifications to the format and organization of the newsletters. While it is too soon to discuss any specific details, I am very pleased that the main thrust seems to be bringing the widest range of information to the most people possible, rather than making them self-supporting, their appearance, or other peripheral matters. At any rate, if you have specific recommendations or concerns in the area of the newsletters, now is the time to make them known. Within each of the SIG's, the Chair, the Communications Committee Representative, and the Newsletter Editor would all welcome your input.

Keep up the submissions, folks! Hopefully I'll see you in Anaheim!

LANGUAGES AND TOOLS SIG WOODS MEETING REPORT

July 24-25, 1988

The Languages and Tools Special Interest Group held its Summer Woods meeting at the Merrimack Hilton in Merrimack, NH and at DEC's Spit Brook Road Facility in Nashua, NH. On Sunday, the SIG Steering Committee met from 8:00 am to 5:00 pm to discuss matters of SIG planning and administration. On Monday, the SIG Steering Committee met with Digital personnel at Spit Brook Road for non-disclosure discussions and feedback on Digital-SIG interaction during symposia. SIG Secretary Mark Kidwell assigned attendees to record each section of the meeting, and edited the resulting notes into this report of the Sunday Steering Committee discussions. The SIG is indebted to Digital's Steve and Leslie Klein for their outstanding hospitality on Sunday evening.

Steering Committee attendees were:

* Barry Breen, Seminars Committee Rep
* Earl Cory, Vice Chair, Symposia Rep
* Al Folsom, Newsletter Editor
* Steve Jackson, Incoming Acting Symposia Rep, Standards Activities coordinator/PDP-11 Rep
* Mark Kidwell, SIG Secretary
* Scott Krusemark, FORTRAN Working Group Chair
* Tony Mione, Incoming Session Chairs Coordinator
* Joe Pollizzi, Incoming Chair
* Dave Powell, VAXset Working Group Chair
* Dave Ream, Volunteers Coordinator/Incoming Working Groups Coordinator
* Tony Scandora, Library Committee Rep
* George Scott, Clinic Directory/Incoming Masters Coordinator
* Terry Shannon, Incoming Update.Daily reporter
* Jack Straub, PL/I Working Group
* Mike Terrazas, Incoming Campground Coordinator
* Bob van Keuran, Wishlist Coordinator
* Sam Whidden, Chair
* Kerry Wyckoff, Incoming CommComm Representative
* Celeste LaRock, DIGITAL Counterpart

WORKING GROUPS (WG)

At the WG chairs meeting in Cincinnati WG responsibilities were set, and the WGs were commended for soliciting and getting sessions submitted. WGs are weak, however, in getting submissions to the Al Folsom, the newsletter editor. A suggestion was made (and is being pursued) to have one or two WGs responsible for each issue of the newsletter. For those SIG members on DCS who are unable to KERMIT an article, but are on CSNET, Joe Pollizzi is willing to receive articles and then KERMIT it over to Al. Joe's address is: "POLLIZZI@SCIVAX.STSCI.EDU"

Dave Ream is the new WG Coordinator. In the near future, he will be gathering the Roadmaps for Anaheim, comparing the WG list with the DB list, and scheduling new (possible) WGs as BOFs for the symposia. (Roadmaps are needed by the end of August.)

The volunteers coordinator is not receiving the membership lists from the WG chairs. The only complaint from the WGs is that DEC people often cannot answer technical questions. Also, there

were two CM open WG meetings scheduled for Anaheim. Earl Cory will use the later scheduled meeting for another L&T slot.

The discontinuation of the PDP-11 layered products WG was discussed. The overlap of efforts of this WG with those of the L&T Language WG might justify the discontinuation. The SIG promised Joe Mulvey (DEC) earlier to try continuing and reevaluate the request. PDP-11 WG will continue at least through Anaheim.

The Steering Committee felt there was broad general interest in this area and that L&T's Documentation Production WG might contribute something. It seems reasonable that the new group might evolve into a SIC.

Feedback from DECUS membership is that they like the WGs, the chairs are doing a good/great job overall, and that people really like the WG concept. This feedback is from not only members within the SIG, but from members who are not affiliated with any SIG.

ANAHEIM

Roadmap Session: The SIG SAGs/Road Maps are not appreciated as much as the WG Road Maps. Members were concerned about redundancy between the newtimers meeting and the roadmap session. What is desired is introducing the developers and L&T Steering Committee quickly. A suggestion was made to try breaking out to WGs after 20-30 minutes.

Wizard Session: In Cincinnati we could not get the questions/answers in advance of the session; therefore, Joe Pollizzi gave a preview which set the tone for the remainder of the session. The idea was to ensure that wizard stories reported positive software tricks, rather than tales of how to beat the system manager. This was accomplished successfully by calling on one or two people whose interesting stories where known in advance. We will do the same in Anaheim.

Campground: Although there was some confusion, documentation handling went well. Trish Guthrie will work out any remaining problems. It was noted that square tables work better than round, and that a bookcase would work even better (Celeste is looking into a loaned/donated bookcase from DEC which could also serve as a shipping container). After Anaheim, we will take inventory and give it to Trish. Some documentation will be "raffled" off in Anaheim, the rest will be packed for Atlanta.

Sam Whidden will handle buttons, pins, and novelties for Anaheim, but we will need a new coordinator afterwards. We also need to determine the closing time for the campground. Mike Terrazas will work on this. Mike is also responsible for job delegation.

Questionnaires went OK. We need to make a list to be sure we know who gets what forms. The senior counterpart gets the DEC questionnaires.

Steering Committee Meetings: We need to know if there will be a non-disclosure session Saturday. The idea of a full Steering Committee breakfast probably won't work due to conflicting schedules of the DECUS symposia operational units. To keep up communication within the SIG, we will have a breakfast Tuesday for all who can attend. The Anaheim meeting schedule includes:

    Saturday 1 pm - 5 pm; L&T SC
    Sunday 5 pm - 6 pm; symposium committee
    Sunday 8:30am - 10 am; reception
    Tuesday Breakfast
    Thursday 9:30 pm - 11 pm; Open L&T SC meeting
    Friday 4 pm - 5pm; L&T Wrap-up session

Folder: Making changes in the forms can be tricky. There can be a timing problem getting forms from one person to another. The roster and Volunteers list will go to the WG coordinator.

UPDATE.DAILY: Mary McCormick, the UPDATE.DAILY editor, was overwhelmed with articles in Cincinnati and felt that too many were redundant. We agreed that some redundancy is wanted. Joe Pollizzi and Terry Shannon will coordinate L&T submissions to UPDATE so that we have only the redundancy we want. We will submit one article per file. George Scott is to write an article on the Masters program, Dave Ream on the WGs, and Joe on where general L&T questions can be answered. All articles submitted through DCS ahead of time will be routed via Terry for coordination, unless we receive word otherwise. Sessions with notes will be flagged by Earl Cory in the SAGs on the weekend edition. If the L&T/UNISIG/AI suite is closed for a major event, that will be noted in the campground and, if time allows, in update.daily.

Suite: We need to report on the suite usage. We want to keep the suite open to the general public for the majority of time, whereas some SIGs do not.

Reception: Suite & Reception coordinators have the same responsibilities. We will get the food, AI/UNISIG will provide entertainment. Anaheim's gift will be a L&T logo key ring. The Cincinnati gifts, which were misplaced until after the symposia, may be sold in Anaheim at the DECUS Store or returned and used as the reception gift at Atlanta.

MISCELLANEOUS

The SIG SC job descriptions were handed in to the volunteer coordinator. These will be reviewed by the SIG Chair for the SIG's policies and procedures.

George Scott is to reword the front matter on the L&T Masters list to make it clearer what level of expertise is to be expected of a Master. He will implement a recertification plan, asking each current Master is he or she wants to make any changes in the listing. The listings should indicate both Masters expertise and also knows so that a questioner can go to the CMS Master that has used it with Fortran source files, for example. The listings should show PDP-11, UNIX, and other minority affiliations. We also need a box to indicate UPDATE on the Master's form and a box/question to indicate a PDP-11 master.

In a few weeks, Wayne Sewell will become the standards coordinator. He will be responsible for preparing a quarterly newsletter standards update and schedule.

UNIT REPRESENTATIVE REPORTS

Symposia: The incoming acting Symposia Rep attended the symposium scheduling meeting for Anaheim.

The session compression meeting was very successful, and we met our goals for reducing the number of sessions. We will repeat this elimination process before the symposium meeting of Jan 14. CFP cutoff for Atlanta will be Nov. 28, but abstracts will not be available until about 2 to 3 weeks later.

Cincinnati Analysis: Other than the campground problems (which have already been addressed), everything ran smoothly. With more attendees than anticipated, members ran out of hotel rooms. The hot dogs were a bust/fiasco at the reception. Overall, the SIG did an excellent job.

Seminars: We provided 7 seminars at the last symposia, of which only one was marginally attended. The seminars committee may be providing speakers with compensation by providing a certificate good for registration to one symposia over a 5 year period. We agreed it's about time.

A new product idea presented was to offer the sale of speaker notes. The initial plan is to try with 8 sets of notes. This raised the question of the seminars profits verses service.

Library: The SIG tape is out. It consists of 5 reels or 5 savesets if 1600BPI. It contains GNU, TEX, XWindows, and various user submitted programs.

CommComm: Our product, the VAXpad, has been approved for quantity selling by the DEC Bookstore. The DECUS office has provided a general information number for membership inquiries which will be printed on the back cover of the VAXpad.

CommComm has a concern that newsletter editorials by SIG Chairs might be mistaken as official DECUS business. The editor or the SIG chair should make clear when they are stating a SIG, DECUS, or personal stand on some topic.

Potential purchasers were not happy with the reprinting/non-reprinting of the L&T session notes during symposium. Consensus was reached in L&T that when the forecasting of price/quantity of pages is as far off as it was in Cincinnati, we do have to honor the preregistration price orders, but the price can be adjusted upwards for purchase of local reprints at the DECUS store.

The idea of moving the store from CommComm to the library (a tentative idea reported by the Commcomm Rep) was thought a good idea. Tony Scandora (Library Rep) couldn't see any negatives to moving. Another suggested possibility was to move to SIG Council.

Newsletters: DCS submissions for the newsletters must be given to Al by the 20th of each month. Al does not want ANY formatting information in the file. He has translation problems from LaTeX to TROFF.

Session Evaluation Report: George Scott promised to repeat the reports, though the speed of performance may not be repeated. The SIG willpay clerical expenses as needed. The SASE program for individual session reports will be repeated and will be advertised. One suggestion, for Atlanta, is that the SASE program be advertised in the call for session notes.

SIG CHAIR TRANSITION

There was not enough time to discuss this subject in depth. One item of note for the incoming SIG chair was the amount of time necessary for the job. SIG Chair only spends only half his DECUS time running the SIG; other responsibilities take up as much, if not more time. A question raised was why not let the chair appoint someone to represent him on SIG Council occasionally?

It was also pointed out that the L&T SIG has a large budget and some feel it should be cut. The SIG is under constant pressure to spend the same amount as it did a few years ago, before the merger of L&T and CL. However, what should be considered is that the combined budget is less than the two SIGs before the merger.

REPORT OF FORTRAN X3J3 MEETING

by
Rochelle Lauer

The following report outlines the highlights of the FORTRAN ANSI standards meeting held in Jackson Hole Wyoming Aug 8–Aug 12. An official document(X3J3/s8.104) went out for public review at the beginning of the year, and was met with general public disapproval. The intent of this meeting was to present and discuss alternative plans.

Status of the Standard

The meeting began with 9 alternative plans of action, all of which were responding to overwhelming negative comment to the FORTRAN 8X proposal which went out for public review at the beginning of the year. All proposals where somewhat scaled down in order to reduce complexity(a common public complaint). All the plans have added features not included in 8X, but called for by the public comment (e.g. BIT data type, pointers, implicit none, include).

Due to the merger and withdrawal of plans, Wednesday found us with 4 plans. Presentations were made and detailed comparisons were given. Discussion of the plans (arguments ?) took up most of Wednesday.

Straw votes on Thursday indicated that the committee was almost evenly divided among the plans. The plan proposers were asked to renew discussion and try to merge the plans.

On Friday, the committee agreed to present three of the plans to the ISO couterpart of X3J3(WG5). The fourth plan will only be presented if an agreement to merge two of the plans reached early Friday morning does not hold up.

Main points of disagreement between the plans were:
Array valued functions
Modules and module procedures
Optional arguments and keyword arguments to functions/subroutines
Interface blocks for dunctions/subroutine
Structures and pointers in common (storage association vs. namoc.)
multi-byte characters

Further discussions will be held at the next meeting in November.

Other Items Covered During the Week

Goals were defined and discussed.

Visiting commentors presented their views on 8X (Dr. Leyhe and Dr. Hunter).

Reaffirmations of F77 (10 year reaffirmation required for ANSI standards if they have not been updated.)

Responses to public review:
Text error messages —- killed
carriage control on open statement -- to be included
voted 23 to 1 to remove concept of deprecated features

Issues for further discussion:
        implementation of pointers
        how to do structures in common
Voted on IRDS data base standard. Vote was NO due to lack of FORTRAN bindings definition

CM DESIGN CONSIDERATIONS

BY
MARK A. KIDWELL
OF
TEXAS INSTRUMENTS INCORPORATED
DEFENSE SYSTEMS AND ELECTRONICS GROUP

NOTE
This paper was approved and presented for the 1986 Fall
Symposium (San Francisco) and the 1987 Spring Symposium
(Nashville). Since that time, changes have occurred in both the
DEC products discussed and the Texas Instruments Incorporated
SCMS Product. SCMS is not available to the public. DEC CMS
V3.0 now handles non-ASCII files, one of many improvements since
the DEC CMS V2.2 mentioned in the text of the paper.

In today's fast changing arena of computer science the field of Configuration Managament has been
receiving a lot of attention lately. New DOD standards have been approved, books are coming out,
and "Experts" are voicing their opinion – all too often with differing stances. Adding to the
confusion is the need to automate many of the procedures necessary for successful Configuration
Management (CM). The purpose of this presentation is to provide GUIDELINES for CM Managers
and developers when considering developing an automated Software Configuration Management
System.

Experience in and out of Texas Instruments Incorporated (TI) has shown that groups developing
CM systems often experience the same problems. Large corporate entities usually find that
multiple groups are developing methods and tools designed to attack the growing need for CM.
Not surprisingly, the approaches are often similar. The time and money spent by the different
groups would have been better spent by having a centralized focus point to develop, distribute,
and act as consultants for an automated CM system. The Digital Equipment Corporation has
voiced this problem in several of the past DECUS symposiums. This was further evidenced by
the presentation "LT086 HOW VMS DEVELOPMENT USES CMS" given at the Fall 1986
symposium in San Francisco where the VMS development team admitted an initial resistance to
using the DEC layered product CMS (Code Management System). The main objection raised was
that of "CMS cannot handle all of our needs." This raises the first and most important criteria for
any CM system to be developed:

    $ USER_INPUT == "ESTABLISH A USER TEAM"

User input – and establish a user team. All too often this is made into a trivial matter by the
developers of a CM project. Developers (and the author is ONE!) believe they KNOW what is
needed more than the CM managers who will eventually use the product. At TI we solved this
problem by establishing a User Team consisting of representatives from each division using the

CM product. The User Team is also the Configuration Control Board for the product – ensuring that CM is NOT a rubber stamp for the developers. Periodic meetings once every four to six weeks has been found to be the most cost/benefit effective for TI and a few other companies (having representatives in the DECUS L&T CM working group). Meetings on a more frequent basis than four weeks tends to have the developers spending more time preparing for the meetings than actually working. Meetings occurring less than every six weeks often indicates a failure in communication between the development/maintanence community and the user/future user community.

The User Team should be used to its utmost – a properly working User Team will provide the designer/developer with the input necessary to come up with a viable requirements analysis. At the same time – a good development group will realize the need for "designing through evolution". The requirements analysis should not freeze the design – unless the detected changes are major enough to cause massive overhauls to the preliminary and/or detail designs. Even when major changes are indicated, remember, a CM system, like any other, can be redesigned to death. As with all corporations, time and budgetary constraints must be taken into consideration.

The User Team is not finished when the product is released. Rather, its true emphasis is realized – a functioning User Team. The CCB that will control future changes to the product. Have the users rank and prioritize what are the system's major faults. What may seem to be an inconsequential cosmetic change to the developer may in fact have major impact upon the acceptance/use of the system. The involvement of the users also provides an excellent way for determining which sites are good candidates for being Alpha and Beta test sites.

DEFINING THE CM NEEDS

After establishing the User Team, the question is raised what functions are needed. To decide this, the best way is to define CM and the immediate impacts/needs of the CM managers. CM "is a discipline applying technical and administrative direction and surveillance to:

1. Identify and document the functional and physical characteristics
   of a configuration item.
2. Control changes to those characteristics, and
3. Record and report change processing and implementation status.

It includes:

- Configuration identification,
- Control,
- Status accounting, and
- Audits."

as defined by the Department of Defense (DOD) Military Standard 483A (4-Jun-1986). This definition will be expounded upon in a later paper by the DECUS L&T CM Working Group.

Indeed, what is distinct about the DOD's definition of CM is based on the definition of certain terms. This is characterized by the definition of Configuration Item, that being: "HARDWARE or SOFTWARE or an aggregation of both, which is by the contracting agency for configuration management (DOD-STD-2167)."

It may well be up to the User Team to decide whether your CM system will address solely software or a combination of software and hardware. At TI, the in-house Engineering Information Systems Hardware CM control is not only well documented and controlled, but well

designed and in place with no major problems existing. Therefore, there was no question that the system we were looking for was to be primarily for software. The User Team established was also targeted for the VAX series of computers. The following functions were defined as desired:

- TROUBLE REPORTING
- SOFTWARE RELEASE REQUESTS
- LIBRARY (Software/Documentation) MANAGEMENT
- SOFTWARE TRACKING
- REPORT GENERATOR
- VERSION DESCRIPTION DOCUMENTS
- SYSTEM MANAGER

The following restrictions, IAW (In Accordance With), were also to apply:

Military Standards
User Defined
Internal Use Only
Used on contracts (Government, sub-contractor, bid, etc.)

USER ACCEPTANCE

The biggest impact we've found implementing the User Team concept from the initial start has been the user receptiveness both prior to and during the delivery of the product. All groups had a vital interest not only in the need for the product, but felt that they had made major contributions towards the design of the CM product. As a result, the attitude developed was not "we're being forced to use this CM product", but rather, "the Advanced Computer Systems Laboratory is developing and maintaining a CM system for us, to our specifications."

SOFTWARE TROUBLE REPORT

The first item of importance for the User Team was the online tracking and submission of the STRs, or Software Trouble Reports. STRs are exactly what the name implies - Notification to the development team or maintenance group of perceived problems with a given system, whether hardware or software. STRs are also known as Software Performance Reports (SPRs - Digital), and Software Change Request (SCR - Mil-Standard nomenclature). Some groups may also use the form to accept requests for enhancements or upgrades to the documentation set.

Perhaps the most important aspect of any system is its user friendliness. The user friendliness of a system is not only based on the help screens and error checking capabilities of a system, but the positioning of fields in orders of their importance and input sequence. What has literally killed the acceptance of several systems is the positioning of optional or least likely to be used fields in the input sequence prior to a field(s) that is ALWAYS entered. If the development team is given the go ahead using SMG routines or some type of forms manager, the input sequence does not have to match the order layout of the screen ->>> Just remember, TRY NOT TO CONFUSE THE USER!

Given the understanding of user friendliness, the forms desired can be considered and layed out. Briefly, an STR needs the following information:

- Who
- When
- What

- How
- Solution
- Status

Each of the above information fields can be greatly expanded upon. Several systems – both manual and automatic literally have the Who being a full page/screen of information:

- Name
- Address
- Phone
- Organization

This information does not need, however, to go to the extreme of having the submitter enter information concerning the name and age of everyone in their family. The briefer the better. This also saves in disk/tape space.

The information may also need to be broken into several fields. If following Mil-Std 2167 associated Data Item Desriptions (DIDS), (in particular DI-MCCR-80009) the solution field is actually separated into two distinct fields – recommended solution and implementation solution. If going by some standard, whether government or company internal, then the content of the STR is easily determined – what is left is the determination of the layout of the STR.

In laying out the STR several factors have to be considered. User friendliness being, of course, the biggest. Standards will also have a large impact. Perhaps for TI, the biggest was the determination of having the STR consist of multiple pages versus a single page. Although having multiple pages reports takes up more room, the visual content was much easier to comprehend than the version where everything was jammed (literally!) onto one page. Another problem identified quite early is the mistake of having the screen representation and the printed representation being restricted to the same format. What is user friendly on the terminal screen might not be the best layout for a full page printed document. The same consideration is given for the break in the screen pages versus the break in the printed pages. Terminal wise, the password protected signoff fields on an STR are desired as the last page of the system (although the page can be quickly chosen through the STR main menu). On the STR printed reports the signoff signatures were found to be desired on the first page.

The question of separate or integrated database elements arose along with how to implement the database. For quick user acceptance, feedback, and deliveries, the development was done on a sub-system by sub-system approach. The databases are integrated in that each sub-system may reference the other's. The problem arose, however, with customer requirements and the fact the User Team spanned across several divisions each with different projects and cost factors. Requiring each installation to also have a database product/handler X was unfeasible. The agreed upon solution used RMS file structures. The drawback, of course, is that anytime a new database schema is arrived at conversion routines have to be written.

SOFTWARE RELEASE REQUEST

Associated with the Software Trouble Report (STR) is the Software Release Request (SRR). Ideally, every STR will at one time or another have an SRR associated with it. The Software Release Request is the formal document detailing the necessity for transferring software from a development area/account to the CM control libraries (directories).

Like the STR, the SRR provides an auditing methodology. For groups like DCAAs (Defense

Contracts Auditing Agency), the SRR provides an audit trail of when the software move was requested, who approved it, and when it occurred. With a fully automated system, records will be generated upon successful completion of the move stating that module X in directory Y was moved by the direction of SRR Z on a given date.

The SRR also provides project control. At TI, projects using the SCM system do not permit the movement of software without an SRR. The SCMS system also enforces this. Therefore, once again the user friendliness of the system is important. The minimal information needed for an SRR is:

- Name of requestor,
- Contact information (phone, organization, etc.),
- Modules to be released,
- Signature Approval block,
- Special instructions, and
- SRR status.

LIBRARY MANAGEMENT

Perhaps the most crucial part of the system will be the library management subsystem. The library management portion will control how software can be setup and where it will go. At TI, the Library Management System:

- Defines the project directory structures at the front end,
- Controls movement in and out of the control directories,
- Automatically processes file types, and
- Provides audit trails.

In defining the project directory structures, SCMS works very much like DEC CMS. Prior to depositing software into a control library, the SCM manager MUST create the directory path and substructure via the SCMS system. Data files vital to the system are then created along with the directories and software movement can then occur. Indeed, if the SRR has modules being moved into non-control directories, the system will provide the user with a warning message during the Verification phase of the SRR processing.

Defining the project directory structures is rather simple. Decide on a standard, ie. have the top directory node be the SCM owner account, the second being the CSCI (Computer Software Component Item, and the third being the TLCSC (Top Level Computer Software Component). Use even further divisions if necessary. The definition of reasonable being determined by the conditions of your contract obligations, SQA, or constraints implied by the use of any supporting software your system may utilize. An example of a constraint could be the use of the DEC layered product, CMS, where users have reported that when more than 250 files are in a CMS library, the CMS fetching, store, and reporting execution times become irritatingly slow. Project directories could then look like:

```
[SCM.CSCI1.IOCONTROL.IN]
[SCM.CSCI1.IOCONTROL.OUT]
[SCM.CSCI1.PROCESSOR]
[SCM.CSCI2]
```

If the control of the object files and executable images are desired, an additional sublayer of directories my be desireable to the project directories:

Directory for the general CMS source, Directory for the object and executable files, and Directory for the listings.

Additional directories may be desired for the different control phases the source/object/executable represents. For example: test and integration, development, formal release, cutomer deliverables, etc.

The automatic processing of file types needs to be decided once the directory structures have been established. Automatically processing file types is an area of contention even among several members of the DECUS CM Working group. If your organization can guarantee 100% that the code you get is always at least compilable your organization may not want this. Also the depth of the processing may not be open to debate. At TI, the latest SCMS release recompiles all the defined dependencies related to a file that is being released. The resulting process files are then moved into the appropriate directories. The automatic processing for files can be accomplished through several methods – the use of the DEC layered product MMS (which has compatibility with DEC CMS), UNIX MAKE (or MAKE-like utilities), some other third party software, or like TI, build yourself an inhouse tool to meet your needs if you can't find one that does within a "reasonable" price range.

Part of the controlling movement in and out of control directories has already been hinted at. Only certain people should be able to move files into the control directories. The debate of who can retrieve copies is open, and needs individual evaluation as to a project's/company's security needs. TI and several other companies in the CM Working Group have procedures (which may be enforced by written or electronic means) set in place where a given program has to be the mechanism through which the software is moved into control libraries.

Several companies require files being placed under control to be processed in a temporary directory. The file is placed into the control directories ONLY if processing occurs without any errors. This method is often used to avoid contamination of the control directories. The problem with such a scheme is the false sense of security a "controlled file" may give the user. This file is only guaranteed as "good" as the error detection mechanism, which in many cases is only a compiler and possible linking.

An automated system is a definite blessing for providing audit trails. With the SCMS system, TI has records generated for the movement of any file. The records indicate what module was moved, what SRR moved the module, or, for files which required recompilation because of an interdependency, which SRR caused the recompilation, and which directory the resulting process files were moved into. Deletion of files is also detected and recorded. The only way to remove evidence of a file's existance or deletion is to literally delete the whole history of a control library's file element.

CMS

DEC sells and maintains a layered product commonly called CMS, or Code Management System. This product allows the user to keep the entire code history of a file in ONE file. Additional records are entered into CMS "library" files which maintain the history records of module creation, copying, reserving, fetching in a LIBRARY history. This is unlike several versions of the UNIX based Source Change Control System (SCCS) where all references are maintained in the source file. The disadvantage being, of course, that the user cannot easily use the copy or rename commands to move the ENTIRE file (including history) from one CMS library to another – this causes corruption of both libraries and gets CMS to get "upset" at both libraries. A major advantage being that even if the file is deleted, the history for that file will still exist - provided of course, the user does NOT use the CMS DELETE HISTORY command. I personally swear by (and

sometimes at) CMS and have seen major improvements since the initial release. CMS is generally fast, except when the number of changes/modules in a given library reaches some undefined limit. CMS is also generally efficient. A novice user is likely to complain about the "extra" space CMS takes over a flat file for the first one or two generations of a file, but CMS can cause DRAMATIC disk space savings on keeping all versions of a file - especially when the changes to a file are relatively minor. CMS as of 2.2 is still targeted for ASCII text files. It does not work efficiently for object or executable files and may not return the file in a useable format if used for such, but then, CMS was not developed for such files (users always want the world!!).

MMS

DEC offers a product that will rebuild a system based on a combination of date stamps and a set of defined dependencies. The dependency sources CAN reside in a CMS library, although the last that I was aware of, the requirement existed that all the associated source files have to exist in the SAME CMS library unless the user went through some carefully thought-out CMS/MMS library manipulations. The syntax structure for MMS is very close to the UNIX based MAKE, and I'm told in many instances MAKE files can run under MMS (and vice-versa).

MMS is very powerful, but has the drawback of requiring image activation for each call to MMS, requiring some effort on the CM manager or developer to do the minimal amount of system rebuild (Yes, I know, this is also subject to debate). The debate concerns the option to rebuild the entire system every time, or just portions. If portions are to be rebuilt and an SRR using MMS has 15 modules, MMS is called and image activation occurs 15 times (some overhead occurs even if the system manager makes MMS memory resident). Of course, manual intervention might be made to ensure the minimal number of calls to MMS is made - to automate would require the developemnt of a system to understand the hierarchy of MMS described files, or to guarantee all the files affect the same system or group of executables. Taking this to an extreme, the user could be required to literally create a "mini-MMS". Someone might want to put on a wishlist for the ability to invoke MMS and stay in an MMS "native" mode until an exit or non-MMS command is used. The user could then give MMS a series of module names to act on. Better yet, keep the dependency structure around UNTIL the user selects another description file.

A nice feature is the option to shut off the date/time stamp checking, but this will cause the preprocessing of ALL files in the system for the given target. Because of the very nature of CM, every file movement causes the building of associated files, thereby usually eliminating the need for the date stamp check for a given file.

What I find particulary annoying is the requirement for specifying ALL dependencies from the executable to object to source — I've been spoiled by a system that lists "if this source file changes these source files are affected." The affected source modules may be includes, inherits, environments, compools, etc.

I do like MMS in that compared to other products on the market it is very easy to use. As mentioned earlier, for the user going to and from the IEEE/ANSI POSIX related OS's, it is very close to MAKE. Generally the user does not have to create convoluted MACROS, tables, or 3 million (a slight exaggeration) command files to perform a build.

My apologies to DEC if any of this has changed or is incorrect, but at the time of the San Francisco symposium (Fall 1986) this was the status of the two products to the best of my knowledge. Hopefully the silence of the DEC employees in the back of the room indicates my information is correct.

SOFTWARE TRACKING

Software Tracking provides three main functions:

1. Parts list
   - Listing components which make up specific versions of a system,
   - Represented in a manner to reflect module dependencies, and
   - Possibly provided by inputs from other systems (ie. MMS/UNIX MAKE files?)

2. Product Versions
   - Storing,
   - Tracking, and
   - Rebuilding.

3. Audit Trails
   - Records left for each version/movement - could be the parts list itself

The software tracking portion of CM can be a nightmare. Not only for the CM Manager as well. Often times non-government CM managers center themselves on what DOD-STD-2167 calls the baseline configuration allowing the developer or program manager to be responsible for the developmental configuration. A fully automated system will manage all libraries so that at any given time a previous of the system can be recalled, at least from the sources, and if necessary, rebuilt. There are several products that are supposed to do this on the commercial market.

Horror stories told by government personnel attending DECUS Symposia often have companies getting in trouble when their system "guru" leaves and no one else knows how to rebuild the system. A properly represented "parts list" not only solves this, but should meet government contracts following standards 483A and 2167A requiring representation of the system build. 2167A even encourages the representation to be stored electronically. The parts list can be an MMS or equivalent build file, but I would recommend some other form, possibly representing the hierarchy of the system through flow chart or indented summary explosion. By providing a visual hierarchy of the system, maintainability is improved, and identification of subparts for the Functional Configuration Audit (FCA) is usually enhanced. With the use of some simple I/O routines, the Physical Configuration Audit of the software can also be automated.

At TI, the SCMS system can build the parts list from MMS files or from user input through the "Relational Tracking System (RTS)". When tied with the use of CMS classes and groups, the user can establish a product version at any given time. SCMS has the planned future expansioin to use callable CMS to extract the generation/version information, providing us with the capability to state SRR X caused the following system configuration. That configuration will always be available as the parts list will have the generations/versions tied to it.

Tied with the library management system, the software tracking mechanism provides methods for "storing" the product version. The product version is stored when the updated parts list is automatically captured. Capture and storage of a file is not the same as capture and storage of a product version. If 25 versions exist in a library and noone has ever bothered to list how the 1500+ pieces of hardware and software relate to what version, you might find yourself never being able to recreate a given version. When files are moved into the control libraries, information should be passed through the database(s) permitting the options of stating "capture this product version as it now exists."

If you have the product version list detailing all module relationships and the corresponding generation/version numbers for that version, then you also have the capability not only to perform a rebuild, but to provide mass "checkout" capabilities for the developers.

REPORT GENERATOR

In addition to ensuring the integrity of code, the CM organization is often looked to for providing reports on the status of software. Any automated CM system needs to be able to access its data and produce reports which tie the different types of data into informative reports. The most commonly requested reports are therefore:

- Information by subsystem (STR, SRR, Library Management, etc.), and
- By data relationships (SRR x relates to STR 's y and z)

When initially starting, creating a system that produces one or two generic reports for each subsystem may meet the immediate needs. As use and acceptance of the system grows users will want more varying and dynamic reports. The SCMS system has two report generator "parameters":

1. Selection criteria, and
2. Sorting criteria

On the first, selection criteria, the user enters a screen where they can select to include or exclude the data to be used by a report. A smaller subset of the selection criteria is also used to sort the report by. Even with this flexibility, users still want more – each project wants to define what the report will look like in format, which leads to the next subject, "In what form?".

Defining the report format has been touched upon briefly during the explanation of STRs and SRRs. Of vast importance is the visual context. The more crowded a report is, the more chance of "user overload". I remember reading somewhere that like the advocates of structured programming state a context diagram should not have more than 7 to 9 items per chart, that a report page should have distinct areas of information and the number of distinct areas should not number over 7 to 9 areas. This allows for the scanning of reports. There needs to be a balance between the whitespace on a report and the data areas. Impacting on the report format should be the consideration of "Is this report to be viewed on the terminal or hardcopy". Whether the report is limited by 80 column or 132 column width is another important consideration. With the SCMS, the "hardcoded" STR/SRR formats number three:

1. SUMMARY – a listing of the reference number, the control levels, short problem description and the status,

2. ONE PAGE – a listing of the report giving the most requested information:
   STR/SRR number,
   Contact information,
   Status,
   Short problem description,
   Detailed problem description,
   Solution, and
   Signature sign offs.

3. COMPLETE REPORT – a listing of all of the information in the databse on that particular item.

The software tracking/library management subsystems have similar reporting capabilities in them.

One final question to consider on the reporting capabilities is the placement of the reporting capabilities. Should the report generation come from within each subsystem, the SCM system, or should the report generation system be a separate document system by itself (or maybe part of a larger documentation system)? TI chose to have the report generation system as part of the SCM system. The reason for this was that in using the RMS file structures, we were able to reuse the subsystem code for opening and closing files throughout the whole system. It was also easier to maintain. In a few instances, the individual subsystem has it's own reports that cannot be generated by the Report Generation Subsystem. If we had developed the system using a database handler, the approach would most likely had been different. When using database handlers (ie. FOCUS, DATATRIEVE) it is a lot easier to provide a core of generic routines from which the user can copy and modify to generate their own specific reports.

VERSION DESCRIPTION DOCUMENT

Although a form of report, the Version Description Document had enough requirements that it "deserved" its own subsystem. In particular, not only did the system need to be able to access the database and build the report, but because of the size and requirements of the document generated, there existed the need to edit subsections of the document.

Companies generating VDD's per Mil-Std 2167 can often spend weeks, even months generating the document. TI also had the unique requirement that due to waivers and internal requirements that almost every possible combination of the data available had to be included to or excluded from the document. The resulting VDD generator had two options therefore:

1. Mil-standard Version Description Document Generation, and
2. User defined

The user defined option provided users with a list of over 30 items from which to include into/exclude from the document.

The greatest benefit was that the VDD operated in modes – create and update. In the create mode, the database is scanned for data matching the selection criteria entered by the user and a VDD document file was created. In the update mode, the user had the option of updating via the database and/or editing sections of the VDD document file data. If used properly, SCMS provides a VDD requiring very little editing. What used to take 4 weeks to enter into VDD format now takes only 10 - 15 minutes, most of which the engineer can take a break during. After that, several hours, a day at the most should do.

SYSTEM MANAGER

Crucial to the success of an automated CM system is the existance of a control database to handle the security needs of a system designed to control the access into and out of project software. Basically, your system will need:

1.   Password Management : Unless you can guarantee electonically the identity of an individual signing off items, you'll need passwords for at least the signature signoff portions

of the STR and SRR subsystem. Additional password protected areas may be desired.

2. Verification Field Management : This serves two purposes – "dynamic" help for users and a database driven verification scheme for given input/selection fields.

3. Protection scheme for system manager database. This can be implemented by:

   - ACL lists,
   - File protection settings, and
   - Database encryption.

4. Selective Access : For individuals, groups, or the whole world. A few examples:

   - Read only,
   - limited Write,
   - full READ/WRITE capabilities

What may be overkill for one particular group/application may not be enough for another. Configuration Management need not be an road block to the development of software, but rather, with automation can be an invaluable partner towards the validation and safeguarding of project software.

The Languages and Tools Special Interest Group (L&T SIG) has an active Configuration Management Working Group which has manages to provide information and symposia sessions concerning topics involving CM. If you are interested in becoming involved in the working group or have any ideas, suggestions, please feel free to contact the Working Group Chair:

Mark Kidwell
Texas Instruments Incorporated   or E-Mail : "KIDWELL@EG.CSC.TI.COM"
PO Box 801 / MS 8006
McKinney, TX 75069   Phone : (214) 952 – 2058

PATCHING YOUR CODE WITH VAX DEBUG

by Jerry Oberle
Stamford, CT LUG Chair
Survey Sampling, Inc.

"Patching" means directly modifying the machine code in an executable image, or in memory. The VMS Patch Utility may be used to modify image files; it won't be discussed here. This article explains how to patch your program code or data using the VAX Symbolic Debugger, so that you can continue the execution of an errant program without having to leave the debugging environment. This saves the time spent correcting your source code, recompiling, relinking, and re-executing your program up to the point of failure. The techniques described here are not a panacea for all programming errors. Sometimes there is no practical choice but to recompile your source after making the necessary changes. But often, simple things can be fixed right in the debugger, if you know what to do. To use most of the techniques described herein, you'll have to be proficient in MACRO-32, the VAX/VMS assembly language. It is most important to note that any changes made to your code using the VAX Debugger affect only the working copy of your program in memory. Therefore, you must make corresponding changes in your source code, and recompile, etc. when you get the program fully debugged.

USING THE "SET TRACE ... DO (cmd-list)" COMMAND

This is the one technique which does not require fluency in MACRO-32. If you know the programming language that the source code is written in, you can use this method. The debugger command "SET TRACE address-expression" establishes a "tracepoint." Once set, each time the program reaches the tracepoint, Debug displays a message telling you that the program has just reached the statement or instruction being traced. In this respect, it is similar to the "breakpoint," except that Debug does not interrupt your program's execution as it does with breakpoints. Both the "SET TRACE ..." and the "SET BREAK ..." commands have some useful options. I won't discuss all the possibilities; you can consult the reference manual for that. The two that are useful in patching your program are the "/Silent" qualifier and the "DO (command list)" parameter.

The "Silent" qualifier on both of these commands tells Debug not to display a message on the terminal telling you that execution has reached the breakpoint or tracepoint. If used by itself with the "Set Break" command, this qualifier suppresses the message which tells you which breakpoint you've reached, but execution is still interrupted, and you still receive the "DBG>" prompt. If used by itself with the "Set Trace" command, it effectively renders the command useless.

The "DO (command-list)" parameter causes the debugger commands in "command-list" to be executed each time the breakpoint or tracepoint is reached. When used with "SET TRACE/SILENT", this parameter can be used to modify designated variables in your program each time the tracepoint is reached.

An example of this technique is given in the VAX/VMS Symbolic Debugger Reference manual. (I happen to have only a version 4.0 manual handy; in that manual, it's on page DBG-147.) The trick is to use the "Deposit" command in the command list to affect a variable which has been improperly computed by the program.

For example, suppose that your program is written in FORTRAN, and in a certain Do loop, you have the following statements (numbered 100 and 101, respectively, by the compiler):

J = N * 2   WRITE(1,99) Z(J)

While debugging, you realize that the first statement should be

$$J = N * 2 + 1$$

To correct this problem, use the following debugger command:

SET TRACE/SILENT %LINE 101 DO (DEPOSIT J = J + 1)

Note that the tracepoint must be set on the line AFTER the one which computes J. This is because tracepoints (and breakpoints) take effect immediately BEFORE execution of the specified statement or instruction.

REPLACING VAX INSTRUCTIONS

To use this technique, you must be able to understand the VAX assembly language, MACRO-32. Of course, if your program is written in MACRO, then it can be fairly well assumed that you know it. But if your program is written in FORTRAN, C, or any other high level language, you can still apply this technique.

A situation in which this technique might be used is a FORTRAN program that doesn't have an explicit declaration for a system service. For example, consider the following code:

ISTAT = SYS$CREMBX (...)
IF (.NOT. ISTAT) CALL LIB$SIGNAL(%Val(ISTAT))

The compiler would treat Sys$Crembx as a "real function." Therefore, it would generate code to convert the presumably F_floating return value in register zero into a longword integer. Typical generated code might look like this:

CALLG   B^20(R11),@#SYS$CREMBX
CVTFL   R0,B^0FC(R11)

When executed, this sequence of instructions will most likely signal a bizarre error condition. This problem can be corrected by replacing the CVTFL instruction with a corresponding MOVL instruction, using the debugger command:

DEPOSIT/INSTRUCTION %LINE ??? + ?? = "MOVL R0,B^0FC(R11)"

Note that this technique can only be used when the new instruction(s) are not longer than the instructions they replace. In this example, the new instruction was exactly the same length as the instruction it replaced, so nothing special had to be done. However, when replacing a longer instruction with a shorter one, "NOP" instructions would have to be added to fill up the intervening space.

For example, suppose your program included the following statement:

K = L + 1

In this case, you determine that K should simply be set equal to L. Assume that both K and L are longwords. The generated machine code might look something like this:

ADDL3 S^#1,W^104(R11),W^200(R11)

You could replace this instruction with the following one:

MOVL W^104(R11),W^200(R11)

However, the new instruction is one byte shorter than the old one. If you don't put in a "NOP" instruction following your new instruction, the VAX will try to interpret the last byte of the old instruction as the op-code of the "next" instruction. That would produce disasterous results in this case, since the byte being left behind contains hex 02, the opcode for "return from interrupt."

The debugger commands to perform this operation are:

```
DEPOSIT/INSTRUCTION %LINE ??? = "MOVL W^104(R11),W^200(R11)"
EXAMINE/INSTRUCTION .
    {debugger displays the instruction just entered}
EXAMINE/INSTRUCTION
    {debugger displays the "next" bogus instruction}
DEPOSIT/INSTRUCTION . = "NOP"
```

The reason for using the sequence of examine instructions before the second deposit is to force the debugger to re-evaluate the address of the logical successor to the current entity. If you do not do this, the debugger puts the second instruction at the address where the instruction following the original instruction begins. (In other words, it puts it in the wrong place.) (Of course, in this particular case, you could avoid this problem by using zero as the literal value in the original instruction. Such are the problems with contrived examples.)

ADDING NEW VAX INSTRUCTIONS IN A PATCH AREA

Once in a while, you might determine that you need to modify your code in such a way that the new, replacement instructions require more space than the old instructions they replace. In order to insert the new instruction sequence, you must allocate a "patch area" in which to write the new instructions. At the point in the original program where you wish to execute the new instructions, you must jump to the patch area. Likewise, at the end of the new sequence of instructions, you must jump back to the original instruction stream. This can be done with JSB/RSB instructions, as long as you pay careful attention to what is on the stack, or you can hard code JMP instructions at each place.

The tricky thing about this technique is allocating the patch area, since VAX Debug does not directly support doing this sort of thing. You'll need to find a spare longword somewhere in your program's address space first. A good place to look might be some variable that is about to be overwritten, or perhaps a filler area in a data record. If all else fails, you can use any writable address by just copying down its contents on paper and restoring it to its original state when you're through.

Once you've found a spare longword, call the Run-time library procedure LIB$GET_VM_PAGE to allocate a page or two of memory, using the debug commands:

```
SET MODULE SHARE$LIBRTL
CALL LIB$GET_VM_PAGE (%REF(2), %ADDR(spare_longword))
```

The debugger should respond with the message: "value returned is 00000001". If the return value is an even number, of course, something is wrong. (SHARE$LIBRTL will always be present, because the debugger itself calls the Run-time Library's memory management routines to allocate space for its symbol tables.)

Assuming all is well with the call to LIB$GET_VM_PAGE, your spare longword will contain the address of two pages of memory that you can use as a patch area. Assign a symbol to this address so you don't need to remember what it is. For example,

```
DBG> CALL LIB$GET_VM_PAGE (%REF(2), %ADDR(I+4))
value returned is 00000001
DBG> EXAMINE I + 4
00000484: 00139600
DBG> DEFINE/ADDRESS PATCH_AREA = 00139600
```

I would recommend allocating two pages of memory before even starting your program, unless the program is very memory intensive, just to have the space available. If you do this before the program begins, you'll also find that you generally have more variables you can use to temporarily store the address of the patch area. Two pages should be ample space to patch several sections of code.

The program below will be used to demonstrate this technique. A new expression will be used to compute the value for I, replacing the computation at line 4. (The line numbers are generated by the compiler, of course; they are not part of the program source code.)

```
0001  9       FORMAT(1X,'*',$)
0002          TYPE 9
0003          ACCEPT *, I
0004          I = I * 2 + 1
0005          TYPE *, I
0006          END
```

The first step is to allocate a 1K block of memory for additional instructions, as described above. Then, step through the program until line 4 is reached, and examine the machine code.

```
DBG> EXAMINE/INSTRUCTION %LINE 4 : %LINE 5
DEMO$MAINLINE 4:        MULL3      S^#02,(R11),R0
DEMO$MAINLINE 4+4:      ADDL3      S^#01,R0,AP
```

From this sequence of instructions, it is apparent that the current value of I is at the location pointed to by register 11, and that the result of the computation is placed in register 12 (the argument pointer). These two instructions will be replaced by a longer sequence that effectively computes I as follows:

$$I = (256 - I) * 2 + 1$$

The new sequence of instructions are placed at the beginning of the patch area with the following commands:

```
DBG> DEPOSIT/INSTRUCTION PATCH_AREA = "SUBL3 (R11),I^#100,R0"
DBG> EXAMINE/INSTRUCTION .
00139600: SUBL3 (R11),I^#00000100,R0
DBG> EXAMINE/INSTRUCTION
00139608: HALT
DBG> DEPOSIT/INSTRUCTION . = "ADDL2 R0,R0"
DBG> EXAMINE/INSTRUCTION .
00139608: ADDL2 R0,R0
DBG> EXAMINE/INSTRUCTION
0013960B: HALT
DBG> DEPOSIT/INSTRUCTION . = "ADDL3 S^#01,R0,AP"
DBG> EXAMINE/INSTRUCTION .
0013960B: ADDL3 S^#01,R0,AP
DBG> EXAMINE/INSTRUCTION
0013960F: HALT
```

```
DBG> DEPOSIT/INSTRUCTION . = "RSB"
DBG> EXAMINE/INSTRUCTION PATCH_AREA : 0013960F
00139600: SUBL3 (R11),I^#00000100,R0
00139608: ADDL2 R0,R0
0013960B: ADDL3 S^#01,R0,AP
0013960F: RSB
```

Next, the original code at line 4 is altered to branch to the new code in the patch area. Since the replacement code ends with a RSB instruction, a JSB is used to reach it.

```
DBG> DEPOSIT/INSTRUCTION %LINE 4 = "JSB L^PATCH_AREA"
DBG> DEPOSIT/INSTRUCTION %LINE 4 + 6 = "NOP"
DBG> DEPOSIT/INSTRUCTION %LINE 4 + 7 = "NOP"
```

Notice that since the JSB to the patch routine is shorter than the original code at line 4, the extra bytes are filled with NOP instructions. In effect, the original in-line code has been replaced with a shorter sequence that calls the subroutine in the patch area.

SUMMARY

The VMS Symbolic Debugger can provides skillful programmers with the tools to not only locate programming errors, but to correct many of them "on the fly." This allows the programmer to catch several bugs in one debugging session, rather than having to back and recompile again for each error uncovered. Naturally, some judgment is required to decide whether to correct the source code and recompile, or whether to modify a program in the debugging environment.

The foregoing techniques are useful not only to correct programming errors. Often, seldom used code paths (such as error handling routines) are difficult to test using the conventional approach of running the program with selected test data. Instead of changing the source code to force execution of a particular path, you can modify the program within the debugger so that the desired routines are executed.

LANGUAGES AND TOOLS SIG WISHLIST

ADA   **Name: Lisa Kerby-Rogers**   **ID 44**
A DEC UNIX Ada compiler

ADA   **Name: Lisa Kerby-Rogers**   **ID 45**
Strongly typed support in interface packages, i.e., system services STARLET package.

The STARLET interface uses base types does not use enough subtypes, i.e., range constraints. There seems to be a global use of system.unsigned_xxx. To support the Ada programming practices, most, if not all, should be subtypes of appropriate base types.

ADA   **Name: Lisa Kerby-Rogers**   **ID 46**
Support for DEC's counted string type.A predefined counted_string type might be useful for such applications as mailboxes, where VMS expects a counted string. This would help developers, since they often don't use the field(s), but still must account for it in the structure.

ADA   **Name: Lisa Kerby-Rogers**   **ID 47**
Separate windows in the debugger for each Ada task.

This may be possible only on a VAXstation, but it would be nice if each taqsk, when activated, created a separate window when run with the debugger.

ADA   **Name: Lisa Kerby-Rogers**   **ID 48**
CDD interface to Ada and other languages.

Allow for definitions (structures) in CDD to be accessed by all languages.

ADA   **Name: Lisa Kerby-Rogers**   **ID 49**
Cross-compilation to other languages from Ada.

This was from a working group session. Unfortunately, I don't know what languages they had in mind.

ADA   **Name: Lisa Kerby-Rogers**   **ID 50**
Please let us know whether or not DEC has plans to implement Ada tasking on multiprocessors.

I was surprised to find support for other languages via PPL when Ada's tasking was designed into the language to support multiple processors. Not one will tell me (and several others who asked) what the status is at Digital regarding this matter.

ADA   **Name: Gus Altobello**   **ID 52**
Ada should use multiple processes for tasking, rather than simulate this within a single process.

Ada now implements tasking within a single process. With the advent of multiprocessor VAXen, this becomes a liability.If Ada would compile each task into a process, the language would naturally lend itself to parallelism on new VAX machines.Under the present conditions, parallel processing in Ada can only be implemented in a roundabout and non-portable manner.

ADA   Name: George L. Scott   ID 53
Integrate Ada with CMS.

Let ACS pull elements from and update elements into a CMS library.

ADA   Name: George L. Scott   ID 54
Cross-compilers and cross-systems test tools.

Tools for development, test, and integration of systems targeted to other computers, including a test driver that will drive a test on another computer from a VAX.

ADA E   Name: James Ayers   ID 51
Integrated CASE tools.

Provide enhanced set of products for the entire software life cycle.Front-end design analysis and prototyping tools, integrated with VAXset, would provide a valuable and profitable capability which no vendor has yet provided.

ALL   Name: Mark Kidwell   ID 42
Provide DEC sales reps with video cassette presentations on L&T products for the customer to view.

Provide sales reps with video cassetes of products. This will permit the customer to possibly let actual users of the product view the product and determine its productivity impact on a given project. Currently, managers often decide not to buy VAXset and other products because only a small subset of management sees the SPD's and promotional material. If end users were able to see integrated examples, their input will also be involved in the decision.

ALL   Name: Chet Small   ID 43
I'd like to see DEC pursue the cross-development area.They will loose the market to the PC world. They must be competitive in price. PLEASE!!

APL   Name: Karlton J. Hickey   ID 55
Improve Type II array performance.

Currently display of Type II or nested arrays is very slow. While the use of these arrays in code may not be as slow, the slow display creates the impression that the Type II arrays are inefficient.

APL   Name: Karlton J. Hickey   ID 56
Enhance APL user interface.

Continue to enhance and improve the user interface. Short term, a full DCL style CLE in immediate mode would be useful. Longer term, a window-oriented interface based on DEC- or X-windows would be useful.This might include the following:
    - Multibuffer/window editing of mult. functions simultaneously
    - Immediate mode in one window, function editing in another
    - Copy buffer to buffer: allow a user to quickly copy code or
      prototyped command into a function.

BASIC Dialect: All   Name: BASIC Wish List Session   ID 15
Logical OR, AND -- stop checking parts of condition sooner.

BASIC   Dialect: VAX   Name: BASIC Wish List Session   ID 16
Make the examples in the documentation easier to read.

BASIC   Dialect: VAX   Name: BASIC Wish List Session   ID 17
Faster procedure activation.

BASIC   Dialect: VAX   Name: BASIC Wish List Session   ID 18
Explicit fixed length string [presumably other than mapped]

BASIC   Dialect: VAX   Name: BASIC Wish List Session   ID 19
Better integrated screen editor.

Many micro BASIC's have a screen editor integrated with the environment. I rarely use the
environment any more because the editing is too difficult. What would be great is an environment
that combines the syntax-checking and immediate mode of the current environment with the
screen-editing capabilities of TPU. Maybe two windows.

BASIC   Dialect: BASIC-Plus   Name: BASIC Wish List Session   ID 20
Remove backslash and ampersand.

BASIC   Dialect: VAX & BP2   Name: BASIC Wish List Session   ID 21
Warning for line number out of order

BASIC   Dialect: VAX & BP2   Name: BASIC Wish List Session   ID 22
Optimize generated code

BASIC   Dialect: VAX   Name: BASIC Wish List Session   ID 23
Pointers and dynamic allocation, so linked lists will be possible.

BASIC   Dialect: VAX   Name: BASIC Wish List Session   ID 24
Pointers in VAX BASIC for things like linked lists.

BASIC   Dialect: VAX   Name: BASIC Wish List Session   ID 25
Handle CDD date fields properly.

BASIC   Dialect: VAX   Name: BASIC Wish List Session   ID 26
Simpler way to zero variables.

BASIC   Dialect: VAX,BP,BP2   Name: BASIC Wish List Session   ID 27
Better runtime optimization. Maybe optimizing pass forcompiler.

BASIC   Dialect: BP,BP2,VAX   Name: BASIC Wish List Session   ID 28
Pointer data types

BASIC   **Name: BASIC Wish List Session**   **ID 29**

Looping structures, BEGIN..END for multiple statements.

BASIC   **Dialect: BASIC-Plus-2**   **Name: Kelvin Smith**   **ID 3**

In the debugger, a way to print out all 16 significant digits of a floating-point number.

The BP2 debugger PRINT statement shows only the six most significant digits. Frequently one needs to see the last couple of digits. This requires PRINT USING and recompiling; i.e., not using the debugger. An extended PRINT statement would help enormously.

BASIC   **Dialect: VAX**   **Name: Mike Wheeler**   **ID 57**

Pointers in VAX BASIC, so things like linked lists will be possible.

I would like to be able to dynamically allocate a specified number of bytes and have a pointer returned to me. And be able to use pointers to build linked lists.

BASIC   **Dialect: BP2**   **Name: Kelvin Smith**   **ID 8**

Debugger — break the nth time one reaches a breakpoint.

Frequently one needs to check a program after it has gone through a loop a number of times, but still inside the loop (for example, iterations 49 and 50 of FOR I = 1 TO 50. It would be useful to be able to specify breaking the nth time. Possible syntax: BREAK 1200.3(5) where(5) indicates stopping the fifth time one reaches line 1200, statement 3.

BASIC   **Dialect: BP2**   **Name: Kelvin Smith**   **ID 9**

Load debugger properly when chaining to a line number.

In BP2 V1.6 (RSTS), the debugger was properly loaded when one chained to a program with or without a line number. In V2, the debugger does not get loaded when chaining to a line number. My SPR was acknowledged, but no fix assured. The workaround of putting a GOTO at the beginning of every program is awkward, since it requires modifying both calling and receiving programs.

C   **Name: Louise Wholey**   **ID 58**

Services for a C programmer to convert strings from C to VMS descriptors and back to C string format.

C   **Name: M. Erik Husby**   **ID 59**

A #pragma that turns portability checking on and off.

We are developing C code to run in multiple environments: VMS, MS-DOS, UNIX, and want to ensure that our code is portable by flagging non-portable constructs. If one turns on the VAX C portability checking switch, a lot of the VAX C .H files generate warnings. It is hard to filter out those warnings from those in our code. H files are where we want nonportable constructs isolated, so a #PRAGMA to turn checking on and off would reduce the number of extraneous messages.

CASE   **Name: Donna C. Hall**   **ID 32**

Hierarchical data dictionary; tools for obj-or analysis and design; etc.

For CASE industry, not just DEC.1. Hierarchical data dictionary. The CASE tools we've looked at have a flat database. Data entity names are global and must be unique. This is contrary to our

philosophy of information hiding, encapsulation.

CASE   **Name: Donna C. Hall**   **ID 33**
Tools for object-oriented analysis, design, and programming.

Current CASE tools for SA and SD are based on methodologies of the 70's and early 80's: i.e., top-down functional decomposition. We have beenusing object-oriented design, programming techniques, and O-O languages for 4+ years. We don't want to take a step backward just to use an automated SA, SD tool.

CASE   **Name: Donna C. Hall**   **ID 34**
Documentation system integrating text and graphics online.

Documentation system capable of integrating text and graphics online as opposed to integration on hardcopy preview.

CASE   **Name: Donna C. Hall**   **ID 35**
Configuration mgmt. system to support all phases of the s/w life cycle.

Configuration mgmt. system used to support all phases of the software life cycle. The CM system must be able to control source, binary libraries, as well as support a hierarchical file structure. We'd like to control the environment in addition to the source and object files.

CASE   **Name: Donna C. Hall**   **ID 36**
CASE tools which support simulation of analysis and design tools.

We have a CASE tool in-house which supports simulation of design models, but we are simulating our analysis models by hand and by using an internally developed object-oriented language.

CASE   **Name: Donna C. Hall**   **ID 37**
Project management tool fully integrated with tools supporting each life cycle phase.

CASE   **Name: Donna C. Hall**   **ID 38**
Code generators off the back end of the str. design CASE tools, where user can define the language.

Not everyone is working in Ada or C.

CASE   **Name: Donna C. Hall**   **ID 39**
Support for graphic output to large plotters.

All but one CASE vendor we have talked with think it's acceptable for software engineers to tape sheets of paper together. CAD/CAM vendors and users would not find this acceptable. Why should the software engineering industry be told this level of support is enough.

CASE   **Name: Donna C. Hall**   **ID 40**
A standard interface between CASE tools.

CASE   **Name: Joseph A. Pollizzi**   **ID 60**
That new CASE tools consider that projects at any phase can be integrated into the tool set.

CASE   **Name: Stan Schultes**   **ID 61**

Broader definition, strategy statements, and implementation plans on tools integration across CASE spectrum.

This would be helpful for DECUS members from DEC. Seamless integration of tools is a must. Development cycle must be integrated with tools, from design, development, test, maintenance, and thru technology transfer, including software and documentation, and interfaces to everything outside.Idea: maintain a development cycle database used as common basis for tools, with standard interfaces for users and each other.DEC should provide DECUS with a better statement of their direction

CASE   **Name: Mike Berman**   **ID 62**

Uniform user interfaces among DEC tools.

DEC tools lack uniformity. Often different tools with similar functions have different implementation. For example, DEBUG has KP2 for scroll down, KP8 for scroll up. But in VAXnotes, KP2 is down, KP5 is up.Command recall and keypad layouts should be similar in all tools. This would shorted the learning period and increase productivity.

CASE   **Name: Bryan Taylor**   **ID 63**

CASE tool integration--standard tool interfaces

Currently we have bits and pieces of CASE development environment. All the vendors blythely ignore each other and even their own products in their rush to be CASE vendors. There has to be a set of standard interfaces between tool classes if we are going to move in the direction of integration.

CDD E   **Name: Shirley Bockstahler-Brandt**   **ID 64**

Better integration of CDD and the VAX languages

Languages are becoming data-rich. Ada offers structures similar to DBMS's. Yet CDDL has not kept up. Those who work in multi-language environments need the data-sharing assistance CDD offers.For example, symbolic constants, initial values. Languages should provide better translation, more qualifiers (like /NOINFORMATION_MESSAGES). SCAN and OPS5 can help in translation.

CDDL   **Name: Shirley Bockstahler-Brandt**   **ID 65**

Make CDDL a language.

We would like to share data definitions amoung VMS products, especially languages. This is important for consistency in a multi-language program. CDDL is inadequate, even given incompatibilities of data structures among languages. We need to be able to really describe our data in the CDD and then access it directly from the languages. Examples: symbolic constants and default values.

CMS   **Name: George L. Scott**   **ID 12**

Want problem report tracker with automatic cross-checking to changes entered thru CMS.

CMS   **Name: John Isakson**   **ID 13**

Remote CMS -- ability to use CMS across network, via DECnet or DECnet-DOS.

CMS   Name: Kevin L. Lundeen   ID 164
Segregated reference copy directives.

The V3 feature that automatically creates subdirectories is nice, but it still leaves overpopulated reference directories.We would like this feature in two ways. - Group: ref. directory mapping - Reference directory for a variant line of descent, although I don't know of any way to currently specify this. Classdoesn't work because it doesn't include future generations along the variants.

CMS   Name: Kevin L. Lundeen   ID 165
$CMS search elem-expr/gen-expr string with DCL search qualifiers: /match=and,or, /window=, /output, etc.

I want a DCL level search to go through CMS elements, e.g., search the most recent generations in one particular group for the string "foo". Would be nice in LSE, too!

CMS   Name: Kevin L. Lundeen   ID 166
A way to see the deleted lines in an ANNOTATE/FULL output.

CMS   Name: Kevin L. Lundeen   ID 167
$CMS SHOW HISTORY/SINCE=class-expr

CMS   Name: Kevin L. Lundeen   ID 168
$CMS SHOW GROUP/CONTENTS=elem-expr$CMS SHOW CLASS/CONTENTS=elem-expr

E.g., I want to see all of the *.H elements in class "V3.2".

CMS   Name: Chris Carroll   ID 66
String search command similar to the VMS search utility to use in CMS libraries.

CMS   Name: Debra Scarbrough   ID 67
Ability to search with a text string in CMS libraries.

Implement a string search thru an entire CMS library, selecting a specific generation.

CMS   Name: Mark Kidwell   ID 68
CMS needs to work across the net.

CMS 3.0 acl's should be able to handle proxy CMS accesses for networking capabilities.

CMS   Name: John Isakson   ID 69
Remote CMS -- ability to use CMS across network via DECnet or DECnet DOS.

CMS   Name: George L Scott   ID 70
Problem report tracker with automatic cross-checker to changes entered thru CMS.a

Provide a problem report tracker, including reports, user entry approvals. Include CMS addtions or shell that requires problem report number or CMS REPLACE annotation. Include cross-check with problem database to verify existence and approval of all numbers used. May include automatic status change in database.WG chair addendum: if this is not possible, what about to existing DEC SPR system?

CMS   **Name: Susan Wolcott**   **ID 71**
More comprehensive release notes

Release notes now say 'Bug in such and such was fixed.' and 'other bugs were fixed.' We have no way of knowing if our bugs were fixed and we have to write test programs to find out. For problems such as eating up and never releasing virtual memory, this may not be quick to test. We need a list of all bugs fixed, with description of how that bug manifested itself, so we can identify ours.

CMS   **Name: Susan Wolcott**   **ID 72**
CMS search command

search command and causing lock problems. Need to be able to find all elements containing a string.

CMS   **Name: Don Gummow**   **ID 73**
Option on CMS$SET_LIBRARY to create or supersede the CMS$LIB logical name.

While it's not hard to use LIB$SET_LOGICAL, it would be convenient to have this as an optional option of the SET LIBRARY command.

CMS   **Name: Donna C. Hall**   **ID 74**
Support for hierarchical file structure.

We now either flatten our source structure when putting it under CMS control or else we place a CMS library at every level in our project directory structure.There is a reason for the way the software is structured, and that architecture should be preserved by the CMS system.

COBOL   **Name: Robert G. Ribokas**   **ID 30**
Ability to sort USING/GIVING a working-storage data structure (table).

It would be nice to be able to specify a data structure in working storage that contains a table as the USING/GIVING source/object in the SORT command. It would save programmingtime wasted by having to write INPUT/OUTPUT procedures, or having to write the table to a file and then specify the file as the USING/GIVING object.

DCL   **Name: John A. Ektermanis**   **ID 75**
DCL TYPE and/or COPY option

Capability to copy or type subset of file. Maybe "COPY fil1(n1,n2) fil2"; or TYPE fil(n1,n2);where n1 is starting record # and n2 is either number of records or ending record number.Restrictions: typable file for TYPE.Or maybe: COPY/RANGE=(n1,n2) fil1 fil2 TYPE/RANGE=(n1,n2) fil1

DEBUG   **Name: Kevin L. Lundeen**   **ID 174**
Automatic SET SOURCE to DBG$SOURCE just like LSE.

DEBUG   **Name: Jim Bassich**   **ID 76**
Document the DEBUG interface.

DEBUG   Name: George L. Scott   ID 77
Remote DEBUG interface for cross-compilers.

Define DEUG interface so that execution of cross-compiler output can be debugged. VAX drives I/O lines to target computer, client-provided interface in target control test there, VAX is primary control and results analyzer.

DEBUG   Name: Don Gummow   ID 78
Set language automatically when you go from a routine in one language to a routine in another.

May also want way to disable automatic SET LANGUAGE in some circumstances, such as in MACRO.We often program in multiple languages, and it's annoying to step from FORTRAN to C and get "Misplaced operator" when typing "EXAMINE *V".

DEBUG   Name: Teri McNamara   ID 79
Tools for software testing and analyzing non-DEC target software.

Ability to use DEBUG, PCA, DTM so software built with cross tools could be simulated/tested/debugged to some level on VAX before being sent to the target environment.

DEBUG   Name: Paul Plum   ID 81
Integrate CDD and SQL with DEBUG, LSE, and SCA.

DESIGN   Name: Chet Small   ID 80
Provide tools for low-level work, such as compilers, assemblers, debuggers. Need tools for design and consistency.

DIBOL   Name: Roger Nelson   ID 82
Multi-dimensional arrays on RSTS and for Pro Toolkit.

VMS DIBOL allows these, I think with a GROUP in the data section.This would be nice under RSTS, and on Prod/350's.

DIBOL   Name: Roger Nelson   ID 83
Backward search on an RMS indexed file.

This is for RSTS. Current work-around is to maintain an extra key, with same key value, subtracted from 9999. I'd like to be able to get rid of the extra key.

DTM   Name: Kevin L. Lundeen   ID 171
Run time logical name translation.

Currently does full translation at object creation time. I move the files and DTM is useless.

DTM   Name: Kevin L. Lundeen   ID 172
Some way to get the PCA command SET DATAFILE/SHARE in order to do coverage on shareable images.

The current work-around is to edit the DTM scripts.

DTM   Name: Kevin L. Lundeen   ID 173
Some way to turn off creation of (??)abrule(??) screens files.They are gigantic and we don't use them.

DTM   **Name: David Powell**   **ID 84**

Need ability to access the history file by collection title.

DTM   **Name: David Powell**   **ID 85**

Logical names resolved at run time instead of when stored.

Logical are not very useful, because they are resolved when stored, i.e., the template directory logical name is resolved and then stored. When I changed disks, all my session files were not able to be found, as were the benchmarks. Want logical names stored as logicals, and resolved at run time.

EVE   **Name: Mike Berman**   **ID 31**

Easier handling of buffers

1. If a buffer name is not unique, selection of the intended buffer should be directly from the screen (menu-like).

2. When getting files, if a logical name or directory is used but is incorrect, an empty buffer is created with that name. Recalling the GET command will require that a new buffer name be entered. There should be a yes/no option to overwrite the existing buffer with that name on a GET.

FMS   **Name: Larry Burgh**   **ID 86**

Enhancements: VMS-like form test program, alter field attributes, and user action routines. Also use of Ctrl/Z.

Want VMS features for RSX-11M+. 1) Form test utility; 2) alter field attributes command for more screen control; 3) user action routines for data validation; 4) ability to return Control-Z as a field terminator for use in running FMS from older terminals such as VT100. VT100 user could signal that input is to be aborted.

FORTRAN   **Name: Valerian Nita**   **ID 87**

RDBPRE/FORTRAN should work with FORTRAN files that contain more than one subroutine.

FORTRAN   **Name: Luther Atkinson**   **ID 88**

FORTRAN file record size in open statement preprocessors.

In OPEN(...), if form = FORMATTED, use the number of bytes of the actual record. If form = UNFORMATTED, then use n = 1/8 the number of bytes in the actual record. Why can't I just use the number of bytes regardless of form value? Preprocessors Ratfor-like for resequencing format and statement labels and formatting for readability and in-line documentation.

FORTRAN   **Name: Bryan Taylor**   **ID 89**

Qualifier to compiler to invoke module-level variable, label, and dummy argument usage checking.

FORTRAN-LINT from IPT Corp. performs intra- and inter-module static analysis (a.k.a. lint picking). SCA can do the argument list checking of FLINT, but DEC has no comparable local module level usage checking. this includes identification of unused variables, labels, and arguments, referenced but not set local variables. We use FLINT to clean up code and catch errors. GRC produces RXVPSO which also does this local checking. It would be a nice addition to the compiler.

FORTRAN   Name: Douglas McCune   ID 90
Quote delimiter mismatch error message when compiling FORMAT statements.

Currently: 1000 FORMAT (' This is a test) (note missing quote) -- generates a two-line error message for each character in the intended quotes string, which is rather messy.

FORTRAN   Name: Steve Goldberg   ID 91
A little more documentation of examples re lock manager and different communication possibilities for task to task.

FORTRAN   Name: David Jones   ID 92
Improve output formats.

1) Support integer data type with implied decimal which would output the decimal. E.g., "I8.2" format with data 12345 would output " 123.45". 2) Support new date formats in Version 5 of RTL in output statement formats, i.e., D13 or D32 format for new data types for LIB$CNV_DATE_.

L&T I   Name: Mark Kidwell   ID 41
Install streams need to verify that systems files updated are correctly placed.

L&T Install streams need to verify that system files that are updated do not have misplaced copies existing in other system/cluster directories that might overlay the actions of the install procedure. An error/informational message could be given to the system installer for correction or later action. Doing this will further insure the integrity of the system and install process.

LSE   Name: Kevin L. Lundeen   ID 155
EMACS-like features: mark stacks, kill rings, command completion, bring text from buffer down to prompt.

We've done all this in TPU, but it would be great if it shipped with LSE. I have more! Just give me a call (617-423-3500).

LSE   Name: Kevin L. Lundeen   ID 157
Translate unprintable backward question marks to octal or hex, or provide command which tells what the character is.

LSE   Name: Kevin L. Lundeen   ID 158
Command completion.

Example: In command line in LSE, type S and press Control-E. LSE puts up a menu of all commands starting with S, including user-defined commands. Another example is filename completion. LSE> GOTO FILE x <Control-E> either completes the filename, if complete, or gives a menu of all matching files. Editor's note: Maybe wildcards, too? E.g.: A*BC

LSE   Name: Kevin L. Lundeen   ID 159
Allow wildcard filespecs to read multiple files in, one per buffer, both at the DCL command and within LSE.

EXAMPLES: $ LSE *.C   LSE> GO FILE B*.C

LSE   **Name: Kevin L. Lundeen**   **ID 160**
Shareable environment files, so we can install them like the TPU section files.

LSE   **Name: Kevin L. Lundeen**   **ID 161**
LSEDIT cms-element should set LSE$CURRENT_FILE.

LSE   **Name: Kevin L. Lundeen**   **ID 177**
Full DCL support, including that nasty "$".

LSE   **Name: Kevin L. Lundeen**   **ID 179**
RESERVE /OUTPUT= qualifier

Don't check out the file into the current directory. We have code that assumes that reserved copies are all in a particular place. We can't enforce that now.

LSE   **Name: Kevin L. Lundeen**   **ID 180**
Save cursor position with RESERVE. Even approximate is great!

LSE   **Name: Kevin L. Lundeen**   **ID 181**
Multiple line recall.

LSE   **Name: Kevin L. Lundeen**   **ID 182**
Accept input before the end of multiple screen help (like DCL). (Editor's note: and EVE!)

LSE   **Name: Kevin L. Lundeen**   **ID 184**
$ LSEDIT cms-element should set LSE$CURRENT_FILE.

LSE   **Name: Lindsey Todd**   **ID 93**
Command line recall; better control of packages.

1) LSE should allow READ_LINE to recall more than one line. I built this for LSE and it gets messy. However, composed lines like SMG$READ_COMPOSED_LINE is probably going overboard; not asking for this! 2) I want tokens to expand into menus of routines, in a form that works across languages. Example: token SMG, expand into menu of SMG routines, or maybe classes that expand into routines. I don't want to define this manually....

LSE   **Name: John Schmidt**   **ID 94**
Expand LSE to auto-gen Rdb precompiler shells and capability of pre-compile as it can do for VAX BASIC.

LSE   **Name: M. Eric Husby**   **ID 95**
Enhance to allow calling a TPU procedure when expanding a token.

Example: DEFINE TOKEN DATE/EXPAND=INSERT_DATE procedure insert_date copy_text(fao("!%D", 0)); endprocedure

LSE   **Name: Tritt Graham**   **ID 96**
More tools for concept, specification, and design phases--based on LSE, SCAN, and YACC/LEX, but open for user development.

Implementation languages for tools like these is not important, but the user should be helped to easily develop his tools, integrated with LSE or as preprocessor to a compiler, also integrated with VAX document. Examples: For concept-building: a tree/list editor; for specifications: a decision table to language translator, a state-event table/ translation diagram aid; for integration of documentation, design, and code; for design: a module/hierarchy/cross-ref editor

LSE   **Name: Shirley Bockstahler-Brandt**   **ID 97**
Batch submission from LSE.

I can issue a COMPILE command, but my compilers reside on a different VAX, so I need to be able to submit a batch job to compile my buffer.

LSE   **Name: Stan Schulter**   **ID 100**
Development environment in RSX for enhanced productivity, especially for F77, both on PDP and VAX.

Need access to internals of RSX FORTRAN (and other languages) to get better code at higher efficiency our of the development cycle. LSE type integration of tools would really help. We'd like common tools on PDP and VAX, so we don't have to learn two entirely different systems.

LSE   **Name: Chet Small**   **ID 98**
Encourage vendors of cross-compilers to provide interfaces with LSE and SCA. At least provide simple, user friendly tutorial. Users often don't have the time, background, knowledge to do these tasks!

LSE   **Name: Teri McNamara**   **ID 99**
Better integration of cross compilers and tools into VAXset.

My group develops embedded control software. While CMS-MMS can understand what our tools are, we would like to use a lang. sens.editor and compile, test, and debug from the edit session, but with our cross tools instead of VAX native tools. Document data and execution interfaces so we can interface with VAXset. Encourage more cross tool vendors to use interface. We want to do module tests and low level integration tests with simulator on VAX and use DTM to capture and drive the tests; PCA/SCA to get profiles of structure and

MACRO   **Name: S K Wykoff**   **ID 101**
Add functionality to MACRO-32 to declare a subprogram to be a formal procedure.

Give Macro-32 the ability to declare the number of arguments and the expected calling mechanisms, so that the linker can p rovide the same checking as with high-level languages.

MACRO   **Name: Ted Marshall**   **ID 102**
Allow MACRO-32 programs to generate exotic object file constructs.

The VMS object file format contains several special constructs, mostly for high-level languages, which constructs to generate these object constructs. I don't really care how complex these macro constructs are since I can build macros to do exactly what I want.

MACRO   **Name: LLL Working Group**   **ID 142**
Add .RADIX to Macro-32

MACRO   Name: LLL Working Group   ID 143
Add translator from Macro-11

MACRO   Name: LLL Working Group   ID 145
Formal function declarations in MACRO-32

MACRO   Name: LLL Working Group   ID 146
Typed and sized symbols (both : & =)

MACRO   Name: LLL Working Group   ID 147
Universal labels and symbols

MACRO   Name: LLL Working Group   ID 149
Allow option type and size checking.

MACRO   Name: LLL Working Group   ID 150
Allow local symbol names (non-global) longer than 32 characters.

MACRO   Name: LLL Working Group   ID 151
Allow longer ASCIC strings.

MACRO   Name: LLL Working Group   ID 152
CLI$ option to NOT upper-case.

MACRO   Name: LLL Working Group   ID 153
INCLUDE files, allowing search list and CDD.

MACRO   Name: LLL Working Group   ID 154
Have MMS allow "." as first character of module name in text and object libraries.

MANY   Name: George L. Scott   ID 14
Full labeling and carry-thru of source files and tools used to create a linked object. used to create
a linked object.

For good CM, the listings and source files used to develop an object executable must be clearly
identified. Add such carry-through identification so that a part of the executable can be listed
(possibly by a special processor) to show all sources, compiler versions, linkers, and other tools
used to shape that executable.

MMS   Name: Michael Berman   ID 103
Brief MMS log files

There is far too much verbiage in an MMS log file. I would like to see an option for a brief log file
– one that would specify which commands (action lines) were issued and why. Currently the
/Log option will specify the date/time of every object in the track. I'd like to see only the ones

that necessitated the firing of the action line.

MMS   **Name: Kevin L. Lundeen**   **ID 169**
/CREATION_DATE qualifier. Use creation date instead of modification date of files.

MMS   **Name: Kevin L. Lundeen**   **ID 170**
Much larger command strings (10000 +).

I know it's a DCL limitation, but that's no excuse! At least let me get large macros into a file somehow.

NOTES   **Name: Michael Berman**   **ID 104**
NOTES should have a callable interface.

A callable interface would allow more flexibility. Users could then write tailored user interfaces, new-note checkers, etc.

NOTES   **Name: Mark Katz**   **ID 105**
NOTES should have at least one more level of hierarchy available between conference and topic

Large conferences with hundreds of topics are cumbersome, but large numbers of small conferences can be confusing. An additional level (sub-conference) would allow a more logical breakout of discussions.

PL-I   **Name: Donovan Dean**   **ID 106**
Rdb pre-processor

PL-I   **Name: Matt Madison**   **ID 107**
Unsigned Attribute for FIXED BINARY VARIABLES

It would be nice to extend support for unsigned manipulation beyond what is offered by POSINT. An unsigned attribute for FIXED BINARYS would be a logical next step.

PL-I   **Name: Greg Gerke**   **ID 108**
Nested comments would be nice to have to comment out sections of code that have comments in them.

RTL   **Name: Don Gummow**   **ID 109**
How about a LIB$PURSE-FILE routine?

The VMS run-time library already has routines to create directories and delete files. Doing a purge function isn't hard (Find-file and Delete-file does the trick), but its the sort of thing I'd expect in the RTL.

SCA   **Name: Teri McNamara**   **ID 110**
Ability to use cross-compilers with SCA

It's difficult to justify LSE over other editors (TPU) without the extra benefits of SCA. PC users in our shop are doing better than VAX users for targeting support.

SCA   **Name: B. McIntire**   **ID 111**
A way to customize SCA the way one can customize LSE today.

Our target system is assembly language-based. We can use CMS and LSE, but we have no way to

use the other VAXSET tools like SCA, PCA, and DTM.

| SCA | Name: Mike Neale | ID 112 |

SCA should be able to generate source-file dependency information suitable for MMS description files.

Since SCA already knows what source files/modules are included by other source files, it should be able to generate a fragment MMS description file. This file should be as syntactically correct as possible, but need not be complete; manual editing to finish it would be acceptable.

| SCA | Name: Lindsay Todd | ID 113 |

Increase speed, reduce disk requirements!

This product requires far too much CPU time to be useful from a VAXstation 2000. It also uses far too much disk space for me to use it; we have many other users! I have never had the time or space to set up an SCA library for a large application. It also takes so long to incorporate changes into a library. Compressing the library often fails, leaving the library in an inconsistent state. When I have established libraries for a subset, this is a useful item. If it were usable, I would certainly use it!

| SCA | Name: Don Gummow | ID 114 |

Callable interface to SCA and DTM.

In a project management application, we wanted to set up a new development environment, includeing establishing the SCA and DTM libraries. There was no way (other than SPAWN) to:
    SCA CREATE LIBRARY
    DTM CREATE LIBRARY or any commands to modify the libraries.

| SCAN | Name: David K. Ream | ID 115 |

Look-behind or lead-in token operator.

Given: SET alpha ('a'..'z'); TOKEN code_name { '<' & alpha...: '>'}; MACRO find_code { c: code_name}'c' would be set to 'beta' given that the input stream contained "<beta>". The leading '<' would pass to the output stream automatically.

| SCAN | Name: David K. Ream | ID 116 |

Non-token answering.

A new ANSWER statement attribute specifying that the text being answered should not be tokenized, but should just be sent to the output stream. ANSWER OUTPUT_STREAM 'output';

| SCAN | Name: David K. Ream | ID 117 |

A built-in token to match the first occurrence of one of a set of strings. [picture_variable,*,*:] FIND_EARLIEST(s1,s2,s3...) This would function like FIND, but would match the first or earliest of s1, s2, s3, ... that occurs in the input stream.

| SCAN | Name: David K. Ream | ID 118 |

SIGN function

integer = SIGN(integer);Return -1, 0, or 1, depending on the sign of the integer parameter.

SCAN   **Name: David K. Ream**   **ID 119**
Maximum length of string function

integer = MAXLENGTH(varying_string)Return the maximum length of the declared variable-length string, not its current length.

SCAN   **Name: David K. Ream**   **ID 120**
Start of line SCAN special character.

S'SOL' would be useful in some situations in addition to or instead of S'EOL'.

SCAN   **Name: David K. Ream**   **ID 121**
Character or token bypass or skip.

The TOKEN attribute IGNORE finishes off the current token being built, if any. An attribute of SKIP_OVER would build that token eliminating the string as if it did not appear in the input stream at all. Thus, a token could be built from characters on either side of the token skipped over. Alternately, skipping over distinct characters would be helpful—maybe via a directive statement.

SCAN   **Name: Larry Killgallen**   **ID 122**
PCA coverage support for SCAN picture matching.

Need to know that test suite covered all possible combinations expressed in a picture.

SCAN   **Name: David K. Ream**   **ID 123**
Suppression of answered tokens during debugging.

Create separate /EVENT=ANSWERED_TEXT so that answered strings don't automatically show as tokens while debugging unless actually triggerable.

SCAN   **Name: David K. Ream**   **ID 124**
Boolean tree subscripts

Allow boolean expressions as legal subscripts to tree variable.

SCAN   **Name: Larry Killgallen**   **ID 125**
Control over whether outer procedures are global or not.

The linker rejects duplicate globals when I didn't want them global.

SCAN   **Name: Larry Killgallen**   **ID 126**
Need %INCLUDE rather than just INCLUDE as in Bliss.

Need to be able to include fragments of a statement rather than.

SCAN   **Name: David K. Ream**   **ID 127**
Extending files

OPEN statement allow 'FOR EXTEND' clause.START SCAN allow 'EXTEND FILE' as well as 'OUTPUT FILE'.

SCAN   **Name: Don Gummow**   **ID 128**
Enhance data types

To use SCAN fully, we need to reference data structures generated and defined in other languages--byte, word, longword integers, plus array support are needed.I tried to code a routine that accepted a string with embedded placeholders and an array of descriptors that pointed to caller variables that were to be substituted (as ASCII representation) for the placeholders. I couldn't declare a descriptor record type much less an array of them. Workaround: drag around the address of the array and call C to do the conversion.

SMG   **Name: LLL Working Group**   **ID 148**
Have SMG be able to signal as well as return.

SPM   **Name: Teri McNamara**   **ID 129**
Add support and emphasis for non-software portions of projects in SPM and its title and documentation.

Software in my company is part of a larger system. All aspects must be managed--mechanical, electrical, marketing, manufactur- ing. It is difficult to convince these groups that a tool called SOFTWARE Project Manager will meet their needs. We must solve our whole planning needs, not just software. Change the name so the sales job is easier.

TECO   **Name: Mark J. Hyde**   **ID 183**
Update DECUS library TECO submission.

This request is to reinforce my efforts to persuade DEC to update the PDP-11 TECO library submission (currently V36) to V40; that is, to submit V40 to the library and the SIG tape. I would also request that the new submission include all TECO sources, including the kernel and the .RNO of the manual.

TEX   **Name: Mona Smith**   **ID 130**
Concentrated group of sessions on publishing techniques.

Suggested topics including: TeX vx. VAX document--pros and cons,or bats and brickbats; Desktop Publishing News; More info on WEB--the TeX interface discussed in LT063.

TOOL   **Name: Shirley Bockstahler-Brandt**   **ID 131**
Tool integration--reverse engineer documentation.

I need to produce documentation, including graphs and pictures, from existing source code. This wish includes mil-spec (both B5/C5 and 2167).

TOOL   **Name: Jim Sroga**   **ID 132**
Backend engineering tool to help modify old poorly documented code to meet current industry standards.

I am modifying FORTRAN IV code written by a third party. This program not only uses FORTRAN IV, but also Macro-11 subroutines.Equipment: PDP-11/40, 23+, 70.Software: FORTRAN-77, FORTRAN IV, Macro-11.

TOOL   **Name: Arthur Varady**   **ID 133**
A table-based cross-compiler supported by the VAX tool set (PCA, SCA, etc.) to generate obj. code for target from VAX C.

Target system is a custom microprocessor for which we supply in-house tools currently.

TPU   **Name: David Jones**   **ID 134**
Built-in rectangular cut and paste; support fixed-length output files.

We edit 300-byte fixed length files and would like to have the output file created in the same file format as we read in.

TPU   **Name: John McMahon**   **ID 135**
DCL lexical type interface for TPU.

TPU built-in procedures currently do not have a simple way to call the $GETSYI and $GETJPI services (and other system services). Having a F$GETSYI, F$GETJPI, etc., lexical-type calling mechanism for TPU would be helpful. Especially for MAIL editors and system management editor-based tools.

TPU   **Name: Michael Welch**   **ID 136**
A next-buffer command for TPU which would cycle through all buffers/files.

TPU   **Name: Mark Kurtz**   **ID 137**
SPAN should match the longest possible qualifying string when searching in reverse.

TPU searches in reverse by moving its starting point backwards one character at a time and searching forward for a match. This means that SPAN will not match a full matching string in reverse since it will stop when it finds the FIRST character of the matching string.

TPU   **Name: Kevin L. Lundeen**   **ID 156**
User-specified QIO terminators for reads, not just Return and Control-Z.

Would be useful for command completion, etc.

TPU   **Name: Bob van Keuren**   **ID 162**
Option to have the Tab key insert spaces instead of tabs.

Tab characters do funny things in text files, and can sometimes be hard to work with. An option to have TPU (or at least EVE)insert spaces instead of tabs would be very handy. It should calculate how many spaces are needed to get up to the next tab stop, and insert them. I've implemented this, but it would be better to have a built-in SET option.

TPU   **Name: Kevin L. Lundeen**   **ID 163**
Ability to resignal an error in the ON_ERROR block, to allow selective error catching.

TPU   **Name: Kevin L. Lundeen**   **ID 175**
Some way to ask whether current position is visible.

Can I do this now?

TPU   **Name: Kevin L. Lundeen**   **ID 178**
Visible marks at end-of-line.

TPU,   **Name: Joe Pollizzi**   **ID 138**
Line mode editing for TPU; selective copy (i.e., COPY ALL "---" TO "---").

TPU,   **Name: Phil Wettenstein**   **ID 139**
Vertical window support in TPU.

Allow editing two procedures side by side.

TPU,   **Name: Sam Whidden**   **ID 140**
Create access to DCL command recall buffer.

Allow insertion of saved DCL commands at VMS level into an EVE buffer (perhaps the command buffer), so they can be edited and re-executed within the TPU session.

TPU,   **Name: Kevin L. Lundeen**   **ID 176**
Show current line number; go to line number.

Our TPU code now counts from the beginning of the buffer; very slow.

UTILI   **Name: Ladd Vagen**   **ID 141**
Command recall in ALL VAX utilities; abbreviated command recognition uniformly implemented (e.g. SH for SHOW).

VAXSE   **Name: LLL Working Group**   **ID 144**
Put VAXset on PDP-11's

WISHL   **Name: Kelvin Smith**   **ID 10**
Print questionnaire lines at six lines per inch, to mesh with typewriters and printers!

The questionnaire is at 7 1/2 lines per inch, which makes typing these a real hassle!

NETwords

The NETWORKS SIG Newsletter

Application
Presentation
Session
Transport
Network
Data Link
Physical
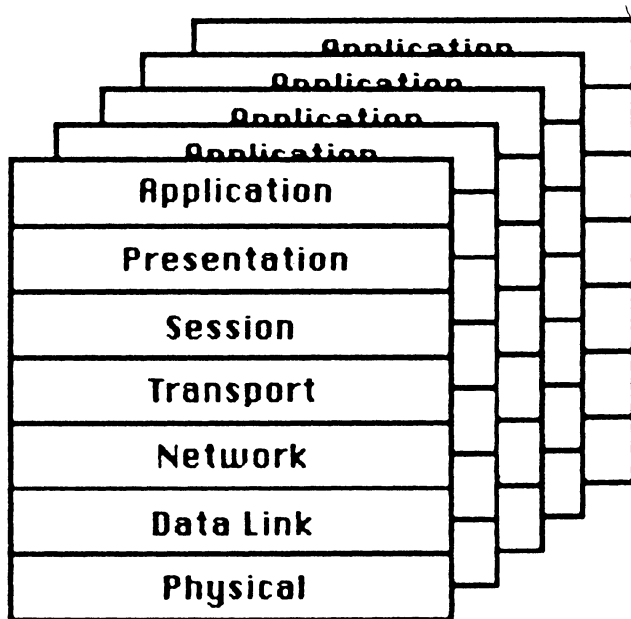
Greetings from the spiderweb!

It seems the spider is about to be resurrected as the Network SIG mascot, after a mysterious absence. Rumor has it that the hairy pet DECNET-I, the official NETsig spider, "crashed and did not reboot while going from version 2 to version 3 (i.e. during its third molt) Consigned to NLA0:..R.I.P." The rumor monger is none other than Jim Ebright, editor emeritus; gone but not forgotten...

Any comments or opinions on this mascot issue should see the appropriate authorities at the Network SIG Suite in Anaheim.

Now that the important business is out of the way, we can get down to "other business" - i.e. stuff to be on the lookout for at DECUS!

Barbara Goward from DEC CSS Marketing in Merrimac has written the following article outlining the CSS products that will be shown at DECUS. Thanks!

Stu Labovitz, the SIG Wishlist/Improvement Request coordinator, explains how to express your desires and fantasies of added functionality, on the appropriate medium. Stu'll take it from there.

And a note of thanks for Stuart Vance, the session coordinator, for taking on the messy job of juggling the network sessions into one short week. A lot of blood, sweat and tears goes into getting all the sessions accomodated within the constraints of time and space. Thanks Stuart!

See you in Anaheim!

Judi Mandl
UCONN Health Center
263 Farmington Ave.
Farmington, Ct. 06032

****

CONFIGURING REMOTE TERMINAL NETWORKS
Barbara Goward, DEC CSS Marketing

The key strength of Digital's distributed computing architecture is its flexibility to provide easy access to computer resources from anywhere in the network. For a local environment, the usual front-end communications solution is for terminal users to be connected through DECservers to the Ethernet LAN. But what options can Digital provide to customers who need to extend their network to include remote sites?

Digital's Remote Terminal Network products provide alternatives for enabling remote users to easily and efficiently gain access to distributed computing resources. These products, shown in the following diagram, enable remote connections through phone lines, X.25 networks, or microwave links.

REMOTE TERMINAL NETWORK PRODUCTS

```
Local Ethernet
      o               o               o            o   o            o
      |               |               |            |   |            |
DECserver         VAX       MUXserver 100       MIRA         METROWAVE
    200               |               |            |            |
      |             DFM       DF129 modem        /          Microwave
Scholar Plus          |               |            |            Link
    Modem            /               /          X.25            |
      |               |                           Network     METROWAVE
     /              DFM       DF129 modem         |            |
      |               |                          /            |Remote
Scholar Plus       16 T      DECmux II            |            |Ethernet
Modem             or          |          |       DFX        -o--o-----o--
      |           modems  8 T       DECmux II     |            |       |
    1 T                            |            16 T      DECserver |
                                 8 T                        500      MicroVAX
```

NTW-1
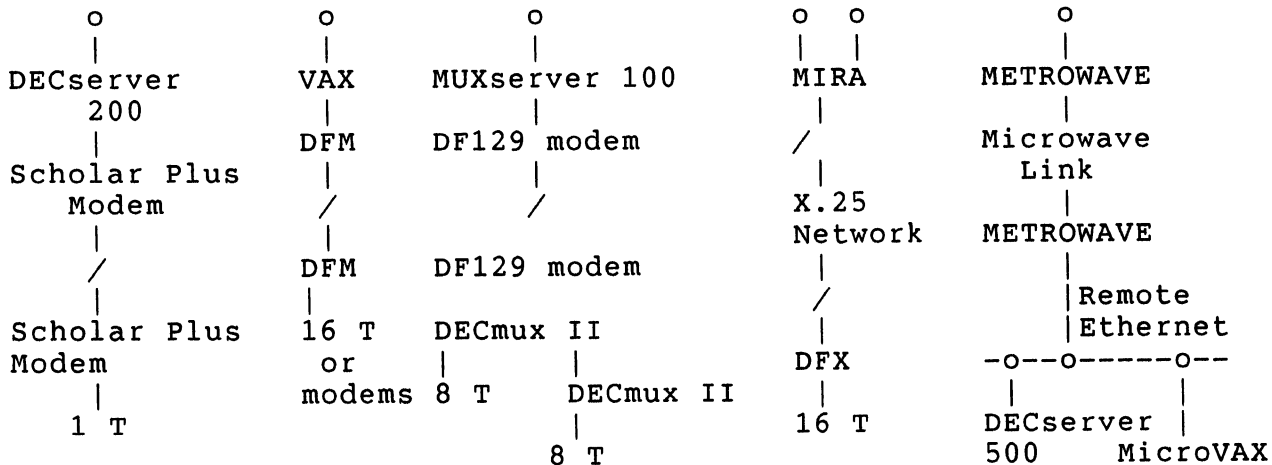
Scholar Plus Modems

The Scholar Plus modems communicate in asynchronous or synchronous mode over a dial-up or leased line at up to 2400 bits per second. They are available in compact desktop enclosures or rack-mount modules.

Scholar Plus modems add an additional layer of security to prevent unauthorized system access. Modem parameters are password protected, and one of four levels of dial-in security can be selected to screen calls. Security ranges from pass-through with a preset password, to a complete modem disconnect and dial-back to a preset phone number.

For ease-of-use, two command languages are available with the Scholar Plus modems. The industry standard Hayes "AT" command language assuresthe experienced modem user quick set-up, and the Digital Modem Command Language (DMCL) guides the novice user through set-up using a series of parameters in a menu-like format.

The Scholar Plus modems assure accurate data transfer by implementing Microcom's MNP and TYMNET's X.PC error correction protocols. This is especially important in the financial market where errors can be costly. For example, small branch banks may use this solution for updating customer files in the home office database concerning account transactions and loan information.

MUXserver 100/DECmux II Remote Terminal Server

The MUXserver 100/DECmux II Remote Terminal Server functions as a combined terminal server and statistical multiplexer to extend an Ethernet LAN through a leased phone line to include up to 2 remote sites. Each MUXserver 100 connects up to 16 users to the Ethernet, and each DECmux II connects up to 8 remote devices. Up to 2 DECmux II's can be connected to 1 MUXserver 100 either in a star or daisy-chain configuration.

It is more cost-effective to share one leased phone line with stat muxes than it is to have multiple dial-up phone lines with modems when there are multiple devices residing at the same remote site. Dial-up lines are charged by the hour whereas leased lines have a set monthly charge regardless of usage or number of devices.

In the banking industry, small branches need an easy and cost-efficient way to have teller terminals communicate to computing resources located at hub centers. The MUXserver 100/DECmux II can solve this need. Remote users connect to the Ethernet in the same way that they would if they were locally connected to a DECserver. Terminal Server Manager software enables central network management from the local site. Hardware costs and space are reduced since the MUXserver 100 takes the place of both a terminal server and a stat mux at the central site.

DFM

The DFM is a statistical multiplexer that concentrates up to 16 users from a remote site to a local host-based site through 1 leased phone line. The DFM can also be used in a daisy-chain configuration to connect up to 2 remote sites. This point-to-point solution also has data switch capabilities so that users can share local or remote resources, such as printers or CPU ports.

In the retail industry, small stores and distribution centers scattered through an area may need to communicate a long distance to a regional office to input sales transactions, inventory needs, and other administrative information. To save on phone costs, modems can be connected to a remote DFM to link outlying offices, and the DFM can then connect back to the regional office with only one phone line.

DFX PAD

The DFX PAD (Packet Assembler/Disassembler) concentrates data from up to 16 terminals and packetizes the data for transmission over an X.25 Packet Switched Data Network (PSDN). The PSDN charges are based on the amount of data transferred, rather than the call duration or distance.

The DFX PAD X.25 connection offers the flexibility of local channel switching so that users can share local or remote resources. Users can also communicate with any other site on the PSDN that is X.25 compliant.

As an account inquiry system, the DFX PAD may be the most cost-effective solution for connecting remote branches to the main site. Remote branches that do not need to be on-line continuously and do not transfer large volumes of data are potential DFX PAD users. For example, small brokerage offices that need to check on stock prices periodically may use this solution.

Some of the more common DFX PAD applications include customer records updates, database inquiries, and satellite order entry/tracking. The common factors are: (1) geographically remote sites of up to 16 terminals, (2) intermittent data transfer, (3) need for centralized record storage, and (4) do not need to be continuously on-line with host site. Small catalog stores in the retail industry exemplify this type of application.

MIRA High-availability System

MIRA is a general purpose computer system which uses a backup MicroVAX II to reduce downtime through automatic switching of communications lines. Higher computer availability is achieved through MIRA's master/standby architecture.

CIM applications which perform on-line monitoring of inventory and finished goods are ideal for MIRA since they generally require backup during system failure. Hours of downtime suffered while a non-redundant computer is being repaired causes unacceptable costs plus a great deal of anxiety since the downtime duration is not predictable. Using a master/standby system such as MIRA can reduce the downtime to minutes, which lowers the cost considerably.

There are applications in which a MicroVAX is used to monitor and control parts and materials through on-line sensors that are microprocessor controlled. If the MicroVAX is not operational, each controller continues under the control of its microprocessor until it needs instructions. This situation is very common in the factory environment and is perfect for MIRA. Typically, the controllers do not stop during the brief lapse in control. Therefore, the backup MIRA MicroVAX can continue communications with the controllers in the event of a failure of the master MIRA MicroVAX.

MIRA is also applicable to the finance industry. While many large OLTP systems require fault tolerance, there are niches where MIRA's master/standby architecture fits. One very good niche for MIRA is in the communications front ends for larger OLTP systems. MIRA's master/standby design provides a backup communications route for data coming into the OLTP system.

Message switching applications are also a good fit for MIRA. This generally means that MIRA is used in a network to make critical nodes redundant. These are typically the nodes that carry much of the communications traffic in and out of the network. In most recent cases, the communications handled by MIRA have either been DECnet or X.25 Packet Switching.

METROWAVE

The METROWAVE Bridge connects two or more Ethernet LANs having a line-of-sight distance of up to 4.5 miles, bridge to bridge, for a single hop. METROWAVE provides an Ethernet transmission speed of 10 megabits per second via a 23-gigahertz microwave link which is manufactured by a joint marketing partner, M/A-COM, Inc. METROWAVE extends a customer's access from an Ethernet LAN in one facility to an Ethernet LAN located at a remote facility. This type of link is seamless to the user and gives the appearance that one is directly connected to the local network.

METROWAVE is an ideal solution for a campus or metropolitan environment. It can be cost justified when cable cannot be installed or is too expensive. It is easily installed resulting in quick availability. Since it operates at Ethernet speed, METROWAVE provides a faster, larger bandwidth alternative than does T1 (1.544 megabits per second) technology. Leased line alternatives which typically operate at or below T1 speed require recurring monthly operating costs which METROWAVE does not. METROWAVE can also be deinstalled and easily moved to another location thereby preserving the initial investment, unlike cable which typically remains in the ground and can be susceptible to "backhoe" attenuation if it is accidentally damaged during construction activities.

The financial industry is a typical example of a METROWAVE application. A major banking institution needs to connect a remote facility with its main data center located a few city blocks away. Using the METROWAVE Bridge, this transparent connection is achieved and at full Ethernet speed. This connection accomodates host-to-host linkages as well as remote terminal users who by way of terminal servers are linked to the main data center via the METROWAVE Bridge.

FOR FURTHER INFORMATION

For further information on Digital's Remote Network Products, please consult your local Networks Specialist or call 1-800-832-6277.

Hayes is a registered trademark of Hayes Microcomputer Products, Inc. TYMNET is a registered trademark of TYMNET, Inc. Microcom is a registered trademark of Microcom, Inc. M/A-COM is a trademark of M/A-COM, Inc.

****

HOW 'BOUT SOME NETWORK FEEDBACK?

Stu Labovitz,
Networks SIG Wishlist Coordinator

Have you thought of a networks or communications product or service that you WISH Digital would offer? Let your voice be heard!! Attend the Networks SIG Wishlist Session (Thursday, 12:30 until 1:00 pm, Salon E), listen to what others have asked for, and tell us what you want, too! The Wishlist will be submitted to Digital after the Symposium ends, and this is an important channel for communicating the users' needs back to DEC. Remember, this is your chance to let DEC know what kind of network/communications hardware, software, and services you would like to see.

(Of course, if you can't attend the session because you simply can't get away from what you're doing (such as getting "stuck" at the La Brea Tar Pits, or having the monorail break down halfway over the parking lot on your way back from lunch in Disneyland) stop by the Networks SIG suite and fill out a Wishlist form.)

Don't forget to give the Networks SIG and Digital your feedback!!

****
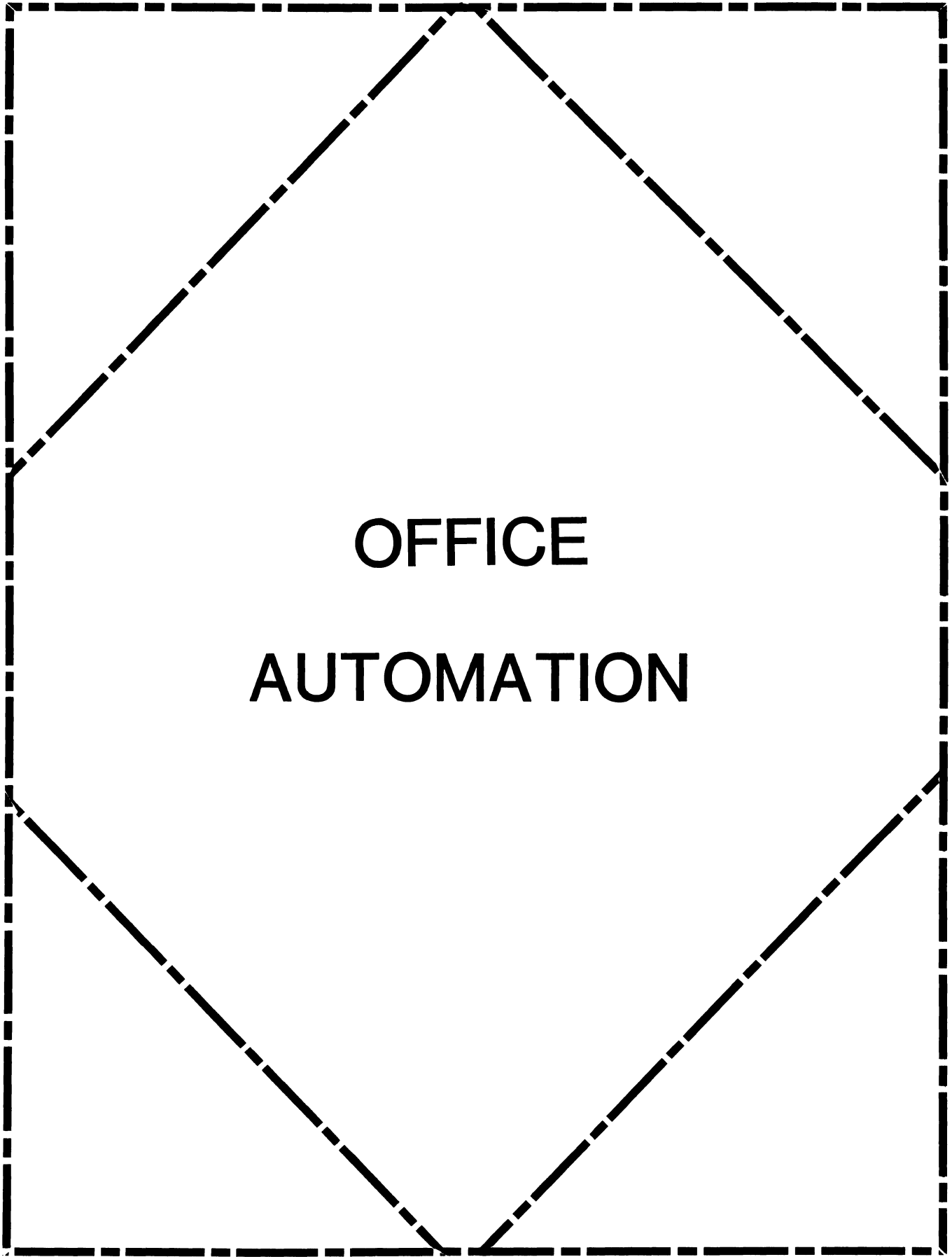
OFFICE

AUTOMATION

IN THIS ISSUE...

FROM THE EDITOR...

This month's "From the Editor" will be brief as I am preparing to leave in a few days to attend the Australian Symposium (it is currently early August). I have attended DECUS Canadian (and thoroughly enjoyed it) but this will be my first experience at a symposium on a different continent. I am looking forward to learning many new things, not just technically, but as an observer and participant from another cultural background.

You will be all reading this issue in October, just prior to your own departures for our U.S. Fall Symposium in Anaheim. For those of you who are attending, we have an outline of some "don't miss" OA activities and an overview of what the OA SIG has to offer you in the way of sessions. We've also got several terrific and useful technical articles. Two on UDP's and security issues, and one very helpful article on setting up shared laser printers so that your print job comes out on the right letterhead!

This is already longer than I had anticipated...

Just one final thought on Anaheim: At each of our U.S. Symposiums I meet attendees from other countries. Some of them are visiting the U.S. for the first time. They participate enthusiastically in what must at times seem like our unusual American rituals and events. I hope that those of you who attend Symposium have the opportunity to meet and talk with these people, they help to provide new insights and solutions to our everyday problems and concerns. Through them. DECUS becomes our own little window on the World.

Regards,

Therese LeBlanc
OA Newsletter Editor
275 London Place
Wheeling, IL 60090

OA HIGHLIGHTS FOR THE ANAHEIM SYMPOSIUM

-

The Office Automation SIG is continuing to broaden and strengthen its offerings at Symposia. All five days are FULLY scheduled, so you should plan on staying through Friday afternoon to attend all the sessions of importance to you and then join us at the OA SIG Wrapup to let us know how you feel about the SIG's activities. We look forward to welcoming all interested DECUS members to the Symposium and the OA sessions!

If this will be your first Symposium (or even if you've attended before), here is a rundown things you should know to get off to a good start and make the most of attending Office Automation sponsored sessions and activities:

SUNDAY EVENING:

- First Timers Meeting
- Welcoming Reception (Stop by our table for your own OA button)

MONDAY MORNING:

- OA SIG Roadmap: Get your roadmap for the week, meet the OA Leadership, find out about special meeting and activities during the week, get the location(s) of the Campground and OA Suite.

DURING THE WEEK

- Visit with DEC specialists or just relax and meet other people interested in OA in our Campground area.

- Join us in our suite for after hours refreshments and relaxation.

SESSION HIGHLIGHTS OF THE WEEK INCLUDE:

- The various product updates (including ALL-IN-1, WPS-PLUS, DECalc, etc.).

- Several technical sessions — both for novices and experts — on customizing and managing ALL-IN-1.

- Managerial insights into the current state of the OA art, including departmental computing, office networks and Electronic Publishing.

- Several sessions on performance measurement and management of OA systems.

- A very popular experiment from Cincinnati in which YOU get to give feedback to Digital in a 'DEC Asks The OA SIG' session.

- Several sessions on Thursday provide you with another opportunity to give feedback to Digital. This will begin with the OA Wishlist session, followed by the Question & Answer session, during which an impressive array of Digital's managers and engineers are present to respond to questions about their products and to get YOUR ideas on how they can make their products better. Come and watch the feedback process to Digital in full operation! Digital will also provide their answers to the top ten sirs provided to them by the SIG. (Remember to vote!)

- Friday will provide several security sessions on Office Automation tools including ALL-IN-1. The last session of the day will be the OA SIG Wrapup. It is *ESPECIALLY* important that you plan on attending and share your thoughts, ideas and suggestions with us. YOUR input is what makes the next symposium better.

The OA SIG Steering Committee is looking forward to your participation in the SIG's activities. We welcome your ideas and suggestions. Please join us for our best Symposium yet!

PROTECTING UDP's FROM "ILLEGAL" FUNCTIONS

Roger E. Bruner, Foreign Mission Board

-

THE PROBLEM:

In a DECUS ALL-IN-1 SECURITY session in Cincinnati, Ray Kaplan made us aware that ALL-IN-1 has some dangerously vulnerable spots. One of these is in the creation and use of User Defined Procedures (UDP's).

The ALL-IN-1 User's Reference Guide GLOSSARY defines a UDP as "A set of steps an ALL-IN-1 user defines and stores for future use; may include commands, keystrokes, and text." UDP's are especially useful for operations which are REPETITIVE or ERROR INTENSIVE. Properly controlled, they are a wonderful convenience and timesaver.

However, even the use of the word "commands" in the definition of UDP's emphasizes the potential security problem which results from letting users create and maintain their own UDP's.

Those of you who know anything about ALL-IN-1 scripts probably realize that a UDP is basically a user-created "script script". Without some type of management control, a user knowledgable in ALL-IN-1 scripting — and the number of such people is growing by leaps and bounds (would *you* think to *prevent users* from *getting to your APR's?*) — can use UDP's to do things he or she does not normally have the privilege to do.

PROVE IT FOR YOURSELF:

If you have doubts about what I am saying, try the following experiment. If you have DCL and/or COMMAND privileges in ALL-IN-1, have them taken away temporarily. Then EXIT from ALL-IN-1 and log back in. Go to the WP UD screen and Create a UDP (let's call it "BADSEC" for "BAD SECURITY"). In BADSEC, you need only two lines:

```
.fx edit "bad_sec.com"
.fx command bad_sec
```

Then file BADSEC. Enter GOLD U with BADSEC as the UDP name to run. Press RETURN. You will be

placed into the editor to create command procedure BAD_SEC.COM. Enter the following line:

$ dir/date/owner/prot oa$do:*.*

Then file BAD_SEC.COM. The next thing that will happen is that your BADSEC UDP runs BAD_SEC.COM!

While a "regular" user would need a great deal of knowledge to do any real damage (let's HOPE this is true, but can we COUNT on it?), you can already see that a knowledgable "captive user" is not very CAPTIVE at all if permitted to create UDP's at will!

SOME SOLUTIONS:

Here are some possible methods for solving this problem.

1) Do away with UDP's altogether. You will likely need security for yourself if you suggest this alternative seriously, however!

2) Disable the WPEDUDP screen so that users cannot create or edit their own UDP's. In this case, you must be prepared to use System Manager/ALL-IN-1 Manager powers to create UDP's when needed for individual users.

3) Disable the WPEDUDP screen and initiate the use of SYSTEM UDP's only (see the OA SIG TAPE). Again, you must be prepared to create the UDP's as needed, but at least each UDP is stored in a central location and is therefore easily maintained.

4) Or you may make the following changes to the WPEDUDP screen to prevent any ".F" function from being included in a user UDP [NOTE: ".FX", ".FZ", and ".FU" functions allow "do script" commands to be run from within "script scripts"].

TO PREVENT THE INCLUSION OF .F's:

Edit form WPEDUDP's Named Data and change ;;C;; and ;;E;; to look like this:

;;C;;

DO WPEDUDP

;;E;;

DO WPEDUDP

Then install the WPEDUDP.SCP script and UDP_CHECK.COM command procedure in the directory you use for new and modified procedures (use OA$LIB if you do not have a separate directory).

What was previously in the Named Data for WPEDUDP has been transferred into WPEDUDP.SCP, which branches according to the CHOICE made on the WPEDUDP screen. ALL-IN-1 stores the UDP filename in symbol #UDPFILE.

A Create or Edit of the UDP causes it to be renamed as "UDP.NEW" before editing so that the real VMS UDP file can never contain changes without those changes first being subjected to a "$ SEARCH" for ".f".

This search is done after "UDP.NEW" is filed. If no ".f" is found, the UDP is renamed to its real name. Otherwise, the user is prompted with:

.Fxx functions not permitted! Press RETURN, edit UDP, and remove them.

The user will be placed back within the editor after pressing RETURN. (An EXIT_SCREEN will function like a RETURN.) The file will be searched again for ".f" after the next effort to file the UDP.

If the user is truly "CAPTIVE" (see CONDITIONS below), he or she will have no way to get out of this loop other than to edit the UDP and remove any and all references to ".f". However, even a non-captive user who uses CTL/C to get out of the procedure (or a captive user who turns off his machine and waits for the Manager to kill his process) will find the "illegal" UDP to be unusable since it will still be named "UDP.NEW" at that point.

WPEDUDP.SCP:

```
!       WPEDUDP.SCP                        REB, 09-JUN-1988
!


.label begin
        .if oa$choice:1 eqs "c" then .goto create else .goto edit

.label create
        form udpent
        .if oa$form_terminator = 112 then .goto exit_procedure
        display Creating new UDP . . .\force
        get #t_udpfile="[.udp]udp.new"
        edit #t_udpfile
        get #curudp_file=oa$dir:"*.*".%allbutver[#t_udpfile]
        get oa$function="dump " #curudp_file
        .goto search

.label edit
        .if $curudp eqs "" then .goto no_udp_sel
        get #udpfile="[.udp]" $curudp ".udp"
        get #t_udpfile="[.udp]udp.new"
        copy #udpfile #t_udpfile
        edit #t_udpfile
        get oa$function="purge_file " #t_udpfile
        get #curudp_file=oa$dir:"*.*".%allbutver[#t_udpfile]
        get oa$function="dump " #curudp_file
        .goto search

.label search
! NOTE:  The following line is necessary because a GOLD GET of a UDP "loses"
! the #UDPFILE value in this context
        get #udpfile="[.udp]" $curudp ".udp"
        command udp_check
        .goto exit_procedure

.label no_udp_sel
        .fx display No currently selected UDP\force
        oa$fld_done
        .goto exit_procedure

.label exit_procedure
        .exit
```

UDP_CHECK.COM:

```
$!                          UDP_CHECK.COM           REB, 31-MAY-1988
$!
$! Called from WPEDUDP.SCP -- UDP is already renamed to UDP.NEW
```

```
$!
$         set message/notext/noseverity/noident/nofac
$! Get UDP filename to VMS from ALL-IN-1
$         udpfile := ""
$         write oamailbox "oa get #udpfile"
$         @dclmailbox:
$         udpfile :='result'
$!
$search:
$! Search VMS file for ".f"
$         search/nooutput [.UDP]UDP.NEW .f
$         if $status .eq. 1 .or. $status .eq. 3 then goto warn_and_edit
$         goto okay_now
$!
$warn_and_edit:
$! Let user know of illegality
$         write oamailbox -
"oa prompt '.Fxx functions not permitted!  Press RETURN, edit UDP, and remove
them.' "
$         @dclmailbox:
$! Edit the UDP file in its UDP.NEW state
$         write oamailbox "oa edit #t_udpfile "
$         @dclmailbox:
$! Prepare to search again for illegal ".f"
$         write oamailbox "oa get oa$function='dump_cache ' #t_udpfile"
$         @dclmailbox:
$         goto search
$!
$okay_now:
$! Now that UDP is okay, rename it to be usable, purge it, and get rid of any
$! possible leftovecopy '[.UDP]UDP.NEW' #udpfile "
$         @dclmailbox:
$         write oamailbox "oa get oa$function='purge_file ' #udpfile"
$         @dclmailbox:
$         delete [.UDP]UDP.NEW;*
$!
$exit_procedure:
$         set message/fac/text/sev/ident
$         exit
```

CONDITIONS:

This fix cannot be used effectively except in a totally CAPTIVE user environment. Not only must "average" users be deprived of DCL and COMMAND privileges within ALL-IN-1, but they must also be restricted in their access to the FD, PF, and PD screens as well as to the RV (Receive from VAX), SV (Send to VAX), and GOLD W to VMS functions. Any type of access to VMS by "regular" users becomes a potential threat to ALL-IN-1 security. A future article will deal with these other security issues in detail.

CONCLUSION:

ALL-IN-1 SECURITY is an issue that needs to be dealt with — by DIGITAL and by us as technically knowledgable DECUS members.

Unfortunately, it is sometimes difficult to close doors without first having to advertize just how wide open they are. I hope the information in this article will alert you to the UDP security problem and give you some specific approaches for alleviating the potential dangers at your site.

Please experiment with approach 4. Try to "break" it. If this method is "breakable", we all need to know.

BEYOND UDP SECURITY ISSUES

Roger E. Bruner, Foreign Mission Board

-

INTRODUCTION:

You may wish to review the article *PROTECTING UDP's FROM "ILLEGAL" FUNCTIONS* in this issue of the NEWSLETTER before reading this article.

BEYOND UDP SECURITY ISSUES

Whereas "ILLEGAL FUNCTIONS" deals with the security issues involved in allowing users to create and edit their own UDP's without restraint, this article focuses attention on other doors which need to be closed to make "captive" ALL-IN-1 users *more completely captive*. If these doors remain open, any effort to safeguard ALL-IN-1 from UDP's is almost worthless.

This article deals with four specific issues:

   (I) SV (Send to VAX)
   (II) GOLD Write to VMS
   (III) Printing to a VMS file
   (IV) Accessing special forms & functions

I. SEND TO VAX:

You should strongly consider the desirability of removing SV as a visible menu option from the DT (WPDXMENU) screen and leaving it as a hidden option. By editing the DXSV.SCP in OA$DO so it functions in the following manner, you may limit SV access to users who have the ALL-IN-1 DCL (PRVDCL) privilege. New lines have been bolded.

```
!         DXSV.SCP
!         This script performs the SV operation on the Document eXchange
!         sub-application.

.LABEL CHECK_AUTHORIZATION
         .IF PROFIL.PRVDCL[OA$USER] NES "Y" THEN .GOTO NOT_PERMITTED

.LABEL IF_OKAY
         FORM FCPUTVMS
         .IF OA$FORM_DISPOSE NE 2 THEN .EXIT
         DISPLAY Copying file to VMS . . .
         FORCE

!         GET #FILENAME=CAB$:DOCUMENT.FILENAME[@#CURDOC]
!         COPY #FILENAME " " #VMSFILE
         COPY "@OA$CURDOC" #VMSFILE
         .IF OA$STATUS == 1 THEN DISPLAY File copied to VMS -
          ELSE DISPLAY File copy failed
         .GOTO EXIT_PROCEDURE

.LABEL NOT_PERMITTED
```

```
              OA$MSG_PURGE
              DISPLAY You are not authorized to use SV.\FORCE

        .LABEL EXIT_PROCEDURE
              .EXIT
```

II. GOLD WRITE TO VMS:

You may also restrict the use of the **GOLD** Write to VMS to **PRVDCL** users by editing the Named Data of form **GOLDWMENU** like this:

```
    :;VMS;;
    .IF PROFIL.PRVDCL[OA$USER] EQS "Y" THEN FORM WPSVMSPUT\\
    CLOSE_PRIOR\\OA$FLD_EXIT ELSE DISPLAY You are not authorized to write to      VMS.
```

III. PRINTING TO A VMS FILE:

Unless your site has a legitimate need to use "FILE" as a document destination, you might consider simply removing it from the **PRNTTYPE.DAT** file (using **SM**, **SEL**ecting "PRINTER", and using Delete). Since printing to "FILE" was not available before version 2.2, those of you still using 2.1 need not worry about this problem YET.

However, if yours is a 2.2 site which cannot afford to disable the use of "FILE" completely, you may edit the forms **WPPARG** and **WPPARGPLN** to use the script **VAL_PRT_NAME** for validating field **DOCDES** and thus limit access to **PRVDCL** users. Change the "/VALID=" to "/RECOG=" since the validation itself is being changed to a **POST_FUNCTION**.

```
    ;;DOCDES;;

    .../POST='DO VAL_PRT_NAME'...



    !       VAL_PRT_NAME.SCP                        REB, 6-JUL-1988
    !

    .LABEL BEGIN_HERE
            .IF DOCDES EQS "FILE" THEN .GOTO VALIDATE_PRVDCL
            GET #VALID="N"
            FOR SMPRINTER WITH .SM_PRT_NAME EQS DOCDES DO -
                    GET #VALID="Y"
            .IF #VALID EQS "Y" THEN .FX OA$VAL_SET_VALID ELSE -
                    DISPLAY Enter GOLD L for valid printer names.
            .GOTO EXIT_PROCEDURE

    .LABEL VALIDATE_PRVDCL
            .IF PROFIL.PRVDCL[OA$USER] EQS "Y" THEN .GOTO DCL_OKAY
            DISPLAY You are not authorized to use 'FILE'.\FORCE
            FORM .
            .GOTO EXIT_PROCEDURE

    .LABEL DCL_OKAY
            .FX OA$VAL_SET_VALID

    .LABEL EXIT_PROCEDURE
            .EXIT
```

IV. ACCESSING SPECIAL FORMS & FUNCTIONS:

Forms like **FD** (Forms Development), **PD** (Program Development), and **PF** (Programming Functions) allow captive users to utilize functions they normally do not (and should not) have access to. You may want to restrict use of these forms to users who have ALL-IN-1 COMMAND privilege (**PRVCMD**) by editing the **DEFAULT** form in MEMRES like this:

> ;;FD;;
>
> .IF PROFIL.PRVCMD[OA$USER] EQS "Y" THEN FORM FD ELSE DISPLAY
> Invalid Choice - Re-enter

This same method may be used to regulate functions which are considered potentially dangerous:

> ;;.GOLD N;;
>
> .IF PROFIL.PRVCMD[OA$USER] EQS "Y" THEN FORM OA$VIEW_ND ELSE DISPLAY
> Invalid Choice - Re-enter

By the way, "**FORM AUTO VIEW NAMED DATA**" provides a whole-screen display of Named Data rather than the small window shown by "**FORM OA$VIEW_ND**". Therefore, you might consider making this change at the same time you implement the control described above.

CONCLUSION:

A site is not very secure from users and abusers if the ALL-IN-1 MANAGER restricts **PRVDCL** and **PRVCMD** without also governing the users' ability to *write to a VMS file* from within ALL-IN-1 or to *access highly privileged screens and functions.* **SV**, **GOLD** Write to VMS, Print to "FILE", and the use of forms like **FD** and functions like **GOLD N** are examples of this potential source of abuse. By editing some of the code related to these functions, access may be restricted to users who have ALL-IN-1 **PRVDCL** and/or **PRVCMD** privilege.

Continuing thanks to the inspiration of Ray Kaplan's ALL-IN-1 SECURITY session, to DEC's Richard Warford for an idea which inspired the UDP security procedure described in the previous article, and to George Bone for his concern about ALL-IN-1 security problems left unsolved by the UDP procedure itself.

WPS-PLUS: LETTERHEAD AND FORMS

By: Michael Error, Croation Fraternal Union

-

Since our site produces a variety of correspondence using WPS-Plus and various letterheads, we have begun to run into problems of managing the output of users. The biggest problem is that users print their letters to our Laser Printers without checking which letterhead type is in the printer. Also several programs produce letters that use different fonts or digitized signatures. Users may have changed these cartridges on that Laser printer. Using different Forms would be an easy solution, but alas, WPS-Plus doesn't give such options. We can always modify WPS-Plus but why break it to fix it?

The solution to the problem is users must now use logical Queues instead of the normal print Queue name when printing a document from WPS-Plus. By using these queue correspondence will not print unless the Logical Queue is assigned to a specific Print Queue. In this way we can control the output from WPS-Plus and determine easily when to print, where to print, and make certain that the printer is ready using the correct form and Font Cartridge. Using the following example the Logical Queues are define by that persons letterhead and the regular printer Queues are simple printer names. Queues are simple printer names. In this way the user only has to print to the Letterhead and not to any specific Queue.

Printer queue BRIGICH, stopped, mounted form DEFAULT (stock=STANDARD)

Terminal queue LASER1, on TXF7:, mounted form GRAPHIC (stock=STANDARD)

Terminal queue LASER2, on TXC7:, mounted form DEFAULT (stock=STANDARD)

Terminal queue LASER3, on TXD7:, mounted form WPSLUS (stock=STANDARD)

Terminal queue LASER4, on TXE7:, mounted form DEFAULT (stock=STANDARD)

Terminal queue LASER5, on TXB7:, mounted form DEFAULT (stock=STANDARD)

Printer queue LUKETICH, stopped, mounted form DEFAULT (stock=STANDARD)

Logical queue PLESH, assigned to LASER3

Printer queue SCHOLARSHIP, stopped, mounted form DEFAULT (stock=STANDARD)

Printer queue SPECIAL, stopped, mounted form DEFAULT (stock=STANDARD)

Since some system generated letters or forms use special fonts they are sent to the logical Queue but also use a special form. In this way we can further control the printing of letters on the system. Each of these forms that use a specific font are named for the font itself. An example is the form name "ITC_Souvenir" that is related to the LN03 Cartridge by that name.

To further eliminate the need for the Computer Department to control much of this printing operation, certain individuals who do most of the word processing are given privilege to change Queues. Since users are never able to directly access the operating system, they are limited to accessing a DCL command file that has a menu of selected commands necessary to handling these Queue changes. If any larger problems arise then we get involved. In this way we can reduce the need for us managing the Queues and also give our company greater flexibility in what it prints and how.

The following screen is what the WPS-Plus Operator accesses when a different form is needed or a Queue change necessary.

— —

Plus — Operator Menu 28-APR-1988 17:25:22.05

— —

Sn = Show/Status queue n, where n is   L = Luketich   S = Scholarship
Gn = Go/Start queue n   B = Brigich
En = End/Stop queue n   P = Plesh
                                               A = All queues allowed

Bn x = Backup x pages in queue n
Dn x = Delete entry x in queue n (show status first for
entry #)
Hn x = Hold entry x in queue n
Rn x = Release entry x in queue n

L = Load LOGO & Signatures

Fn x = Change to Form x for print queue n
SHF = Show all print FORMS available

X = Exit

Menu Choice:

PERSONAL COMPUTER

SPECIAL INTEREST GROUP

NEWSLETTER

COLUMN OF THE CHAIR
By Lynn Jarrett, PC SIG Chair

It's been a great year and we're looking forward to a great symposium in Anaheim. The program is packed full of sessions of interest to everyone with a machine on the desktop. We have Rainbow sessions, a multitude of PCSA, Workstation and Macintosh sessions. In fact, we had more sessions submitted than we could possibly schedule in our time allotted!

The SIG has been growing and we've recruited some excellent new volunteers. Of course, that doesn't mean that we aren't looking for volunteers anymore. It's like everything else, you can never have too many.

In Cincinnati, we gave a mug to volunteers who chaired sessions, worked in the Campground and who did other "chores" that had to be done. This went over very big, so we are repeating it this time around.

We are also offering public domain software, so stop in the store and check into this--Macintosh, Rainbow and IBM software. You'll also find a handy small pocketsize Atlas roadmap, sponsored by the PC SIG, for sale in the DECUS store.

Jimbo Wilson, our Symposia Coordinator, has done an outstanding job in scheduling sessions this time, and Jim Hobbs, our Campground Coordinator, has worked very hard to put together the Campground that you all have come to know and expect! Ken LeFebvre, our Comm Comm Rep, is resigning after this Symposium to manage the DECUS store and Tom Warren has stepped in to fill these shoes as well as manage to get the Session Notes out, too.

Gary Rice, our Newsletter Editor, one of our major contributors of volunteer time, is in Anaheim this week, recruiting writers for articles to the newsletters, etc. Most of our working group chairs are new in their current positions and are getting burned in in their areas of expertise. Tom Hintz, our "senior" member of the SIG, one of the founders of the group and previously our PRO Working Group Chair, now fills the position of Workstations Working Group Chair. Mike Prezbindowski is working as the Macintosh Workstations Working Group Chair and Fran Garrett is working as the PCSA Working Group Chair. Vince Perriello, our Rainbow/DECmate Working Group Chair, is dealing with our mature products.

We also have a new store rep, George Dover, who is learning our needs for the store, and is conjuring up new ideas for store items for Atlanta already! Tim Bundrick, our Seminars Rep, is working behind the scenes with the Seminars Committee in planning Pre-Symposia and regional seminars with the rest of the Seminars reps.

Our counterparts have been very helpful in assisting us in every way. Anita Uhler and Jeff Slayback have made great contributions to the SIG and we recently have had a new counterpart, John Gaucher, join us. We are still trying for an additional counterpart to aid in some of our work with Digital.

The PCSA sessions have been tremendously successful and we received more Macintosh submissions that we could handle. The Workstations sessions that are scheduled are extremely valuable. However, in the Rainbow arena, activity has died down due to the product having become mature. The VAXmate working group merged with PCSA as they go hand in hand.

Hope you will join us at Symposia and come up and introduce yourselves to the members of the SIG. We hope to see you at the NEW PC Magic on Thursday and the party to follow in the PC SIG Suite. If you can't make it to the Symposium, don't hesitate to keep in touch with us by phone or letter!

Table of Contents

PRO Section

Workstations Section

PCSA Section

PROgramming Quickie - Extending P/OS Menu capability
By Gary Rice

The P/OS menu programming environment offers a lot of capability, but one of the things that I felt that it lacked was an ability to prompt a user for additional information after a menu choice was selected but from the SAME SCREEN that was currently displayed. Now, you say "Why don't you just issue a READ statement of some kind?". Well, I ALSO wanted all of the normal menu keys to still work. Here is the result implemented as four subroutines.

```
C  GETSTF.FTN - This subroutine gets "stuff" from the user from the currently
C              displayed menu screen
C
C  ORIG VERS:      1.0
C
C  CURR VERS:      1.0
C
C  AUTHOR:         Gary Rice
C
C  CREATED:        June 25, 1988
C
C  REVISIONS:      None
C
C  INPUTS:         None
C
C OUTPUTS:         CHARACTER*5 STUFF entered by the user
C                  INTEGER*2 KEY pressed by the user to end the input (These values
C                  are documented in the PRO/Toolkit manuals. To
C                  distinguish function keys from QWERTY keys, the GETFLD
C                  routine makes the function key value negative).
C                  SOME of the values are:
```

```
C                               -9 = <Main Screen> Key
C                               -10 = <Exit> Key
C                               -15 = <Help> Key
C                               -16 = <Do> Key
C                               13 = <Return> or <Enter> key
C                               (All values are in DECIMAL radix)
C
C NOTES:           To change the length of the input field from 5 characters
C                  to some other length, change the declaration of STUFF and
C                  the value of FLDSIZ (assigned at the beginning of the routine)
C                  to match the new length declaration
C
C********************************************************
C
                   SUBROUTINE GETSTF (STUFF, KEY)
C
C
      30           FORMAT ('+',6X,'Enter your input:        ',$)
C
C
                   BYTE POS(2) BYTE CSI
C
                   CHARACTER*5 STUFF
                   CHARACTER*8 TEXT
                   CHARACTER*80 FIELD
C
                   INTEGER*2 PARM(6), STATUS(2) INTEGER*2 KEY, FLDSIZ
C
C
                   EQUIVALENCE (CSI, TEXT)              ! Make it easy to put binary into CHARACTER
C
C
C Begin
C
                   FLDSIZ = 5                           ! Set up the length of the input area
                   CALL GETADR (PARM(1), TEXT)          ! Get tha address of the TEXT variable
                   PARM(3) = 0                          ! NO carriage control on the QIO
                   PARM(5) = 1                          ! Process virtual block #1
                   TEXT(1:7) = '023;01H'                ! Set up cursor position <ESC> sequence
                   PARM(2) = 7                          ! Sequence is 7 "characters" long
                   CSI = "233                           ! 1st "character" is a <CSI>
                   CALL WTQIO ("11010,5,5,,STATUS,PARM)     ! We are now at row 23, col 1
                   WRITE (5,30)                         ! Prompt the user
                   TEXT(5:6) = '25'                     ! Set up to move to col 25
                   CALL WTQIO ("11010,5,5,,STATUS,PARM)     ! We are now at row 23, col 25
                   POS(1) = 23                          ! Fill in POSition array (row 23)
                   POS(2) = 25                          ! Fill in POSition array (col 25)
                   CALL GETFLD (POS, FLDSIZ, FIELD, KEY)    ! Get the user input
                   STUFF = FIELD(1:FLDSIZ)              ! Put it into the output variable
                   RETURN                               ! That's it
                   END
```

```
C  GETFLD.FTN - This subroutine obtains the user's input from a field
C
C  ORIG VERS:        1.0
C
C  CURR VERS:        1.0
C
C  AUTHOR:           Gary Rice
C
C  CREATED:          March 20, 1988
C
C  REVISIONS:        None
C
C  INPUTS:           BYTE by 2 POS array containing the ROW (POS(1)) and
C                         COLUMN (POS(2)) where the field is located
C                    INTEGER*2 SIZE of the field
C
C  OUTPUTS:          CHARACTER*80 FIELD - Contents of the field
C                    INTEGER*2 KEY that terminated the operation
C
C  NOTES:            This routine is lmited to 80 characters. To increase the
C                    maximum field length, replace all occurrances of "80" with
C                    the new maximum size desired
C
C***********************************************************************
C
              SUBROUTINE GETFLD (POS, SIZE, FIELD, KEY)
C
C
      10      FORMAT (I2)
C
              LOGICAL*1 FLAG
C
              BYTE POS(2), ESCSEQ(10), BYTSTR(80) BYTE PTR
C
              INTEGER*2 PARM(6), STATUS(2), KEYS(2) INTEGER*2 KEY, SIZE
C
              CHARACTER*2 ROW, COL, OFFSET
              CHARACTER*80 CONTNT, FIELD
C
C
              EQUIVALENCE (BYTSTR(1), CONTNT(1:1))  ! Allow binary in CHARCATER
C
C
C  Begin
C
              IF (SIZE. GT. 80) SIZE = 80               ! Don't let the field size be too big
              ENCODE (2,10,ROW) POS(1)                  ! Convert row from binary to characters
              ENCODE (2,10,COL) POS(2)                  ! Convert col from binary to characters
              IF (ROW(1:1) .EQ. ' ') ROW(1:1) = '0'     ! Fill in leading blank
              IF (COL(1:1) .EQ. ' ') COL(1:1) = '0'     ! Fill in leading blank
              CALL GETADR (PARM(1), ESCSEQ(1))          ! Get the address of the ESCSEQ
              PARM(2) = 7                               ! <ESC> sequence is 7 "characters"
              PARM(5) = 1                               ! process virtual block #1
              ESCSEQ(1) = "233                          ! The 1st "character" is <CSI>
              ESCSEQ(2) = ICHAR(ROW(1:1))               ! The 2nd & 3rd characters are
              ESCSEQ(3) = ICHAR(ROW(2:2))               !       ROW information
```

```
        ESCSEQ(4) = "73                        ! This is a ";"
        ESCSEQ(5) = ICHAR(COL(1:1))            ! The 5th & 6th characters are
        ESCSEQ(6) = ICHAR(COL(2:2))            !          COLUMN information
        ESCSEQ(7) = "110                       ! This is an "H"
        CALL WTQIO ("410,5,5,,STATUS,PARM)     ! Move to ROW, COL
        PTR = 0                                ! Point to the current offset of CONTNT
        CONTNT = ' '                           ! Make sure CONTNT is empty
120     CALL GETKEY (KEYS)                     ! Get a keystroke from the user
        IF (KEYS(1) .EQ. 2) THEN               ! If "2" then a "grey key" was pressed
                IF (KEYS(2) .EQ. 8) THEN       ! <Cancel> empties CONTNT
                        PTR = 0                ! Reset CONTNT offset
                        CONTNT = ' '           ! Make sure CONTNT is empty
                        FIELD = ' '            ! Ditto for the output variable
                        GOTO 130               ! Branch to re-positioning call
                END IF
                KEY = KEYS(2) * -1             ! Make "grey key" values negative
                IF (PTR .GT. 0) THEN           ! Update the output variable if needed
                        FIELD(1:PTR) = CONTNT(1:PTR)
                        FIELD(PTR+1:80) = ' '
                END IF
                RETURN
        ELSE
                IF (KEYS(2) .LT. "40) THEN     ! if <Ctrl> key then:
                        KEY = KEYS(2)          ! assign to the output variable
                        IF (PTR .GT. 0) THEN   ! Update the output buffer
                                FIELD(1:PTR) = CONTNT(1:PTR)
                                FIELD(PTR+1:80) = ' '
                        END IF
                        RETURN
                END IF
                PTR = PTR + 1                  ! Update the CONTNT offset
                IF (KEYS(2) .EQ. "177) THEN    ! if a "<X|" (Delete) key
                        PTR = PTR - 1          ! back the offset to remove the "<X|"
                        CONTNT(PTR:PTR) = ' '  ! erase the character in CONTNT
                        PTR = PTR - 1          ! back up the offset once more
                ELSE                           ! Otherwise
                        BYTSTR(PTR) = KEYS(2)  ! Add the keystroke to CONTNT
                END IF
130     ENCODE (2,10,OFFSET) POS(2)+PTR        ! Calculate the actual offset
        IF (PTR .GT. SIZE) THEN                ! Make sure the field wasn't exceeded
                CALL BELL                      ! but if it was, ring the terminal bell
                PTR = PTR - 1                  !   and back up
        ELSE
                IF (PTR .LT. 0) THEN           ! But don't back up too far
                        CALL BELL
                        PTR = 0
                ELSE
                        CALL WRITIT (ROW, COL, SIZE,       ! Show the user
        +                       CONTNT, OFFSET, FLAG)      !       what s/he typed
                END IF
        END IF
        END IF
        GOTO 120                               ! and go for more
        END
```

```
C  BELL.FTN - This subroutine "rings" the terminal "bell"
C
C  ORIG VERS:        1.0
C
C  CURR VERS:        1.0
C
C  AUTHOR:           Gary Rice
C
C  CREATED:          November 3, 1987
C
C  REVISIONS:        None
C
C  INPUTS:           None
C
C  OUTPUTS:          None
C
C  NOTES:            None
C
C***********************************************************
C
               SUBROUTINE BELL
C
C
               INTEGER*2 PARM(6)
C
               BYTE VALUE
C
C
C  Begin
C
               VALUE = "7                      ! <Ctrl> G
               CALL GETADR (PARM(1), VALUE)    ! Get the address of VALUE
               PARM(2) = 1                     ! Write 1 "character"
               PARM(3) = 0                     ! With NO carriage control
               PARM(5) = 1                     ! Process Virtual block #1
               CALL WTQIO ("11010,5,5,,,PARM)  ! DO it
               RETURN
               END


C  WRITIT.FTN - This subroutine DISPLAYS the input string at the position
C               indicated
C
C  ORIG VERS:        1.0
C
C  CURR VERS:        1.0
C
C  AUTHOR:           Gary Rice
C
C  CREATED:          April 13, 1988
C
C  REVISIONS:        None
C
C  INPUTS:           CHARACTER*2 ROW and
C                    CHARACTER*2 COL where the field is located
C                    INTEGER*2 SIZE of the field
```

```
C                      CHARACTER*80 CONTNT of the field
C                      CHARACTER*2 EOD indicating the end of valid data in CONTNT
C                               (in absolute COLUMN location)
C                      LOGICAL*1 FLAG set to true if the data should be displayed
C                               WITHOUT reverse video
C
C OUTPUTS:        None
C
C NOTES:          None
C
C*********************************************************
C
                   SUBROUTINE WRITIT (ROW, COL, SIZE, CONTNT, EOD, FLAG)
C
C
                   INTEGER*2 PARM(6), STATUS(2)
C
                   LOGICAL*1 FLAG C BYTE BUFFER(90)
C
                   INTEGER*2 SIZE
C
                   CHARACTER*2 ROW, COL, EOD
                   CHARACTER*80 CONTNT
C
C
C Begin
C
                   CALL GETADR (PARM(1), BUFFER(1))          !Get the address of BUFFER
                   PARM(2) = 7                               ! Write 7 "characters" from BUFFER
                   PARM(5) = 1                               ! Process virtual block #1
                   BUFFER(1) = "233                ! The 1st "character" in BUFFER is <CSI>
                   BUFFER(2) = ICHAR(ROW(1:1))               ! The 2nd & 3rd characters are
                   BUFFER(3) = ICHAR(ROW(2:2))               !        ROW information
                   BUFFER(4) = "73                           ! This is an ";"
                   BUFFER(5) = ICHAR(COL(1:1))               ! The 5th & 6th characters are
                   BUFFER(6) = ICHAR(COL(2:2))               !        COLUMN informtion
                   BUFFER(7) = "110                          ! This is an "H"
                   CALL WTQIO ("410,5,5,,STATUS,PARM)        ! Move there
                   IF (FLAG) GOTO 200                        ! If FLAG is set, skip "reverse video"
                   BUFFER(2) = "67                           ! This is a "7"
                   BUFFER(3) = "155                          ! This is an "m"
                   DO 100, I = 4, SIZE+3                     ! Load the buffer with the user's text
                        BUFFER(I) = ICHAR(CONTNT(I-3:I-3))
       100         CONTINUE
                   I = SIZE + 4                              ! keep track of where we are in BUFFER
                   BUFFER(I) = "233                          ! Put in another <CSI>
                   BUFFER(I+1) = "60                         ! the is a "0"
                   BUFFER(I+2) = "155                        ! This is an "m"
                   PARM(2) = I+2                             ! Process I+2 "characters"
                   GOTO 300                                  ! Branch to the QIO
       200         DO 210, I = 1, SIZE                       ! Do "Normal Video" version
                        BUFFER(I) = ICHAR(CONTNT(I:I))
       210         CONTINUE
                   PARM(2) = SIZE                            ! Process SIZE "characters"
       300         CALL WTQIO ("410,5,5,,STATUS,PARM)        ! Write the buffer
                   IF (EOD(1:2) .EQ. '00') RETURN            ! All done if we're in COLUMN 1
                   BUFFER(2) = ICHAR(ROW(1:1))               ! Set up to move to some other place
```

```
BUFFER(3) = ICHAR(ROW(2:2))
BUFFER(4) = "73
IF (EOD(1:1) .EQ. ' ') EOD(1:1) = '0'          ! Make sure to replace blanks
BUFFER(5) = ICHAR(EOD(1:1))
BUFFER(6) = ICHAR(EOD(2:2))
BUFFER(7) = "110
PARM(2) = 7
CALL WTQIO ("410,5,5,,STATUS,PARM)          ! Move to the current EOD
RETURN
END
```

The effect of the routine (as it is coded in the Quickie) will be to output a line on row 23 requesting input. Beside the input prompt will be a field (displayed in reverse video) that is 5 positions long. At that point you can enter data. If you exceed the field width, the terminal will "beep". ANY "grey key" will end input (as will any <Ctrl> key, the <Tab> key, <Enter> or <Return> key. Since these routines are not a complete program, I am not including a linker commnand file.

In The PRO Mailbag This Week
By Gary Rice

For those of you that have written or called me about the status of the PRO Public Domain Collection on TAPE, here is a copy of a memo that I received this week detailing the progress of the collection to date:

Date:	20-Aug-1988 02:09pm EST
From:	Thomas R. Hintz
	HINTZ
Dept:	PC SIG STEERING C'EE
Tel No:	904-392-5180

TO: Robert Uleski   (ULESKI)
CC: Gary Rice   (RICE)

Subject: PRO pd tape is in the mail

Bob,

Both the diskettes from DECUS and the mag tape were sent to you by 1st class mail on Friday. As mentioned previously, the tape contains 141,057 blocks. It still needs the RT DECUS diskette submissions added. I will leave that up to you since you know how to do it. Also included a printout of the directory tree structure. These can also be printed from a file called SWING.LIS that can be found in each main saveset directory.

If you or Gary have any new submissions they can be added if you have time.

SAVESET	DIR	FILES	BLOCKS
RSX	129	4704	90970
ICON	159	1199	17819
DECUS	204	1751	32268
	492	7654	141057

Thanks for the help.

Tom...

Update to the PRO Public Domain Software Collection
By Gary Rice

This month's entry is the last of the backlog of software that I have waiting to be cataloged. I have been unsuccessful at obtaining the RSX SIG tapes for either 1987 or 1988. If you have copies, please contact me.

CAT-#   Description
MISC-6   This game is similar to the arcade game, PACMAN. This program was setup for the
PRO P/OS V2.0A July, 1988 by Norman Turrill, 5057 Lake Washington Blvd. So.,
Seattle, WA 98118 206-722-5570 (evenings) It is based on the RSX version submitted to
the SIG tapes by Glenn Hoffing (RCA Government Systems Division).

1 diskette; Sources included; Objects included; Task image included
RATFIV, MACRO, FORTRAN-77

If you would like a copy of this diskette or any of my collection, here is what you have to do:

After looking through the "catalog" (available on the diskette known as "CATALOG") and selecting the items you want, send me enough diskettes to hold the software you desire. Diskette counts are listed with each catalog entry. Include a return mailer, box, carton, palette, etc. sufficiently large to hold the diskettes. Include enough postage to pay for the return trip. 1st class mail is recommended, but parcel post is ok. I will then copy the requested software for you and send it along. Give me at least a week for ANYTHING (that is, I need to have your request in my hands that long). If you request more than 5 diskettes, I will likely take longer. Specify the software you want by catalog number.

PLEASE don't ask for "specials". It took a lot of time to put THIS collection together.

Send your software requests to me:
Gary Rice
PC SIG Newsletter Editor
McDonnell Douglas
MS: K20/200
5555 Garden Grove Blvd.
Westminster, CA 92683
(714)952-6582

So: how fast IS your PRO, anyway?
By Bart Z. Lederman

Some time ago, I wrote some programs which, when run, would tell you how fast a particular CPU executed instructions. I did this for the PDP-11, and also for the PRO. I put the programs on the SIG tapes, and thought people would be interested in knowing how fast various models of CPU are, but didn't get any response.

"Basic Instructions" includes the TeST, ROtate Right, SWAp Bytes, MOVe, MOVe Byte, CoMPare, ADD, and BIt Test instructions, all of which execute at the same speed when operating on registers. "addr" is a symbolic address reference: when there are two references to "addr" in the same instruction they are to two different locations.

"Short" and "Long" is the length of the instruction sequence: a short sequence will all fit into cache and give the fastest possible CPU times. The long sequence does not fit into cache, and shows the effect of main memory speed. All times are converted to micro-seconds.

Instruction		11/70 Short	11/70 Long	11/84 Short	11/84 Long	PRO-350
Basic Instructions		0.31	0.91	0.3	0.7	2.38
TST	(R0)	0.76	1.36	0.7	1.3	4.20
CMP	(R0), (R3)	1.06	1.67	1.6	2.1	6.11
MOV	-(SP), (SP)+	1.70	2.55	2.1	2.7	6.50
MOVB	-(SP), (SP)+	2.00	2.86	2.1	2.7	6.51
MUL	R1, R0	3.19	3.81	5.7	6.5	26.64
DIV	R1, R0	1.98	2.59	9.3	10.2	8.96
BEQ	addr	0.31	0.90	2.38		
BNE	addr	0.61	1.21	2.38		
CCC,	NOP	0.61	1.21	0.8	1.25	3.23
MOV	#1, R0	0.77	1.95	0.6	1.5	4.07
ASHC	#1, R0	1.38	2.56	2.1	3.1	11.92
MOV	@#addr, R1	1.67	2.42	1.1	2.0	5.49
TST	@#addr	1.67	2.42	1.1	2.0	5.49
SWAB	@#addr	2.03	3.66	1.9	2.9	6.78
MOV	addr, R1	1.43	2.27	1.4	2.3	6.07
TST	2(R0)	1.51	2.26	1.3	2.2	6.11
TST	addr	1.48	2.27	1.3	2.3	6.10
MOV	2(R0), (R3)	1.87	3.52	2.3	3.4	8.12
JSR	PC, (R2)	3.06	3.68	9.03		
JSR	PC, @#addr	3.80	4.63	10.25		
MOV	#1, @#addr	2.04	4.08	1.9	3.2	7.34
MOV	@#addr, @#addr	2.76	4.54	2.4	3.7	8.96
MOV	4(R0), 2(R3)	2.62	4.44	3.0	4.3	9.68
ADD	#1, @#addr	3.02	4.34	2.1	3.5	8.39
ADD	@#addr, @#addr	2.81	4.84	2.7	4.0	10.10
ADD	4(R0), 2(R3)	2.68	4.75	3.2	4.5	10.73
SETF		1.22	1.83	1.6	2.0	6.14
MULF	AC2, AC0	2.57	2.68	15.4	16.3	61.35
ADDF	AC0, AC1	2.72	2.81	14.3	15.3	72.78
NEGF	AC0	1.22	1.83	4.8	5.4	15.83
ABSF	AC0	1.22	1.83	5.2	5.9	15.99
LDF	R0, AC0	1.22	1.82	11.47		
LDF	(R3), AC0	1.97	2.60	14.67		
LDCIF	R0, AC0	2.27	2.37	29.14		
LDCIF	(R3), AC0	2.27	2.59	20.47		
STCFI	AC0, R0	2.73	3.36	36.00		
STCFI	AC0, (R3)	3.36	4.17	37.59		
SETD		1.22	1.82	1.6	2.1	6.13
MULD	AC2, AC0	3.94	4.01	44.4	46.4	215.44
ADDD	AC0, AC	1 2.72	22.83	19.2	20.4	98.83
NEGD	AC0	1.22	1.82	5.9	6.5	19.32
ABSD	AC0	1.22	1.82	6.2	7.0	19.29
LDF	#1, AC0	1.81	2.88	3.4	4.4	14.29
LDF	@#addr, AC2	2.72	3.49	16.06		
STF	AC0, @#addr	4.22	5.26	4.0	5.1	11.80
CMPF	@#addr, AC0	2.89	3.49	5.7	6.9	30.40
LDCIF	#1, AC0	2.61	2.89	42.01		
LDCIF	@#addr, AC0	2.44	3.18	25.20		
STCFI	AC0, @#addr	3.41	4.89	9.9	11.1	38.24

I had only a limited amount of time on the 11/84, and ran the shortest tests: it would be desirable to run them again to obtain the same precision as for the 11/70 and PRO-350 tests. Also, the longest sequence might not have hit the maximum memory access times on the 11/84. The times obtained for the 11/70

are the same using core and MOS memory, it's not certain if memory was fully interleaved: interleaving makes main memory access faster, and so the times on the 11/70 for the long loop might improve on a machine with interleaved memory. The values for the short loop, where the entire loop is in cache, do match very well the published values in the 11/70 processor handbook

This was an early 11/84P which had the 15 MHz processor clock, and not the faster 18 MHz clock since announced. It also did not have the Floating Point Co-Processor, which is supposed to execute floating point instructions 5 to 8 times faster than the basic J-11 processor.

The times for the PRO-350 are close to but slightly slower than the times published for an 11/23, which uses the same processor. The PRO-350 CPU has to do extra work such as updating the video screen and keeping the system clock working, which may account for the difference.

Effect of Cache
PDP-11/84P

Instruction		Length of Instruction Sequence						
		100	250	500	1,000	2,500	5,000	10,000
ROR	R0	.3	.3	.3	.3	.3	.4	.8
TST	(R0)	.7	.8	.8	.8	.8	1.0	1.3
CMP	(R0), (R3)	1.6	1.6	1.6	1.6	1.6	1.8	2.1
MOV	-(SP), (SP)+	2.1	2.2	2.2	2.2	2.1	2.3	2.7
MULF	AC2, AC0	15.4	15.6	15.7	15.8	15.8	16.1	16.3
MULD	AC2, AC0	44.4	45.2	45.5	45.7	45.9	46.1	46.4

PDP-11/70, Core Memory

Instruction		Length of Instruction Sequence						
		100	250	500	1,000	2,500	5,000	10,000
ROR	R0	.3	.3	.3	.4	.8	.9	.9
TST	(R0)	.7	.8	.8	.7	1.2	1.3	1.3
CMP	(R0), (R3)	1.0	1.1	1.1	1.4	1.6	1.7	1.7
MOV	-(SP), (SP)+	1.7	1.7	1.7	2.0	2.5	2.5	2.5
MULF	AC2, AC0	2.5	2.6	2.6	2.6	2.6	2.7	2.7
MULD	AC2, AC0	3.8	3.9	4.0	4.0	3.9	3.9	3.9

Though the tests showed the effects of the larger cache on the 11/84, the transition point was shifted more twords long loops than I had anticipated. If I had the opportunity to run the tests again I would use a longer loop to insure obtaining the maximum execution times, and would run a larger total number of instructions to obtain one more significant digit of precision.

It is interesting to see what happens with certain instructions, especially floating point: the instructions which take the most time to execute suffer least from not being cached as the time needed to fetch the instruction from memory is only a small part of the total instruction execution time. The fastest instructions suffer most, as here the time needed to fetch the instruction can be as much as or longer than the time the CPU needs to actually execute the operation.

PRO Software Update
By Gary Rice, PC SIG Newsletter Editor

In past issues of the Newsletter, I have included information for software that was no longer available. With the phase down of the PRO, I will also "phase out" any entry on the list that you can't get any more. The information presented here is dated August 20, 1988. I will only delete products from the list as I (or you) confirm that they can no longer be purchased. Therefore, please don't assume that if it is still listed, it is still available.

DEC Software	List Price	Last Rev	Source of info	P/OS v3 Supported?
20/20	495	1.0.54	User	Yes
Athena/Graph	450	1.1	User	Yes
BASIC-11/RT-11	(Replaced - See BASIC-PLUS/RT-11)			
BASIC-PLUS/RT-11	UNK	3.0	DEC	N/A
CT*OS	UNK	1.0	DEC	UNK
Design Graphix/Executive	595	1.0	User	Yes
Easyentry	995	3.0B	DEC	UNK
FORTRAN IV/RT-11	495	2.8	DEC	N/A
Installation & Maintenance	UNK	3.2	DEC	Yes
LOGO	350	1.4	DEC	UNK
MJA Accounts Payable	600	5.2	User	Yes
MJA Accounts Receivable	600	5.2	User	Yes
MJA General Ledger	600	5.2	User	Yes
MJA Order Entry/Inventory	600	5.2	User	Yes
MJA Payroll & Personnel	600	5.2	User	Yes
Phoenix-PRO	1795	1.0A	DEC	UNK
P/OS ADCCP Driver	UNK	1.0	DEC	UNK
P/OS (Hard Disk)	475	3.2	User	Yes
P/OS Hard Disk (Arabic)	783	R3.1	DEC	Yes
PRO 2780/3780	53	1.2	DEC	No
PRO Application Starter Kit	399	1.0	DEC	No
PRO/BASIC	195	1.4	DEC	Yes
PRO/Comm (hard disk)	195	3.0	DEC	Yes
PRO/CPM-80	UNK	1.1	DEC	UNK
PRO/Datatrieve	495	2.0	User	Yes
PRO/DECnet	250	2.1	DEC	Yes
PRO/FORTRAN-77 Debug	(See PRO/Toolkit Symbolic Debugger)			
PRO/IVIS	UNK	3.1	DEC	Yes
PRO/Laboratory Subroutine Lib.	300	1.2	DEC	Yes
PRO/Office Workstation	UNK	2.0	DEC	Yes
PRO/PRODUCER Toolkit	300	1.6	DEC	Yes
PRO/RDT	495	1.1	DEC	Yes
PRO/Scientific Subroutine Library	300	1.3	DEC	No
PRO/SIGHT	295	1.1	User	Yes
PRO/Smart Mailer	53	1.0	User	Yes
PRO/Toolkit	520	3.2	DEC	Yes
PRO/Toolkit BASIC-PLUS-2	495	2.5	DEC	Yes
PRO/Toolkit COBOL-81	495	2.5	DEC	Yes
PRO/Toolkit DIBOL	495	1.7	DEC	Yes
PRO/Toolkit FORTRAN-77	495	5.2	DEC	Yes
PRO/Toolkit PASCAL	495	1.3	User	Yes
PRO/Toolkit Real Time Library	150	2.1	DEC	Yes
PRO/Toolkit Symbolic Debug	200	2.0	DEC	Yes
PRO/VENIX	495	2.0	DEC	N/A
PRO/Videotex	895	1.0	DEC	UNK
Professional CTS-300	995	1.0	DEC	N/A
Professional Real Time Lib/RT-11	250	1.0	DEC	N/A
PROSE PLUS	295	2.0	User	Yes
RS/1	1900	12.0	User	UNK
RSX Host Toolkit	UNK	3.0	DEC	Yes
RT-11	550	5.4D	DEC	N/A
Synergy	695	2.0	User	Yes
VAX Host Toolkit	UNK	3.0	DEC	Yes
WPS/Plus	695	1.0	DEC	Yes

3rd Party Software (Vendor)	List Price	Rev	Info Source	P/OS v3?
D-M-DRIVER for P/OS (PROTO SYSTEMS)	195	V2/V3c	Vendor	Yes
IT*OS (Intermation)	UNK	5.2	User	UNK
Online Disk Unfragmentor (By Hand)	59	2.0a	Vendor	Yes
PRO/Sentinel (By Hand)	47	1.0	Vendor	Yes
PRO/Session Logger (By Hand)	29	2.0	Vendor	Yes
PRO/Text Locator (By Hand)	43	1.1	Vendor	Yes
RDM Relational Data Manager (Interactive Technology)	995	4.1C	User	Yes
SATURN-BASE (SATURN SYSTEMS)	750	1.4	Vendor	Yes
SATURN-CALC (SATURN SYSTEMS)	500	3.0	Vendor	Yes
SATURN-GRAPH (SATURN SYSTEMS)	500	2.1	Vendor	Yes
SATURN-WP (SATURN SYSTEMS)	600	5.0	Vendor	Yes
SPSS/Pro (SPSS Inc.)	UNK	1.1	Vendor	Yes

A reader sent in the following comments:

"I use 20/20 on P/OS v3 extensively. The only problem I've run into is the print function. It will not automatically wrap the spreadsheet and print the whole thing. It will print an active sheet, but you will have to specify the ranges across & down the sheet until you get the whole thing."

If you have information about PRO software and want to share it, please contact me at:

Gary Rice
McDonnell Douglas
5555 Garden Grove Blvd.
MS: K20/200
Westminster, CA 92683
(714) 952-6582

A SPREADSHEET FOR YOUR VAXstation?
By Mark Sebern, Sebern Engineering Inc., P. O. Box 268, Cedarburg, WI 53012 (414) 375-2200

INTRODUCTION
As more users begin to view their VAXstations as personal computers, they obviously start looking for traditional PC application software to run on them. I had held off on serious evaluation of spreadsheet packages until recently, due to the prior lack of ReGIS graphics support by the workstation software (VWS). With the advent of VWS 3.3, this inexplicable omission has finally been rectified, so I was back in the market for a VAXstation spreadsheet.

This article presents the results of an initial evaluation of two spreadsheets for use on a VAXstation. The spreadsheets are 20/20 from Access Technology and C-Calc Plus from DSD Corporation. This evaluation is of course based on my opinions about what makes a good spreadsheet. Your mileage may vary.

ENVIRONMENT
Any evaluation will likely be influenced by the specific characteristics of the system in use. This evaluation was done on a color VAXstation II/GPX with dual RD54's, a TK50, and 16 Mb of memory. Printer output is to an LN03R Scriptprinter interfaced via a DHQ11 port.

ACCESS TECHNOLOGY -- 20/20
The VMS version of 20/20 is one of a family of products from Access Technology. This company has come a long way since the introduction of Supercomp-20, which was not much of a spreadsheet (to put it very mildly!). The demo copy I evaluated was version 2.31.11.

Installation
The installation of 20/20 went very smoothly, using the standard VMSINSTAL procedure. The process of choosing installation options is more nicely done than in any other VMS product I have loaded so far. It's largely a matter of esthetics, but a spreadsheet is likely to be used by non-wizards, and the care evident in the design of the installation procedure is a nice touch.

First Experiences
Once installed, 20/20 started up easily. During installation, you can choose whether alphabetic or numeric column labels are the default, and the cell designations can be changed at any time. The evaluation kit includes a special manual and set of demo procedures called "The 20/20 Revue", oriented to users of Lotus 1-2-3 and other spreadsheets.

Documentation
The 20/20 documentation is nicely done from a production standpoint. I received two manuals, one for 20/20 itself and one for the Database Connection (more on that later). They have a clean layout and incorporate many examples, figures, and model screen displays. The table of contents and indices were more than adequate.

Some Positives
EASE OF USE. Although the 20/20 user interface is not an exact 1-2-3 clone, experienced spreadsheet users should have little problem learning to work with it. The menus and help screens are laid out well, and pointing is improved over earlier Access Technology products.

SYSTEM INTERFACE. A main menu option allows entry of DCl commands without leaving the spreadsheet.

PROJECT PLANNING. A simple project schedule tool is included in 20/20, providing task and critical path information.

Some Negatives
EDITING KEYS. While Access Technology has changed many of the more annoying key definitions from the Pro-300 version, a few still remain. The one I like least is that, while editing a cell's contents using the /EDIT menu choice, you can not use the left and right arrow keys to move from character to character. Instead, function keys F18 and F19 provide these functions. If you forget and use the arrow keys, you start moving around the worksheet and entering cell designators (e.g., A45) into the cell you are editing. Editing DOES work as you would expect if you use the EDIT (PF4) key instead of (or in addition to) the /EDIT menu choice.

The Database Connection

As an extra cost option, 20/20 offers an interface to system databases. The evaluation kit I received included the Datatrieve driver, which allows read only access to Datatrieve domains based on RMS files and Rdb or DBMS databases. I tried this facility out on a few RMS and Rdb domains, and it works pretty slick! It did take me a while to realize that I had to use Datatrieve to define a relational domain for each Rdb relation I wanted to access; previously I had been using "READY <database> USING <relation>" rather than defining separate domains. (Yes, it WAS in the fine print in the 20/20 documentation.) The 20/20 Database Connection produces some undocumented error and log files which were helpful in diagnosing this problem.

One funny thing that took me a while to track down is that if any of your alphanumeric fields accidentally have a <TAB> character embedded, it really messes up the spreadsheet display.

Importing Pro-300 Spreadsheets

Since I have been running 20/20 on my Pro-350 (version 1.0.54) for some time, I was curious to find out how easily I could move those worksheets to 20/20 on the VAXstation. This requires doing a "Store Export Model" on the Pro, to create an ASCII file containing the 20/20 commands necessary to duplicate the data and formulas. This file is then transferred to the VAXstation and read into 20/20 there using a "Store Import Model" command. This worked well except for one problem. The ASCII model format apparently uses the "#" character to terminate a command, and the load process failed whenever it encountered this character in a label.

To see if this problem existed only when the Pro 20/20 exported a model, I tried the same thing on the VAXstation. This time, 20/20 warned me about one of the occurrences, as it was writing the ASCII file. In spite of the warning, I was able to read that version of the model back into the VAXstation 20/20 with no problems. Pro 20/20 crashed when trying to import this new version of the model file, however. I did notice one other anomaly. A range in Pro 20/20 of the form [6,$57..64] got translated to G$57..G64, rather than the correct G$57..G$64. Of course this doesn't matter unless you copy the formula. Just a word to the wise.

DSD CORPORATION -- C-CALC PLUS

I had looked at C-Calc for the Pro-300 a few years back, but at that time it did not support natural order recalculation, so I didn't seriously consider it. That limitation has now been removed, and a number of other improvements added. The demo copy I evaluated was version V1.5-1.

Installation

DSD uses the standard VMSINSTAL procedure, and C-Calc Plus installed smoothly. C-Calc Plus requires the one-time entry of an authorization number which is obtained by calling the vendor.

First Experiences

C-Calc Plus started up without difficulty. It took me a while to figure out the two available command modes. The "power" mode uses one or two letter abbreviations, while the "display" mode uses a limited form of menu which looks a little like 1-2-3. However, it does not display the additional description line as you scroll through the options, so you have to remember what command is under what heading. Formula entry requires a "=" prefix, and you have to be a little careful with function names. For example, "MAX(" (no space) is OK, but "MAX (" (embedded space) is not. By the way, the error message in this case is "Unknown token", which could have been a little more informative. C-Calc Plus has an annoying habit of resizing the terminal window twice each time you start it up. It may be checking to see if it can do it, as on a VTxxx terminal.

Documentation

The C-Calc Plus documentation is neatly done. I received two three-ring binders, containing a reference manual and a training manual. The reference manual seems to be complete, but lacks in organization and clarity. The table of contents and index are very limited. The training manual is considerably better, and contains numerous training exercises, examples, and screen displays.

Some Positives

USER DEFINED FUNCTIONS. C-Calc Plus allows the definition of formulas which can then be referenced by the cell designation where they reside or by a label attached to that cell. This is in addition to a more standard macro facility. While the formula is limited to a single cell per function, I believe this could be a very useful feature.

ENCRYPTION. C-Calc Plus permits encryption (as well as password protection) for individual worksheets.

Some Negatives

COMMAND MODES. As mentioned above, C-Calc Plus has two command modes. The display mode, intended to emulate 1-2-3, is very limited. The power mode seemed cryptic to me, and I had a hard time finding the right command, even with the on-line HELP command list displayed right in front of me.

TWO TIER STRUCTURE. C-Calc Plus operates at two levels, main menu or editor. The editor level is the familiar spreadsheet display, while the main menu is used for utility functions. In itself this is not that bad, but some common functions (such as saving your work) can only be done at the main menu level. Granted, it doesn't take a lot of time to exit and re-enter (because re-entry defaults to the last-edited worksheet), but it is annoying.

FILE STRUCTURE. C-Calc Plus maintains its own index of spreadsheets, and the actual worksheet files have names like CCP001.WS. This means you can't tell from DCL level which one is which.

COMPARISONS

Although I did not do extensive benchmark testing between the two spreadsheet packages, a few comments are probably in order.

FAMILIARITY. Like it or not, Lotus 1-2-3 has become the de facto standard for spreadsheet users. Although radically different user interfaces may have significant benefits, they may be hard to get used to. Both 20/20 and C-Calc Plus offer at least the option of a "Lotus like" command interface. Of the two, 20/20's style is much closer, while C-Calc Plus offers a very limited imitation. C-Calc Plus's two-tier command menu, which requires leaving the spreadsheet display to save your work, seemed rather awkward.

SPEED. Preliminary measurements indicated very little difference in speed. C-Calc Plus may have been slightly faster on repeated recalculations, but was slower the first time a calculation was done after loading a new worksheet. This seemed to be due to disk activity.

VAXSTATION SUPPORT. Both spreadsheets can do graphics in a VT240 window under VWS 3.3. C-Calc Plus also has a driver to work directly on the VAXstation screen, but I couldn't get it to work. When I tried to display a graph in this mode, the terminal window was resized to be much smaller and then C-Calc Plus appeared to go into a tight loop. I had to kill the process. A call to DSD's support line suggested that this might be due to an incompatibility with the current version of VWS or with the QDSS hardware, but I have no confirmation of the problem at this writing (only a few days after the problem was reported).

I did encounter a problem with filled areas in pie charts under 20/20; the filled areas came out white instead of the proper colors. I haven't been able to find out whether this is a problem with 20/20 or with VWS. With VWS 3.3, there are a lot of system logicals to set up for ReGIS, and I'm not sure I have them all right. Rectangular filled areas (like bar charts) work fine. C-Calc Plus uses cross hatching rather than fill for pie charts, and they worked OK.

BUILT-IN FUNCTIONS. Perhaps because both spreadsheets allow importing of 1-2-3 worksheets, there is a lot of similarity in the available functions. The functions provided by 20/20 are pretty standard, while C-Calc Plus offers some unusual ones that can iteratively execute a formula over a

number of cells, read and write files, execute DCL commands, etc. All in all, the extensions provided by C-Calc Plus didn't seem that useful to me, though they might be important for some applications.

MACROS. Both 20/20 and C-Calc Plus allow user defined procedures to be stored in an external file and then invoked easily. In addition, 20/20 allows the definition of worksheet-resident macros, like 1-2-3.

DATA CONVERSION. Both 20/20 and C-Calc Plus permit importing and exporting of data, and can exchange data in DIF, 1-2-3, and ASCII formats. In addition, 20/20 supports PC EXCEL, MASS-11, WordMarc, WordPerfect, and WPS-Plus/DECdx.

HELP. Both packages support on-line help. C-Calc Plus appears to use the VMS HELP facility, and the information is pretty complete. The 20/20 help facility is a menu-based branching design, and I found it a little easier to browse around in.

STYLE. After using the two packages for a while, I became aware of two rather distinct styles. The entire 20/20 package has the finished, end user style that many people expect from a spreadsheet. C-Calc Plus, on the other hand, has a "hacker/programmer" feel to it. Now don't get me wrong. I always considered myself to be a hacker before that term took on its current connotations. C-Calc Plus has a lot of arcane capabilities that someone who really wanted to get into it could probably accomplish some amazing things with. Still, that's not what I look for in a spreadsheet; if I need to write a program, I don't usually want to do it in spreadsheet macros. If you feel that customizing a complete Symphony database wasn't challenging enough, look into C-Calc Plus functions and macros.

OVERALL EVALUATION
Although I would really rather have a spreadsheet designed to support the VAXstation's capabilities rather than one which works only in a terminal window, both of these products offer significant functionality on a VAXstation. I guess I am now inclined to go with a package of this type rather than waiting for all the great stuff we are promised with DECwindows (Real Soon Now). When purchased for use on a VAXstation, even the pricing of these packages is getting to be acceptable, which is a major accomplishment in itself. So, if you've been putting off consideration of a spreadsheet for your VAXstation, take a look around. You might find one you like!

PCSA Wish List Presented To Digital Equipment Corporation
By Fran Garrett, PCSA Working Group Chair

The PCSA Wish List that follows was submitted to Digital in August, 1988. The top ten items are being addressed in Anaheim this month. These wishes were gathered from users at the Cincinnati Symposium and by phone from a multitude of PCSA users within the DECUS community. Look for answers from DEC in an upcoming issue, along with an updated list for each of you to give input to. -- Fran Garrett

1. Transparent access to mail from DOS, including notification
2. VMS read/write access to LAD data
3. Stalled networked printers trigger connection failure
4. Need multiple writer access to LADs
5. More granularity in mount/dismount privileges
6. Add a parameter so that forms could be used from PCs
7. LAT driver removable from memory
8. Provide resource accounting for PC users
9. Change server mounted printer form from workstations
10. Encourage PC software vendors to support PCSA as they do other LANs
11. We want DEC to provide PCSA tech tips to be published in the PC SIG newsletter. DSIN-submitted SPR's should also be included in the newsletter, such as VMS does with DISPATCH.
12. PC to PC local printer support

13. Provide support for Macintosh on PCSA
14. Open LAST protocol for the third party vendors to hook into so that protocol analyzers can be used.
15. Better documentation on server and PC side
16. Specifying a server during login should be optional-- i.e. personal directory is on a LAD
17. Version 2.0 of Windows support along with support of 386
18. Support for VGA
19. Driver to use all 3 mouse buttons
20. VT241 emulator
21. Support for ReGIS on VGA/EGA
22. Please provide VT240 emulator to support PC's andclones with EGA and LK250 keyboards under Windows and SETHOST. Also provide Tektronix emulation.
23. Have PCSA work with expanded or extended memory
24. Have VAX realize that there was a login failure when someone types the wrong password through NET USE.
25. Correct problem with autorepeat of keys in All-In-One
26. Mouse support for VT240 emulator for VAX applications
27. Support TCP/IP protocol
28. Provide transparent secondary boot node access
29. Encourage third party e-net card vendors to supply drivers. DEC should publish driver source code so users can write their own.
30. Provide a comprehensive tuning guide
31. Advanced warning of compatibility issues with new releases
32. Bring Atlanta people back to DECUS

RSTS/E

October, 1988
Table of Contents

FROM THE CHAIR
Charles W. Mustain - RSTS SIG Chairman

Why upgrade? This is a question I am often asked by users of pre-9.0 versions of RSTS. Since I am the SIG chair, these optimists figure I ought to have an answer.

Unfortunately, this is a member of that class of questions which have no single answer. All possible answers I have found begin with a constellation of IFs. Of course, when wearing my official SIG chair hat, the answer comes readily; "Do it because DEC only supports the current version and your support contract funds the continued development of RSTS and its continued support in the field."

As an active user of RSTS (5 systems), I am well aware that this is a less than satisfactory answer for most. Rather than try for a definitive answer, let me try to characterize a class of users who should be thinking of moving to the current version.

These users:

1. Want and need Software Support from DEC.

2. Wish to take advantage of the lower maintenance cost or features of newer hardware, i.e., faster disks, newer processor, high-density tapes, etc.

3. Will be using a RSTS system in a shop that also includes VMS (VAX). Reasons are similarity of user interface, easier file transfer between systems.

4. Wish to have a RSTS system that can play on the network team with VMS, RSX, Ethernet, terminal servers, etc.

5. Are doing active software development with RSTS and BASIC Plus 2 and wish to take advantage of the larger program space available with I&D space machines (11/45, 11/70, 11/44, 11/84, 11/83). Also, wishes to take advantage of the advanced syntax and vocabulary of latest versions of BP2 and COBOL 81. These newer compilers also feature vastly improved upward compatibility to their VMS counterparts.

6. Do not have a full-time system manager and want a RSTS system that is simpler to install and update.

While this list is not exhaustive, it can at least serve as a rough guide to decision making. A yes answer to any of the above is a strong indication you should be thinking update. No doubt readers will think of other items for the list.

NEXT MONTH: Some reasons you may not wish to update.

FROM THE EDITOR
Terry Kennedy - RSTS SIG Newsletter Editor

You may have been wondering about the prolonged absence of the RSTS section from this Newsletter. Well, it wasn't because we were mad about the poor print in the May '88 issue! Actually, I've changed jobs within the organization I work for, and I haven't had time to prepare the Newsletter.

I looked at my calendar, and this is my last chance to get an issue to you before the Anaheim Symposium, so rather than being mobbed by hundreds of irate RSTS users there, I figured I'd better get an issue out. [Actually, I'd do almost *any-thing* to be mobbed by RSTS users at a Symposium, but that's another issue...]

Anyway, we should be back on track for monthly publication as of this month. You will probably have noticed that we are now printed in 'one-up' format, which should make the section easier to read. Additionally, the text is now being prepared on a laser printer. One side effect of this is that all listings will be grouped together in the back of our section, as they are being prepared on a different system.

As always, the SIG is looking for articles to print. To submit an article, you may use the following methods:

```
Via US Mail:                    Via UPS, FedEx, etc.:

Terence M. Kennedy             Terence M. Kennedy
St. Peter's College            St. Peter's College
Academic Computer Center       Academic Computer Center
2641 Kennedy Blvd.             121 Glenwood Avenue
Jersey City, N.J. 07306        Jersey City, N.J. 07306
(201) 435-0252                 (201) 435-0252
```

You may electronically submit material by calling the RSTS SIG bulletin board system at (201) 915-9361, or you may reach me as user KENNEDY on both DCS and DECUServe, if you have access to either of those systems.

You may submit material on RX50's or RX33's (in RSTS or RT11 format), on 800, 1600, 3200, or 6250 BPI 9-track tape (in DOS, ANSI, BRU, RSTS BACKUP or VMS BACKUP format), or on PC-DOS floppies (5 1/4 or 3 1/2 inch format). If you are really desperate, I can also accept RSTS or RT11 format RL02 and RK07 disks. You may also submit hardcopy documents, but these will take longer to get into print.

Anaheim Symposium Session List

```
Session    Day      Time       Title
 RS006     Mon    12:00 PM     RSTS Roadmap
```

```
RS009      Mon     12:45 PM     RSTS Announcements
RS014      Mon      1:30 PM     RSTS V9.6 Tech Changes
RS017      Mon      6:00 PM     RSTS Performance Comparison
RS001      Mon      7:00 PM     RSTS 80th Birthday Preview
RS022      Mon      8:00 PM     RSTS/E Performance Tuning

RS011      Tue      9:00 AM     Intro to DECnet/E (why me?)
RS010      Tue     10:00 AM     Appl. Programming on DECnet/E
RS003      Tue     11:00 AM     RSTS Tech Tips I (SPR)
RS015      Tue      3:00 PM     What is LAT/E?
RS013      Tue      4:00 PM     Modems on RSTS
RS016      Tue      5:00 PM     RSTS Term Service Internals

RS008      Wed      1:00 PM     RSTS SIG Wish List I
RS019      Wed      2:00 PM     Using RSTS Batch Services
RS018      Wed      3:00 PM     Using RSTS Backup
RS007      Wed      4:30 PM     RSTS SIG Tape/Library
RS023      Wed      5:00 PM     Ask the RSTS Experts

RS012      Thu      3:30 PM     RSTS/VMS Migration - Sys. Mgt.
RS021      Thu      5:00 PM     Alternatives to VAX Migration
RS020      Thu      6:00 PM     PDP-11 to VAX: Hardware Issues

RS004      Fri     11:30 AM     RSTS Tech Tips II (Bits & Bytes)
RS002      Fri     12:30 PM     RSTS SIG Wish List II
RS005      Fri      1:30 PM     RSTS SIG Wrap-up
```

Related Sessions of Interest:

```
Session    Day        Time      Title
LT075      Mon     11:00 AM     PDP-11 Language Status/Futures
RX069      Mon      3:00 PM     PDP Futures - User's View
RX068      Mon      4:30 PM     DEC Asks PDP-11 Users
HM049      Wed     12:00 PM     PDP-11 Upgrade and Migration
LT074      Thu      9:00 AM     PDP-11/VAX Compatibility Tools
LT081      Thu      9:30 AM     PDP-11 to VAX Basic Migration
LT023      Thu     10:15 AM     RSTS/VMS Migration with Basic
LT068      Thu     11:00 AM     RSTS to VAX Conversion
LT071      Thu      1:30 PM     Basic-Plus-2 Pitfalls and Hints
LT076      Thu      2:30 PM     PDP-11 Langs & Layered Prods Q&A
```

Useful Dibol Subroutines
Paul Flaherty

The following DIBOL subroutine may prove useful to other RSTS/DIBOL users. It allows a running DIBOL program to submit a batch job to PBS without having to do something like chain to a .COM file. It is a plain, vanilla submit, with no parameters, etc., but it is useful in many applications, and perhaps other newsletter readers might want to add some bells and whistles.

```
SUBROUTINE SUBMT                        ;Author:        Paul F. Flaherty, Jr.
                                        ;               DP Manager
                                        ;               DANIELS AND CRONIN
                                        ;               Attorneys at Law
                                        ;               Three Center Plaza
                                        ;               Boston, MA   02108-2003
                                        ;               (617) 227-5570

                                        ;subroutine which allows a Dibol program
                                        ;running under RSTS/E Version 9.x to submit
                                        ;a batch job to run under PBS.  It is a plain,
                                        ;vanilla routine - the equivalent of a DCL
                                        ;"SUBMIT" command with no qualifiers - but is
                                        ;nevertheless useful in many applications.

                                        ;N.B.   This subroutines calls other routines
                                        ;       from LB:UNSUPP.OLB, Digital's
                                        ;       UNSUPPORTED Dibol subroutines library.
                                        ;       ===========


          SUBMIT  ,A                    ;command file name passed from calling routine
          ERR     ,D                    ;meaningless incoming; outgoing zero if success
                                        ;else error code

RECORD FIRQB                            ;the following are for xcall exemt,
                                        ;ptfrq, gtfrq
          FRQ     ,2D3
          SYS     ,30D3


RECORD
          DUMMY   ,D3
          UUO     ,D6,104066            ;looks decimal, interpreted as octal


PROC

          CLEAR ERR, FIRQB, DUMMY       ;initializw some variables
          ON ERROR ERROR                ;set up error trap
          XCALL FSS (SUBMIT,ERR)        ;file string scan to load firqb
          OFF ERROR
          IF (ERR) RETURN               ;let main routine handle error
          XCALL GTFRQ (FIRQB,FRQ)       ;get the firqb
          SYS(1) = 6                    ;sys call to FIP
          SYS(2) = -28                  ;sub call (UU.SPL)
          SYS(3) = 0                    ;according to Programming Manual
          SYS(4) = 0                    ;ditto
          SYS(13) = 66                  ;"B"
          SYS(14) = 65                  ;"A"
          FOR DUMMY FROM 15 THRU 30
                  CLEAR SYS(DUMMY)      ;initialize the rest
          XCALL PTFRQ (FIRQB,FRQ)       ;load the FIRQB
          XCALL EXEMT (ERR,UUO)         ;and do it (if error ERR is set)
          RETURN                        ;return
```

```
ERROR,                                  ;general error trap
        OFF ERROR
        XCALL ERROR (ERR,DUMMY)         ;get the error code
        RETURN                          ;and return it

    END
```

Here are a couple more routines that RSTS/Dibol sites may find useful. The first is a MACRO routine that allows a Dibol program to drop/regain temporary privileges if the privilege bit is set initially. The second is a Dibol routine which allows a Dibol program to check whether or not the job under which it is running has a particular privilege enabled. The code follows.

```
;+
;
; File TPRIV.MAC
;
;-


        .include        /SY:[1,2]COMMON.MAC/
        .title          TPRIV
        .ident          /1.0.0/
        .psect          $TPRIV

;+
;
;       Author: Paul F. Flaherty, Jr.
;               DANIELS & CRONIN
;               Three Center Plaza
;               Penthouse Mezzanine
;               Boston, Massachusetts   02108   USA
;               (617) 227-5570
;
;       Date:   August 24, 1988.
;
;-


;+
;
; Program description
; ======= ===========
;
; This is a MACRO-11 subroutine written in conformance with the Digital
; Equipment Corporation document entitled "Writing MACRO Subroutines for
; RSTS/E DIBOL V5.1."
;
; The calls to this routine are made by calling either DPRIV to drop
; temporary privileges, or RPRIV, to restore temporary privileges dropped
; by DPRIV.  TPRIV itself is not called.
;
;-
```

```
;+
;
; Revision History
; ======== =======
;
; When          Who          What
; ----------------------------------------------------------------------
; 24-Jul-1988   PFFJr        Initial source
;
;-


        .MACRO  DOXRB,  ?A              ;clear the xrb and then set jfsys
        mov     #xrb,   r3              ;point to the xrb
A:      clr     (r3)+                   ;clear a word
        cmp     r3,     #xrb+14         ;done yet?
        blos    A                       ;nope
        mov     #jfsys, xrb+xrlen       ;pick the bit
        .ENDM   DOXRB


RPRIV:: DOXRB                           ;set up the XRB
        .SET                            ;regain privs
        rts     pc                      ;return to DIBOL


DPRIV:: DOXRB                           ;set up the XRB
        .CLEAR                          ;drop privs
        rts     pc                      ;return to DIBOL


        .end                            ;that's it



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



SUBROUTINE PCHEK                ;Author:        Paul F. Flaherty, Jr.
                                ;               DP Manager
                                ;               DANIELS AND CRONIN
                                ;               Attorneys at Law
                                ;               Three Center Plaza
                                ;               Boston, MA   02108-2003
                                ;               (617) 227-5570

                                ;subroutine which allows a Dibol program
                                ;running under RSTS/E Version 9.x to check
                                ;whether or not the job under which it
                                ;is running has a particular privilege.

                                ;N.B.   This subroutines calls other routines
                                ;       from LB:UNSUPP.OLB, Digital's
                                ;       UNSUPPORTED Dibol subroutines library.
                                ;       ===========


        PNAME   ,A              ;privilege name
```

```
            STATUS  ,D                    ;meaningless incoming; outgoing 0 = has priv
                                          ;1 = no priv


RECORD FIRQB                              ;the following are for xcall exemt,
                                          ;ptfrq, gtfrq
            FRQ     ,2D3
            SYS     ,30D3


RECORD
            X       ,D3
            Y       ,D3
            ERR     ,D3
            UUO     ,D6,104066    ;looks decimal, interpreted as octal


PROC

            IF (PNAME.EQ.'        ') GOTO ERROR ;ignore blanks
            UPCASE PNAME                  ;make upper case
            CLEAR ERR, FIRQB, Y, X        ;initialize some variables
            DO INCR X UNTIL (X.GT.6.OR.PNAME(X,X).EQ.' ') ;find first blank
            X = X - 1
            SYS(1) = 6                    ;sys call to FIP
            SYS(2) = 32                   ;sub call (UU.CHK)
            SYS(3) = 1                    ;convert priv name to mask code
            FOR Y FROM 1 THRU X           ;convert string to decimal
                    XCALL DECML (PNAME(Y,Y),SYS(Y+6))
            XCALL PTFRQ (FIRQB,FRQ)       ;load the FIRQB
            ON ERROR ERROR                ;only poss error is rsts error 5
            XCALL EXEMT (ERR,UUO)         ;and do
            XCALL GTFRQ (FIRQB,FRQ)       ;get the result
            OFF ERROR
            STATUS = SYS(3)               ;set up the return value
            RETURN                        ;return

ERROR,
            STATUS = 1                    ;if there's an error tell caller nopriv
            RETURN                        ;and return

END
```

[Ed. Note: The above routines are available on the Newsletter System as files [49,1]SUBMIT.DBL, [49,1]TPRIV.MAC, and [49,1]PCHEK.DBL, respectively. See Page RST-3 for information on accessing the RSTS SIG Newsletter System.]

LINK/RSX/DEBUG Explained
Brett Bump

This memo is a comment about the letter posted in the February 88 issue of the RSTS SIG newsletter. Mr. Lutz asked whether there is a elegant solution to the DCL LINK command with the debug switch:

$ LINK/RSX11/DEBUG/MAP

Having been raised on DOS-11 I was forced to learn all of those cryptic commands that developed in RT11, RSX and RSTS. So I must admit that I have rarely used DCL since its addition to the RSTS O/S. I usually talk directly to the source programs (TKB).

Mr. Lutz is correct that the problem lies with the temporary command file. Since the dawn of DCL on RSTS there have been additional files that were included with DCL in order for DCL to do its thing. SHOTER was the original show terminal program. DCLUTL and PRELIN still exist. So do the .LNK files located in the LB: account for task building programs. The file LB:RSX11.LNK is a skeleton file which is used to create the TKB.CMD file. This file has not changed (only the version number was changed to protect the corporation) and no, the $LINK/RSX/DEBUG command didn't work in previous versions either.

Here is your trivial solution:

```
RUN EDT$:EDT
EDT>LB:RSX11.LNK
*FIND "UNITS
*SUBS/14/12/
*+2
*SUBS/:13:14//
*EXIT
```

If this solution is too trivial for a trip to Switzerland how about an invitation to speak at Decus?

CSPCOM – CUSP of the Month
Brett Bump

The CSPCOM program has been on the distribution tapes for many years now. This program has acquired a bad reputation for not working or having enough bugs that many users will not use it. This is too bad. In all of the years that I have used this program I have only found one bug that has cropped up (if you can call it a CSPCOM bug). A system manager usually runs into this bug after compiling the LOGIN.BAS program for V9. When the program is run the following error occurs:

```
??Memory management trap
(Register dump)
```

The error occurs because of the following system call:

```
FOO$=SYS(CHR$(6%)+CHR$(11%))!DEASSIGN LOGICAL NAMES AND DEVICES
```

The fallacy of this bug is that the call in itself is not used. The system call to deassign all logical names and devices is a .ULOG call. Basic-Plus achieves this

call by converting the .UUO call into the proper format for the .ULOG call. The format for the .ULOG call must have the XRB clear before executing.

Since Basic-Plus programs are interpreted and CSPCOM programs are compiled, the bug in question is not contained within the compiler but within the object library for the compiler. The thread that is called from CSPCOM is called SYS$. This symbol is located within $IESYS in the CSPCOM.OLB file. The problem in the code is one simple programmer mistake and it has been there since 1979. Instead of using the

```
Loop:
        Clr     -(r0)           ;r0 starts at 1000
        Cmp     r0,@#734        ;Does r0=(value in 734) Oops!
        Bhi     Loop            ;If not then loop
```

But as you can see one little @ got in by mistake. The outcome of this mistake is to clear low memory until the KEY word is reached effectively trashing the stack and your program. The following patch is provided so that you can get the bug out of there.

```
!
!               CSPCOM.PAT
!
!               Oct 1985
!               CSPCOM.OLB
!               Bp2 V1.6 Mini Compiler library patch
!
! BY (Brett Bump) Unemployed and available., Chadron, NE.
!
! Patch to fix fip FOO$=SYS(CHR$(6%)+CHR$(11%))
!
!               Thread is Sys$
!               Module is $Iesys
!
File to patch?
Base address? 107430
Offset address? 0
 Base      Offset    Old       New?
??????    000000    005040    ? <LF>
??????    000002    020037    ? 20027
??????    000004    000734    ? <LF>
??????    000006    101374    ? ^C
```

That is for all of you Basic programmers. Now for you macro programmers here is a little something extra to play with.

Since many of the newer system calls are not available through the Basic FIP subfunction calls much of your time may be spent hacking the code in macro. I just hate to experiment in macro. If I must write macro code I want all of my system calls debugged first. It's a very simple matter to include the EMT calls within the CSPCOM.OLB library and to do your experimenting from Basic. First we assemble the following file:

```
        .PSECT  MONITR
```

```
CALFP::  CALFIP                ; Call FIP, with FIRQB loaded
         RETURN
READ::   .READ                 ; READ
         RETURN
WRITE::  .WRITE                ; WRITE
         RETURN
CORE::   .CORE                 ; Change user memory size
         RETURN
SLEEP::  .SLEEP                ; SLEEP job for n seconds
         RETURN
PEEK::   .PEEK                 ; PEEK at memory
         RETURN
SPEC::   .SPEC                 ; Special function
         RETURN
TTAPE::  .TTAPE                ; Enter TAPE mode
         RETURN
TTECH::  .TTECH                ; Enable echo
         RETURN
TTNCH::  .TTNCH                ; Disable echo
         RETURN
TTDDT::  .TTDDT                ; DDT submode
         RETURN
TTRST::  .TTRST                ; Cancel ^O effect
         RETURN
TIME::   .TIME                 ; Get timing information
         RETURN
POSTN::  .POSTN                ; Get device's horizontal position
         RETURN
DATE::   .DATE                 ; Get current date & time
         RETURN
SET::    .SET                  ; Set keyword bit(s)
         RETURN
STAT::   .STAT                 ; Get my statistics
         RETURN
RUN::    .RUN                  ; RUN a new program
         RETURN
NAME::   .NAME                 ; Install a new program name
         RETURN
EXIT::   .EXIT                 ; Exit to default run-time system
         RETURN
RTS::    .RTS                  ; Change to a new run-time system
         RETURN
ERLOG::  .ERLOG                ; Log an error from the run-time system
         RETURN
LOGS::   .LOGS                 ; Check for logical devices
         RETURN
CLEAR::  .CLEAR                ; Clear keyword bit(s)
         RETURN
MESAG::  .MESAG                ; Message send/receive
         RETURN
CCL::    .CCL                  ; CCL checker
         RETURN
FSS::    .FSS                  ; File String Scanner
```

```
            RETURN
UUO::       .UUO            ; UUO hook
            RETURN
CHAIN::     .CHAIN          ; CHAIN to a new program
            RETURN
PLAS::      .PLAS           ; Resident library control
            RETURN
RSX::       .RSX            ; Enter RSX emulation
            RETURN
ULOG::      .ULOG           ; ASSIGN/REASSIGN/DEASSIGN device/user logical
            RETURN
XPEEK::     .XPEEK          ; Extended block-mode PEEK
            RETURN
READA::     .READA          ; Asynchronous read
            RETURN
WRITA::     .WRITA          ; Asynchronous write
            RETURN
ASTX::      .ASTX           ; Exit AST routine
            RETURN
PFB::       .PFB            ; PFB Handling Routines
            RETURN
CMDLN::     .CMDLN          ; Command line read/write
            RETURN
AST::       .AST            ; Disable/Enable AST routine execution
            RETURN
            .END

            RUN $MAC
            MAC>MONITR=COMMON,MONITR
            MAC>^Z
```

COMMON.MAC is usually in account SY:[1,2]. Next we insert this object module into the CSPCOM.OLB library:

```
            RUN $LBR
            LBR>CSPCOM/IN=MONITR
            LBR>^Z
```

CSPCOM.OLB is usually in account [1,2] for V8, UPDATE$: for V9. Then use these routines just as if you were programming in macro:

```
            10 FOO$=SYS(CHR$(6%)+CHR$(-10%)+"FOOBAR")
            20 CALL NAME
            30 INPUT "Type ^T ";FOO$
            40 END
```

Warning: If you are using VSECT for FIRQB and XRB use a version 8 or older TKB. VSECT will not work with version 9 TKB.

The 1987 Nashville/Anaheim RSTS SIG Tape
Franklin Mitchell - RSTS SIG Tape Copy Coordinator

The 1987 Nashville/Anaheim RSTS SIG tape was produced early this year. It consists of a full 2400' 1600 BPI tape of goodies, games, utilities, DCL examples, etc. (see directory below) that most any RSTS user could use. It is available through LUGs (local user groups), the DECUS Library, from the RSTS SIG Tape Copy Coordinator, and through the newly formed RSTS SIG Tape Copy Distribution Tree (see list below).

If you want a copy of the 1987 tape from me, send a NEW tape in a suitable container with return postage or send $10.00 and I'll supply the tape, container, and postage.

Send any comments, suggestions, complaints, or other input to me at the address below or via DCS to MITCHELLF.

* *

Please share your RSTS "goodies" with the RSTS SIG. Contribute to the 1988 RSTS SIG Tape Copy process! Preferred tape density/format is 1600 BPI mag tape in DOS format. V9 BACKUP format is also accepted but please, no V8 BACKUP tapes. Other media can be accepted with some delay for conversion.

Enclose a DECUS "Tape Copy Release Form" with your submission. A "Tape Copy Release Form" can be found in each Fall/Spring DECUS Symposia announcement or you can get one from me at the address below.

Submit your own work. I.E., if you have modified a DEC CUSP, submit only the changes, not the whole CUSP.

Send your submission before 1 December 1988 to:

Franklin Mitchell
RSTS SIG Tape Copy Coordinator
Erskine College
1 Washington Street
PO Box 86
Due West, SC 29639
(803) 379-2131

* *

Directory of the 1987 Nashville/Anaheim RSTS SIG Tape:

Account	Who	What
[87, 0]	RSTS SIG	This README.1ST + MT.LST directory
[87, 1]	DEC RSTS Development Team	Example .COM files, etc.

```
[87, 2]   Erskine College              Goodies
[87, 3]   Mike Mayfield                Goodies
[87, 4]   Brian Nelson, Univ of Toledo Command line editor (CLE)
[87, 5]   Brian Nelson                 New CLE
[87, 6]   The University of Toledo     Kermit 11
[87, 7]   The University of Toledo     Kermit for various PC's
[87, 8]   Ed Moran, Horace Mann School Pascal library routines for RSTS
[87, 9]   James B. Wilkinson           TYPE
[87,10]   Terry Kennedy, St Peter's    DECUS C
[87,11]   Terry Kennedy     College    Fortune Cookie program
[87,12]   Terry Kennedy                Cookie data
[87,13]   Terry Kennedy                Cookie data
[87,14]   Terry Kennedy                Back issues of the RSTS newsletters
[87,15]   Terry Kennedy                System utilities
[87,16]   Terry Kennedy                DEC MAIL utilities
[87,17]   Terry Kennedy                MS DOS Kermit 2.30
[87,18]   Terry Kennedy                Games
[87,19]   Gene Alpern                  Directory of all past
                                       RSTS SIG tapes.
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

Members of the 1988 RSTS SIG Tape Copy Tree

Christian G. Cordova-Rolon
Programmer/Analyst
Quodata Corporation
266 Pearl Street
Hartford, CT 06103
(203) 728-6777

Daniel Stoller
Director of Data Processing
Natural Country Farms, Inc.
58 West Road
PO Box 758
Rockville, CT 06066
(203) 872-8346

Chuck Evans
Director of Data Processing
Times Publishing Company
Times Square
West 12th & Sassafras Street
Erie, PA 16534
(814) 456-8531

Greg Justice
Programmer/Analyst
TDIndustries
13737 North Stemmons
Dallas, TX 75234
(214) 888-9535

Peter K. Wanger, Jr.
Bottom Line Systems
1604 McDonald Way
Burlingame, CA 94010
(415) 692-3870

Dale A. Johnson
Software Engineer
UseWare Software Development, Inc.
2235 Meyers Ave.
Escondido, CA 92025-1070
(619) 745-6006

Michael E. Tharp
Manager MIS
Masoneilan North American Operations
1040 South Vail Ave.
Montebello, CA 90640
(213) 722-8670

Robert A. Fidelman
RAF Associates
4721 Santa Rosita Ct.
Santa Rosa, CA 95405
(707) 538-2820

Paul M. Kvamme
Data Processing Manager
16680 So. Spangler Rd.
PO Box 69
Beavercreek, OR 97004
(503) 632-3113

Kenneth LeFebvre (TK50 only)
Sytek Inc.
19 Church Street
PO Box 128
Brea, OH 44017
(216) 243-1613

Eugene W. Alpern
Saber Computer Services, Inc.
Box 232
Morton Grove, IL. 60053
(312) 998-5950

Randy Chenowith
Computer Services
Canadian Union College
Box 430
College Heights
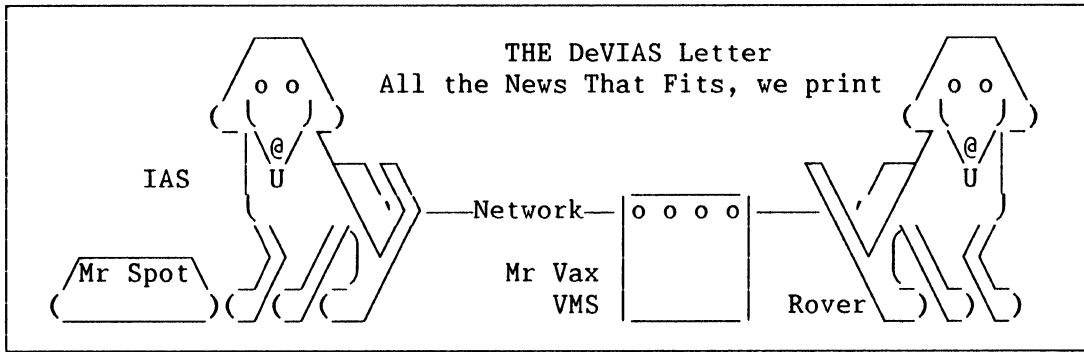Alberta TOC OZO CANADA
(403) 782-3381
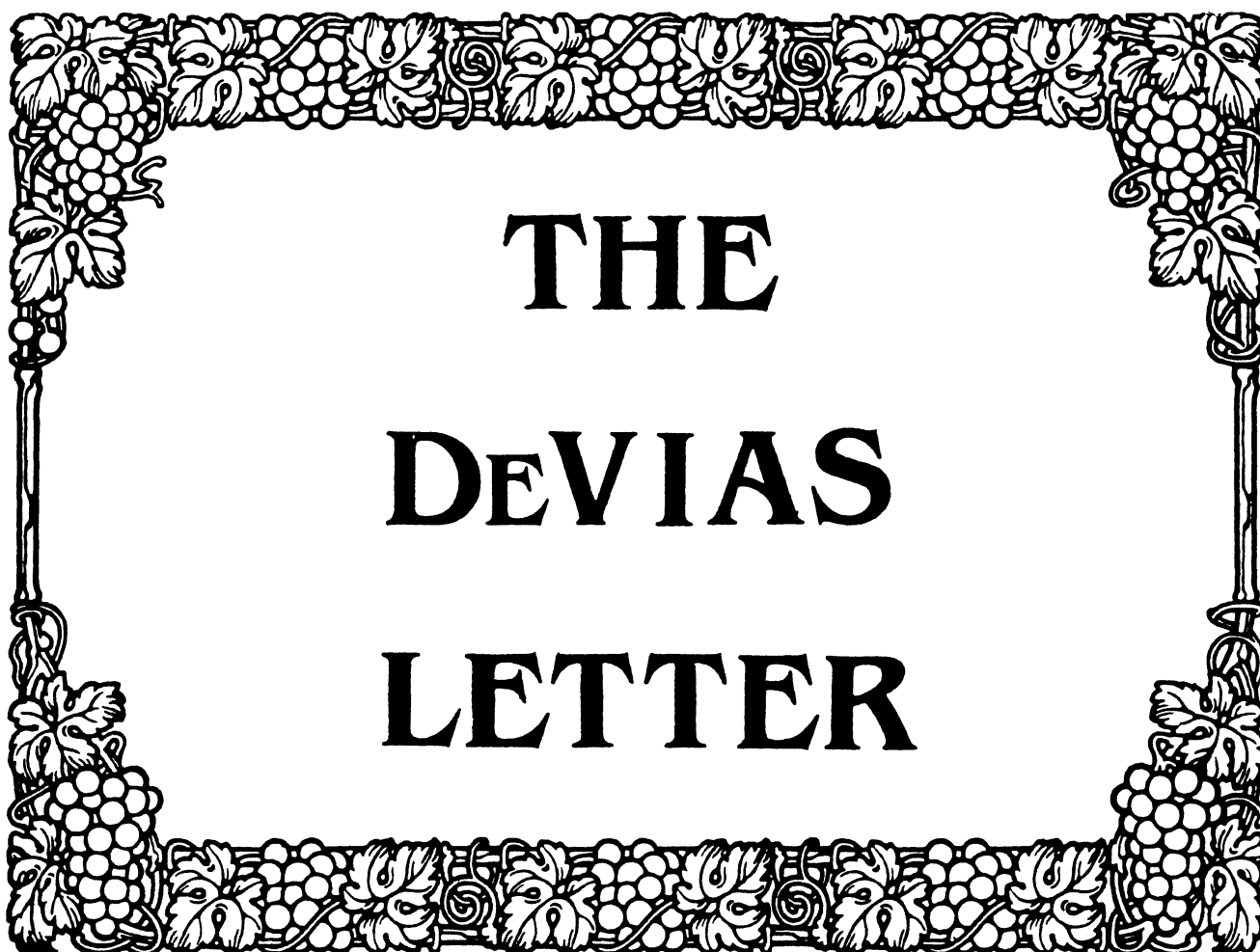
THE DeVIAS LETTER

RSX/IAS

```
                    THE DeVIAS Letter
         All the News That Fits, we print
    /\        /o o\                          /\        /o o\
   /  \      ( \ @/ )                       /  \      ( \ @/ )
  IAS  |      \ @/                         |      \ @/
       U |                                      U |
    \        '  )——Network—| o o o o |——  '  )
  /Mr Spot\    ))//)        Mr Vax                Rover \_)\)\)
 (_____)(_/(_/(_         VMS
```

FROM THE EDITOR'S KEYBOARD

The following editorial is solely the opinion of the author, and does not represent the views of DECUS or of Digital Equipment Corp. Responses to the Editor's remarks are heartily welcomed.

As you read this, fall DECUS will be fast approaching, but right now we are still in the hot days of summer, (although 100 degree cookers are now a thing of the past,) and we find ourselves loath to slave over a hot CRT terminal.

Looking at the preliminary fall program, I am even sorrier that I won't be in Anaheim. The schedule really looks like a full one. The merger of RSX and IAS back into one SIG has certainly presented a stronger program.

Anyway, if I want to get out in that boat this weekend, I better get this newsletter off the ground, so here goes.

IN THIS ISSUE

Our program of the month was originally written some years ago by Frank Penner for RSX, and adapted locally to run under IAS. It takes a look at the Logical Unit Table of a running task, and displays the open LUN information.

It's not a tool for the casual user, but a system operator can monitor the operation of a larger, longer program with ease. Try it on TKB the next time you link the TT handler.

It's also a good intro to the inner workings of the file system. File Control Blocks aren't really hard to understand, they are just not documented well by DEC. If you can understand this program, you get a DeVIAS Demon award.

CONTRIBUTION GUIDLINES

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

Paper submissions will always be accepted. Publishing may be delayed until the editor gets some time at the keyboard to convert them to our current format. We can accept submissions by FAX. Call for info.

Contributions may be submitted on tape, (800,1600, 3200 or 6250 BPI,) DEC-tape II, and DecMate or RT11 floppies. We're not fussy, we'll even accept paper tape or cards. We can read any IAS/RSX, RT11, VMS format. Any media sent to us will be promptly returned.

We have 2400/1200 baud modems on our IAS system and our VAX, with KERMIT for electronic submission. Give the editor a call @ (312)-791-8075 (preferably later in the day,) to obtain access info, etc. You can also submit over DCS, by sending mail to BORGER.

If you have a problem you would like to submit to the Devias Demon, send it to the Editor at the following address. Answers to problems from members (or anyone) should also be sent to the Editor at:

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

TEN YEARS AGO THIS MONTH

A transcription of the Spring RSX11M Q & A session had some interesting questions:

Q. Will PRESRV eventually support the RK06? (That combination wins an award as the ultimate combination of bad hardware with bad code . . . .ed.)

A. No, since PRESRV cannot handle bad blocks. (Now you new RSX/IAS users know what the early days were like.)

Q. Suggest RSX11M support shared code for tasks, i.e. the editor.

A. This is a very high-cost item... as most RSX11M utilities fit into 8K words due to overlaying. Shared tasks require no overlays, stretching the single task size out considerably.

Q. The LP05 outputs lower-case characters as spaces. Suggest you support a case-conversion option for the printer driver.

A. The idea is reasonable and will be considered for a future release.

Q. DSC does not restore large contiguous files as contiguous. Is this going to be fixed?

A. Yes, a solution is being worked on at this time.

Q. If you are careful, can you use DSC from a larger disk to a smaller disk?

A Yes, but DSC does not compress the file structure on the disk. The index file remains the same. As a result, when you go from an RP04 to an RK05, you may have very little RK05 left after the index file is copied.

And finally, an old friend, Roger Miles of DEC's Rolling Meadows office submitted a simple IAS monitoring task. Each time it ran it opened a data file and appended current data on ATL size, free pool, and free FCPCOM holes, to the file. Scheduled to run every n time units, it gave a good idea of when your system was straining at its limits.

```
THE PROGRAM OF
THE MONTH CLUB
```

A program to show any open luns of a running program. As all the programs of the month are, this program is available electronically via Kermit.

```
            .TITLE     LUT
.SBTTL      INTRO PAGE
;           PROGRAM LUT LOGICAL UNIT TABLE
;           WRITTEN BY:          FRANK PENNER
;                                JULY 1979
;           MOD BY :  FRANK BORGER TO IAS
;                                SEPT 1979
;           THIS PRIVILEGED TASK WILL PEEK AT THE LUT TABLE OF AN
;           ACTIVE, RESIDENT TASK AND DISPLAY VARIOUS AND SUNDRY
;           INFORMATION
;
;           CALLING SEQUENCE
;
;           MCR>LUT NNNNNN[/TI:TTNN]        WHERE
;                                           NNNNNN=TASK NAME
;                                           TTNN=TI OF TASK IF OTHER THAN YOUR OWN
;
            .MCALL     QIOW$,EXIT$S,DIR$,ALUN$
            .MCALL     F11DF$,CALL,RETURN,GMCR$
            .MCALL     FHDOF$
R$$11D=1
R$$IAS=1
            F11DF$                         ;DEFINE F11 OFSETS (VCB,FCB,WINDOW)
            FHDOF$     DEF$L               ;DEFINE FILE-HEADER BLOCK OFFSETS LOCAL

    .SBTTL DATA AREAS AND DPB'S
GMCR:       GMCR$
            TSKNAM     =GMCR+G.MCRB+4       ;START OF TYPED INPUT TASKNAME
TSKNAR:     .BLKW      2                    ;SPACE FOR RAD 50 TASKNAME
TSKATL:     .WORD      0                    ;SPACE FOR TASK ATL ADDRESS
TINAME:     .WORD      0                    ;ASCII TI NUMBER
TINUMB:     .WORD      0                    ;OCTAL TI NUMBER
LUN:        ALUN$      DLUN,0,0
NUMLUN:     .WORD      0                    ;SAVE # OF LUNS
PDR4K:      .WORD      77406                ;A 4K READ/WRITE PAGE DESCRIPTOR REGISTER
SAVPAR:     .WORD      0                    ;SAVED AREA FOR OLD PAR
SAVPDR:     .WORD      0                    ;SAVED AREA FOR OLD PDR
SAVACP:     .WORD      0                    ;ACP STD ADDRESS FOR CURRENT LUN
FCPPAR:     .WORD      0                    ;PAGE ADDRESS REGISTER FOR FCPCOM
FCPRAD:     .RAD50     /FCP/
COMRAD:     .RAD50     /COM/                ;RAD50 FCPCOM
PAR3=60000                                 ;OFFSET TO USE PAR/PDR SET 3
IOST:       .BLKW      2                    ;I/O STATUS BLOCK
QIODPB:     QIOW$      IO.WLB,TLUN,TEFL,,IOST,,<HED,HEDS,VFC>
            TLUN=      5
            TEFL=      2
            VFC=       40
OUTBUF:     .BLKB      132.
READLB:     QIOW$      IO.RLB,DLUN,DEFL,,IOST,,<HEDBUF,LEN,,BLKH,BLKL>
```

```
                DLUN=           1
                DEFL=           1
                BLKH=           0
                BLKL=           1
                LEN=            512.
HEDBUF:         .BLKB           512.
                MAX=20.                                 ;MAX OF 20 LUNS
;
;               WHAT A LOGICAL UNIT TABLE CONTAINS
;
;CONTENTS                               OFFSET          SIZE
;LUN                                    0.              2.
;PTR TO DEV NAME                        2.              2.
;DEV NUMBER                             4.              2.
;UIC                                    6.              4. (INIT CONTAINS ABS BL # OF FILE HEADER)
;FILE DESCRIPTOR                        10.             10.
;ACCESS COUNT                           20.             2.
;# OF RETRIVAL POINTERS                 22.             2.
;WINDOW SIZE                            24.             2.
;FIRST BLOCK #                          26.             4.
;NUMBER OF BLOCKS                       30.             2.
;
KLUDGE:         .BLKW           MAX                     ;1 OTHER WORD FOR ASCII DEVICE NAME
A=KLUDGE
X=1
ARGBLK:                                                 ;ARGUMENT BLOCK FOR EDMSG ROUTINE
                .REPT           MAX
                .WORD           X                       ;LOGICAL UNIT NUMBER
X=X+1
                .WORD           A                       ;2ND WORD OF ARGBLK POINTS TO DEV NAME
A=A+2
                .BLKW           14.                     ;15. WORDS (TOTAL) FOR EACH ARGBLK PACKET
                .ENDR


        .SBTTL ASCII DATA & BUFFERS
ISTRNG:         .ASCII          /%54<%47<%43<%38<%33<%16<%3</
                .ASCIZ          /%M%3>%2A%O:[%O,%O]%16>%X%33>%M%38>%M%43>%M%47>%M%M%54>%M/
HED:            .ASCII          /LUN/<11><11><11>/      ACC  RTV  WIND FIRST  # OF BLOCKS/
                .ASCII          <15><12>
                .ASCII          <11><11><11>/       CNT  PTRS SIZE BLOCK  MAPPED/
HEDS=.-HED
                .EVEN


        .SBTTL ASCII ERROR MESSAGES
ERRMS0:         .ASCII          /*** LUT SYNTAX ERROR IN COMMAND ***/
                ERRLN0=.-ERRMS0
ERRMS1:         .ASCII          /*** LUT TASK NOT ACTIVE AT SPECIFIED TI ***/
                ERRLN1=.-ERRMS1
ERRMS2:         .ASCII          /*** LUT TASK ON MRL (CAN'T READ TASK HEADER) ***/
                ERRLN2=.-ERRMS2
ERRMS3:         .ASCII          /*** LUT CAN'T FIND PROPER ATL ADDRESS FOR FXXACP ***/
                ERRLN3=.-ERRMS3
                .EVEN


        .SBTTL GET AND DECODE COMMAND LINE
START:
                DIR$            #GMCR                   ;GET COMAND LINE
                MOV             #GMCR+2,R0              ;POINT TO START OF COMMAND LINE
                MOV             R0,R1                   ;SET TO MAKE END OF LINE POINTER
                ADD             @#$DSW,R1               ;HAVE END POINTER
1$:             CMPB            (R0)+,#40               ;FOUND FIRST SPACE AFTER "INF"
                BEQ             2$                      ;YES
                CMP             R0,R1                   ;PAST END OF COMMAND
                BLE             1$                      ;YES
99$:            JMP             ERROR                   ;NO, AN ERROR
2$:                                                     ;POINTING AT TASK NAME
                MOV             #1,R1                   ;. IS LEGAL CHARACTER
                JSR             PC,$CAT5                ;CONVERT TO RAD 50
                BCC             111$                    ;BR IF OK
                CMPB            R2,#'/                  ;TERMINATED ON / OF SWITCH ?
                BNE             111$                    ;NO
                DEC             R0                      ;PUT R0 AT TERMINATOR
111$:           MOV             R1,TSKNAR               ;FILL IN FIRST HALF OF NAME
                MOV             #1,R1                   ;DO AGAIN FOR 2ND HALF
                JSR             PC,$CAT5
                BCC             222$                    ;BR IF OK
                CMPB            R2,#'/                  ;TERMINATED ON / OF SWITCH ?
                BNE             222$                    ;NO
```

```
                DEC        R0                       ;PUT R0 AT TERMINATOR
222$:           MOV        R1,TSKNAR+2              ;FILL IN 2ND HALF
                CMPB       (R0),#'/                 ;TERMINATED ON SWITCH ?
                BNE        MYTI                     ;IF NOT USE MY TI
                INC        R0                       ;BUMP PAST "/"
                CMPB       (R0)+,#'T                ;CHECK SYNTAX
                BNE        99$
                CMPB       (R0)+,#'I
                BNE        99$
                CMPB       (R0)+,#':
                BNE        99$
                MOVB       (R0)+,TINAME             ;SAVE NAME
                MOVB       (R0)+,TINAME+1
                JSR        PC,$COTB                 ;CONVERT UNIT NUMBER TO OCTAL
                MOV        R1,TINUMB                ;SAVE TI NUMBER
                BR         FINDTA                   ;AND FIND TASK
MYTI:           MOV        .CRTSK,R0  ;GET MY TI
                MOV        A.TI(R0),R0              ;GET MY PUD POINTER
                MOV        U.DN(R0),TINAME          ;SAVE MY TI NAME
                MOVB       U.UN(R0),TINUMB          ;AND NUMBER

        .SBTTL SEARCH ATL FOR TASK IN QUESTION
FINDTA:                                            ;NOW CAN SEARCH ATL FOR TASK
                MOV        #.ATLLH,R0               ;R0 POINTS AT ATLLH
LOOP:           MOV        (R0),R0                  ;GET NEXT ATL LISTHEAD
                CMP        R0,#.ATLLH               ;THROUGH LIST ?
                BNE        1$                       ;BR IF NOT
                JMP        ERROR1                   ;ELSE AN ERROR
1$:             MOV        A.TD(R0),R1              ;GET STD
                CMP        S.TN(R1),TSKNAR          ;DO NAMES MATCH
                BNE        LOOP                     ;NO, TRY AGAIN
                CMP        S.TN+2(R1),TSKNAR+2
                BNE        LOOP                     ;NO TRY AGAIN
                MOV        A.TI(R0),R1              ;GET TI POINTER

                CMP        U.DN(R1),TINAME          ;DO TI'S MATCH
                BNE        LOOP
                CMPB       U.UN(R1),TINUMB          ;LAST TEST
                BNE        LOOP                     ;NO MATCH
                MOV        R0,TSKATL                ;SAVE TASK ATL JUST IN CASE
                                                   ;AND TRY TO MAP TO TASK
                CMP        A.TS(R0),#TS.MRL         ;IS TASK IN CORE
                BLT        GETCOM                   ;YES, TRY
                CMP        A.TS(R0),#TS.MRR         ;CHECK UPPER LIMIT
                BGT        GETCOM                   ;YES, TRY
                JMP        ERROR2                   ;NO, TASK IS ON MRL SO REPORT
GETCOM:                                            ;GET DATA FOR ACPCOM
                MOV        #.GCDLH,R5               ;GET START OF GCD
NXTCOM:         MOV        (R5),R5                  ;GET NEXT/FIRST GCD
                CMP        R5,#.GCDLH               ;END OF GCD ?
                BNE        99$                      ;NO CHECK A GCD ENTRY
                EXIT$S                              ;EXIT IF CAN'T FIND ACPCOM
99$:            CMP        G.BN(R5),FCPRAD          ;IS THIS FCPCOM ?
                BNE        NXTCOM                   ;NO
                CMP        G.BN+2(R5),COMRAD        ;IS THIS FCPCOM
                BNE        NXTCOM                   ;NO
                MOV        G.BA(R5),FCPPAR          ;YES, SAVE HIS CORE ADDRESS
                MOV        PDR4K,-(SP)              ;MAP TO THE TASK HEADER
                MOV        A.HA(R0),-(SP)           ;PDR FOR TASK IN QUESTION TO STACK
                JSR        PC,..SPD3                ;MAP PAR/PDR 3 TO TASK HEADER
                MOV        PAR3+H.LUT,R1            ;POINT R1 TO TASK LOG UNIT TABLE
                CMP        R1,#MAX                  ;IS # OF LUNS GREATER THAN MAX ?
                BLE        1$                       ;IF NOT GREATER THAN O.K.
                MOV        #MAX,R1                  ;ELSE, SET # OF LUNS AT MAX
1$:             MOV        R1,NUMLUN                ;SAVE # OF LUNS FOR PRINT LATER ON
                MOV        #PAR3+H.LUT+2,R0         ;POINT R0 TO FIRST PUD/WINDOW PAIR
                MOV        #ARGBLK,R2               ;POINT R2 TO TEMP STORAGE AREA

        .SBTTL SAVE DEV NAME, WINDOW BLOCK AND FCB FOR EACH LUN
PROLOP:                                            ;BEGIN PROCESSING OF DCB AND UCB
                ADD        #2,R2                    ;BUMP R2 OVER LUN #
                MOV        (R0)+,R5                 ;R5 NOW POINTS TO PUD
                BEQ        4$                       ;BRANCH, IF NULL LUN
                CMP        R5,#1                    ;IS IT TI ?
                BNE        1$                       ;NO
                MOV        #"TI,@(R2)+               ;FAKE A TI DEVICE
                CMP        (R2)+,(R0)+              ;BUMP PAST DEVICE INFO IN ARG BLOCK
                BR         5$                       ;AND PROCESS AS IF NO FILES OPEN
```

RSX/IAS-4

```
1$:        MOV     U.DN(R5),@(R2)+      ;SAVE DEVICE NAME
           MOVB    U.UN(R5),(R2)+       ;PUT OCTAL DEV # IN BUFFER
           JSR     PC,..REDS            ;NOW FOLLOW RE-DIRECT CHAIN TO DISK
           MOV     U.ACP(R5),SAVACP     ;SAVE ACP STD ADDRESS FOR LATER
           TSTB    (R2)+                ;WORD ALIGN AGAIN
                                        ;BEGIN PROCESSING OF WINDOW BLK AND FCB
           MOV     (R0)+,R4             ;POINT R4 TO WINDOW BLOCK
           BEQ     5$                   ;IF NULL THEN NO WINDOW BLOCK
           CMP     R4,#100000           ;IS THIS WINDOW BLOCK IN SCOM ?
           BLO     12$                  ;NO, ITS IN FCPCOM, SO RE-MAP US
           JSR     PC,WBSAV             ;YES - SAVE WINDOW BLOCK INFO
           MOV     W.FCB(R4),R5         ;AND GET FCB ADDRESS NOW
           BR      66$                  ;BRANCH TO SAVE FCB INFO
12$:       CMP     R4,#60000            ;IS WINDOW BLOCK IN FCPCOM OR ACP
           BLO     11$                  ;BRANCH IF IT'S IN THE FCP ITSELF
           JSR     PC,SWAFCP            ;ELSE SWAP TO FCPCOM
           BR      13$
11$:       JSR     PC,SWAACP            ;SWAP TO THE ACP
           ADD     #60000,R4            ;AND FAKE OUT SO WE USE PAR/PDR 3
13$:       JSR     PC,WBSAV             ;SAVE WINDOW BLOCK INFO
           MOV     W.FCB(R4),R5         ;SAVE FDB ADDRESS
           JSR     PC,RESTAS            ;RESTORE TO TASK HEADER
66$:       TST     R5                   ;CHECK FCB POINTER
           BEQ     5$                   ;IF NULL THEN NO FCB
           CMP     R5,#100000           ;IS THIS FCB IN SCOM ?
           BLO     2$                   ;NO, ITS IN FCPCOM, SO RE-MAP US
           JSR     PC,FCBSAV            ;YES - SAVE FCB BLOCK INFORMATION
           BR      6$                   ;BRANCH TO END OF SOB LOOP
2$:        CMP     R5,#60000            ;IS THIS FCB IN FCPCOM OR THE ACP
           BLO     21$                  ;BRANCH IF ITS IN THE ACP
           JSR     PC,SWAFCP            ;SWAP TO FCPCOM
           BR      22$
21$:       JSR     PC,SWAACP            ;SWAP TO THE ACP
           ADD     #60000,R5            ;AND FAKE THINGS SO WE USE PAR/PDR 3
22$:       JSR     PC,FCBSAV            ;SAVE FCB INFO
           JSR     PC,RESTAS            ;RESTORE BACK TO TASK
           BR      6$                   ;BRANCH TO END OF SOB LOOP
4$:
           MOV     #20040,@(R2)+        ;PUT 2 SPACES IN ASCII DEV NAME BUFFER
           CMP     (R2)+,(R0)+          ;BUMP PAST DEVICE INFO IN ARGBLK
5$:
           ADD     #16.,R2              ;BUMP POINTER PAST FCB SAVE AREA
6$:
           ADD     #10.,R2              ;SKIP ALREADY PUT IN ARGBLK RTV PTRS
           DEC     R1                   ;COUNT A LUN
           BLE     99$                  ;BR IF DONE
           JMP     PROLOP               ;GO BACK FOR MORE
99$:

     .SBTTL ASSEMBLE AND PRINT FILE INFORMATION
                                        ;END OF GATHERING INFO
                                        ;BEGIN PRINTOUT
           DIR$    #QIODPB              ;PRINT OUT HEADING ON TERMINAL
           MOV     #OUTBUF,QIODPB+Q.IOPL ;PUT ADDRESS OF OUT BUFFER IN QIODPB
           MOV     #ARGBLK,R2           ;PUT ADDRESS OF ARGBLK IN R2 FOR EDMSG
           MOV     NUMLUN,R3            ;# OF LUNS IN R3 FOR SOB LOOP
PRTLOP:
           CLR     R0                   ;USE R0 AS FLAG FOR NO FILE SPEC
           ADD     #2,R2                ;BUMP TO DEVICE NAME
           MOV     @(R2)+,LUN+A.LUNA    ;PUT ASCII DEV NAME IN ATTACH LUN DPB
           MOV     (R2)+,LUN+A.LUNU     ;PUT BINARY UNIT # IN ATTACH LUN DPB
           MOV     (R2)+,READLB+Q.IOPL+6. ;PUT HIGH BLOCK # IN DPB
           BNE     1$                   ;IF NOT ZERO DON'T SET FLAG
           MOV     #1,R0                ;SET FLAG FOR NO FILE SPEC
1$:        MOV     (R2),READLB+Q.IOPL+8. ;PUT OTHER LOW BLOCK # IN DPB
           BNE     2$                   ;A FILE SPEC SO BRANCH
           TST     R0                   ;WAS FLAG SET?
           BEQ     2$                   ;NOT SET SO VALID FILE SPEC
           SUB     #8.,R2               ;BUMP BACK R2 TO POINT TO START OF PACK
           BR      4$                   ;SKIP FILE SPEC STUFF
2$:        TST     -(R2)                ;BACK UP ARGBLK POINTER
           DIR$    #LUN                 ;ASSIGN DISK LUN TO PROPER DEVICE
           DIR$    #READLB              ;READ FILE HEADER BLOCK
           MOVB    HEDBUF+H.PROJ,(R2)+  ;PUT PROJECT # IN ARG BLK
           MOVB    #0,(R2)+             ;ROUND ARGBLK POINTER TO EVEN WORD
           MOVB    HEDBUF+H.PROG,(R2)+  ;PUT PROGRAMMER # IN ARG BLK
           MOVB    #0,(R2)+             ;ROUND ARGBLK POINTER TO EVEN WORD
           MOV     #HEDBUF+S.HDHD,R0    ;PUT ADDRESS OF FILESPEC IN R0
```

RSX/IAS-5

```
                MOV         #5,R1                   ;PUT # OF WORDS TO TRANSFER IN R1
3$:             MOV         (R0)+,(R2)+             ;FILESPEC FROM HEADER BLK TO ARGBLK
                SOB         R1,3$                   ;DO WHOLE FILESPEC
                SUB         #20.,R2                 ;RESET R2 TO BEGINNING OF PACKET
4$:             MOV         #OUTBUF,R0              ;ADDRESS OF OUTPUT BUFFER FOR EDMSG
                MOV         #ISTRNG,R1              ;INPUT STRING FOR EDMSG
                CALL        $EDMSG                  ;CALL EDIT MESSAGE ROUTINE
                MOV         R1,QIODPB+Q.IOPL+2      ;SIZE OF MESSAGE TO PRINT
                DIR$        #QIODPB                 ;PRINT OUT INFO ON TERMINAL
                SOB         R3,PRTLOP               ;PRINT OUT ALL LUNS
EXIT:           EXIT$S

        .SBTTL ERROR HANDLING
;
;
ERROR:                                              ;COMMAND STRING SYNTAX ERROR
                MOV         #ERRMS0,QIODPB+Q.IOPL
                MOV         #ERRLN0,QIODPB+Q.IOPL+2
                BR          ERRCOM
;
ERROR1:                                             ;TASK NOT IN ATL
                MOV         #ERRMS1,QIODPB+Q.IOPL
                MOV         #ERRLN1,QIODPB+Q.IOPL+2
                BR          ERRCOM
;
ERROR2:                                             ;TASK ON MRL SO HEADER NOT AVAILABLE
                MOV         #ERRMS2,QIODPB+Q.IOPL
                MOV         #ERRLN2,QIODPB+Q.IOPL+2
                BR          ERRCOM
;
ERROR3:                                             ;COULDN'T FIND PROPER ACP ATL ADDRESS
                MOV         #ERRMS3,QIODPB+Q.IOPL
                MOV         #ERRLN3,QIODPB+Q.IOPL+2
;
ERRCOM:         DIR$        #QIODPB                 ;REPORT ERROR
                EXIT$S                              ;AND LEAVE

        .SBTTL SAVE WINDOW BLOCK AND FCB SUBROUTINES
;ENTER BOTH ROUTINES WITH R2 OFFSET 4. BYTES INTO DATA BUFFER
WBSAV:                                              ;SAVE WINDOW BLOCK ROUTINE
                ADD         #16.,R2                 ;POINT TO RETRIVAL PTRS IN ARGBLK
                MOVB        W.CTL(R4),R3            ;PUT # OF RTV PTRS IN R3
                MOV         R3,(R2)+                ;PUT # OF RTV PTRS IN ARGBLK
                MOVB        W.WISZ(R4),R3           ;WINDOW SIZE IN R3
                MOV         R3,(R2)+                ;PUT WINDOW SIZE IN ARGBLK
                MOVB        W.VBN(R4),R5            ;PUT HIGH BYTE OF VBN IN R5
                MOV         R5,(R2)+                ;PUT HIGH WORD OF VBN IN ARGBLK
                MOV         W.VBN+2(R4),(R2)+       ;PUT LOW WORD VBN IN ARGBLK
                CLR         (R2)                    ;CLEAR ARGBLK SUM OF # OF MAPPED BLOCKS
                TST         R3                      ;R3 HAS # OF RETRIVAL PTRS
                BEQ         2$                      ;IF ZERO THEN SKIP SOB ROUTINE
                CLR         -(SP)                   ;USE STACK FOR BYTE TO WORD CONVERSION
                MOV         R4,R5                   ;PUT WINDOW BLOCK ADDRESS IN R5
                ADD         #W.RTRV,R5              ;POINT R5 TO FIRST RETRIEVAL POINTER
1$:             ADD         (R5),(R2)               ;ADD COUNT TO TALLY OF # OF BLOCKS
                ADD         #6,R5                   ;POINT TO NEXT VBN COUNT
                SOB         R3,1$                   ;DO ALL THE RTV PTRS
                TST         (SP)+                   ;RESET STACK POINTER
2$:             SUB         #24.,R2                 ;POINT R2 BACK TO FILE SPEC ARGBLK
                RTS         PC                      ;END OF WINDOW BLOCK SAVE ROUTINE
;
;
FCBSAV:                                             ;SAVE FCB SUBROUTINE
                MOV         F.HDLB(R5),(R2)+        ;SAVE HIGH HEADER-BLOCK # IN ARGBLK
                MOV         F.HDLB+2(R5),(R2)+      ;SAVE LOW HEADER-BLK # IN ARGBLK
                ADD         #10.,R2                 ;SKIP 10 WORDS OF ARG BLK
                MOVB        F.NACS(R5),(R2)+        ;SAVE ACCESS COUNT IN ARGBLK
                TSTB        (R2)+                   ;POINT ARGBLK TO WORD AND NEXT PACKET
                RTS         PC                      ;END OF SAVE FCB SUBROUTINE
;

        .SBTTL SUBS TO SWAP TO FCPCOM AND BACK AGAIN
SWAFCP:         MOV         PDR4K,-(SP)             ;SET TO MAP TO FCPCOM
                MOV         FCPPAR,-(SP)            ;WITH PREV SAVED ADDRESS
                JSR         PC,..SPD3               ;SWAP PAR/PDR 3
                MOV         (SP)+,SAVPAR            ;SAVE PAR AND RESTORE STACK
                MOV         (SP)+,SAVPDR            ;SAVE PDR AND RESTORE STACK
                RTS         PC                      ;AND RETURN
```

```
;
RESTAS:    MOV        SAVPDR,-(SP)        ;PUT OLD PDR ON STACK
           MOV        SAVPAR,-(SP)        ;PUT OLD PAR ON STACK
           JSR        PC,..SPD3           ;AND SWAP BACK TO OLD VALUES
           CMP        (SP)+,(SP)+         ;AND RESTORE STACK
           RTS        PC                  ;AND RETURN
;
;IF ITS THE ACP WE HAVE A LITTLE MORE WORK
;
SWAACP:    MOV        R0,-(SP)
           MOV        #.ATLLH,R0          ;R0 POINTS AT ATLLH
1$:        MOV        (R0),R0             ;GET NEXT ATL LISTHEAD
           CMP        R0,#.ATLLH          ;THROUGH LIST ?
           BNE        2$                  ;NO, OK
           JMP        ERROR3
2$:        CMP        A.TD(R0),SAVACP     ;IS THIS CORRECT ATL ENTRY ?
           BNE        1$                  ;NO, TRY AGAIN
           MOV        A.HA(R0),R0         ;YES, GET ADDRESS OF FCP
           ADD        #4,R0               ;F12ACP HEADER IS 400 BYTES
           MOV        PDR4K,-(SP)         ;SET TO MAP TO FXXACP
           MOV        R0,-(SP)
           JSR        PC,..SPD3
           MOV        (SP)+,SAVPAR        ;SAVE PAR
           MOV        (SP)+,SAVPDR        ;AND PDR
           MOV        (SP)+,R0
           RTS        PC
           .END       START
```

The RSX Multi-Tasker

October, 1988

"SATCHL: .WORD   0   ; Go White Sox"

Fine Realtime Commentary Since 1975

Table of Contents

Food for Thought

"Why level downward to our dullest perception always, and praise that as common sense? The commonest sense is the sense of men asleep, which they express by snoring."

> - Henry D. Thoreau
> "Walden, or Life in the Woods"

The Editor's Corner

Bruce R. Mitchell

Thanks in no small part to massive efforts by the West Coast members of the SIG, we once again have enough material to publish an issue of the Multi-Tasker.

Hint, hint. Gentle reminder.

It's no big deal, ladies and gentlemen, to sit down at your terminal for half an hour, pound out a raw article, put it on floppy, and send it to the Editor. Some of you even have departmental secretaries who can put the stamp on the floppy mailer. From that point on out, the Editor takes over, corrects spelling and grammar, checks for technical correctness, formats to fit Newsletter specifications, sends in finished copy.

I really don't mind doing these things but you've got to send me something to work with. We need submissions from RSX rookies and casual users. After ten years in the saddle the SIG leadership can't write introductory articles worth toad squat. That is, when I can even get them to write anything.

Well, Larry Baker weighs in with an excellent pair of articles this month for all you driver writers. The Editor blushes to disclose that he has a little article on the V4.0

revision to the round-robin scheduler. And DEC responds to some
unanswered questions from previous symposia Q&A sessions.

So here we go once more around the track, but first ...
it's a twofer this time ... one serious ("editoria seria"), and
one funny ("editoria funnya").

----- Tribute -----

As faithful readers of these pages know by now, the Editor
bends the knee to no man. "Civis Romanus sum" is the unspoken
motto of the Editor. This month the Editor breaks that tradition
to offer tribute. An honorable tribute to an honorable man.

All great organizations owe their continued existence to
those individuals of clear sight and uncompromising virtue who
refuse to sell their purpose. For the laws of thermodynamics
apply to organizations as well as to physical systems, and to
preserve order in the face of entropy demands resolution of the
highest sort.

It is to such an individual I would now commend your
attention as members of this group we call DECUS.

In the face of entrenched bureaucracy, this man stepped
forth alone to stand for the right. His election to high office
reminded many that office is held by the consent of the governed,
and not by permit, writ, charter nor indulgence.

He actively sought counsel of the general membership in
making decisions affecting all. When control was slowly and
subtly wrought from the membership, he worked to return it to
them.

When he was but one man among many, his voice was heard
clear and true. When offered tainted bargains, he spurned them.
The record shows that he did not compromise the high ideals he
presented when he offered us his efforts in service of the higher
vision.

Today we begin to see the fruits of his efforts emerging.
If he chooses to rest from his labor, why, it is well deserved,
and bought and paid for manyfold.

"If we see farther and more clearly, it is because we stand
on the shoulders of giants." Robert F. Curley is such a man.

----- Uncut Gems from Your Leadership -----

Many of you Multi-Tasker readers may not be aware that the DECUS leadership publishes its own newsletter. This newsletter is called SUGgestions, and appears about monthly to bimonthly. There are lots of interesting things in SUGgestions. Regrettably you never see most of them. The Editor culls some of what he considers the best from the June issue for this month's editorial.

Page 6. "A new mission statement was developed to read as: 'The mission of the Management Council is to oversee the management of the business, operations, and resources of the U.S. Chapter in support of the Chapter's strategic directions.'" [Ed: The Board of Directors must be glad to have that load taken from their shoulders ...]

Page 26. "EDITORIAL POLICY: It was stated that editorials can be in the newsletters, as long as editorial 1 Are clearly identified as an editorial. 2 Include an appropriate disclaimer 3 Provide opportunity for timely rebuttal 4 Clearly identify author(s) of the editorial. (Final text of the editorial guidelines will be distributed via DCS.) [Ed: My cup runneth over ...]

But don't take my word for it. After all, DECUS is your society. I showed these to a couple of SIG members - a fella name of Jeffrey Bostwick gave me a succinct response which I cannot better:

"Ghaaaaaa!"

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

Please submit machine readable media. RX01/2, RX50 and 9 channel 800/1600 BPI magtape are preferred. Almost any medium I can't read can be converted. All RSX volume formats are acceptable, but please don't send VMS Backup or ODS-2 format media.

You can also submit articles through the RSX bulletin board system at (612) 777-7664. The Editor loves you if you do so. Kermit the file in and send it via MAIL to username MULTITASKER.

Or you can submit via BITNET. Send articles to CICHANOWSKI@MSUS1 and precede the article with "Please forward to WSU::WNIMHACKR".

Submissions which aren't machine readable may take
longer to get into print. If you format your submission in
RUNOFF, please set page size 58,80; left margin 10; right
margin 75; and, when changing margins, use incremental
changes rather than absolute. The editor thanks you for the
consideration.

Send your articles and other submissions to the
luxurious all-new Multi-Tasker offices:

Bruce R. Mitchell
Machine Intelligence and Industrial Magic
RR #1, Box 216
Fountain City, MN   54629

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

... this month in Pool Lowbegone, where the new
leadership is strong, our IAS brothers are handsome, and the
outlook for the future is above average.

Event Flags, Significant Events and V4.0 M-Plus

Bruce R. Mitchell
Machine Intelligence and Industrial Magic
RR #1, Box 216
Fountain City, WI   54629

[Editor's note: The following is an extract from a
forthcoming RSX programming textbook and is copyright (c)
1988, Bruce R. Mitchell and Denny J. Walthers. Used by
permission.]

Keep the following fact in your mind constantly when you
use event flags for intertask synchronization and control:

FIDDLING WITH EVENT FLAGS DOES NOT CAUSE A SIGNIFICANT EVENT.

Now, we're going to repeat that to make sure you remember it,
because it's very important and the majority of RSX
programmers don't seem to know it. I tell you three times,
and what I tell you is true:

SETTING AND CLEARING EVENT FLAGS DOES NOT, DOES *NOT*, DOES
ABSOLUTELY *** NOT *** CAUSE A SIGNIFICANT EVENT.

This is very, very important to you as an RSX programmer.

Don't ever forget it. Here is why it is important.

There are only a few things in the RSX system that can cause a context switch from the current task to a new task. Setting and clearing event flags is not one of those things.

By way of example: Task A sets global event flag 36 and then stops itself. Task B is waiting for global event flag 36. Task B may wait for hell to freeze over before continuing, if there is little or no I/O going on in the host system. I/O causes significant events, which cause the scheduler to reevaluate which tasks are eligible to run, which hopefully eventually causes task B to continue from its wait state. Simply setting the event flag does NOT cause the scheduler to run.

In releases of RSX-11M-Plus and Micro/RSX prior to V4.0, this is not often a major problem due to the way the Exec round-robin algorithm works in those releases. The round-robin code causes a scheduler run at the expiration of every round-robin interval, so the longest a task can wait for a global event flag is one round-robin interval. This is not excellent response, but is often acceptable if the ultimate in response is not required.

In V4.0 of RSX-11M-Plus and Micro/RSX, the round-robin algorithm is changed to improve system performance. The new round-robin code does not cause a scheduler run at the expiration of a round-robin interval if there is no contention for the CPU. In other words, if at least two tasks of the same priority within the round-robin priority limits are not competing, no scheduler run occurs at the expiration of the round-robin interval. Tasks waiting for global event flags can wait for minutes or hours in systems with low I/O activity.

Contrary to some opinions, Digital did not deliberately "break" M-Plus. The problem exists in the user code due to violation of RSX design rules. It was previously masked by the round-robin Exec code.

One symptom typical of this problem in systems with accounting active is that the host system "breaks loose" every five minutes. This is caused by execution of accounting temporary file updates, which cause I/O, which in turn cause scheduler runs.

Running RMD on the system to find the problem causes the problem to go away, of course. RMD updates its terminal screen once per second, ensuring that at least one scheduler run occurs each second.

The solution to this problem is very simple. Remember it and you'll never be burned by response time due to event flag waits.

WHEN SETTING OR CLEARING AN EVENT FLAG WHICH MAY AFFECT
ANOTHER TASK, USE DECL$ AFTER THE SETF$ / CLEF$ TO DECLARE A
SIGNIFICANT EVENT AND FORCE A SCHEDULER RUN.

Fall 1987 DECUS Q&A Update

Brian S. McCarthy
Digital Equipment Corporation

There were a number of questions and requests from the
QA session at Fall DECUS in Anaheim that we were unable to
answer on the spot. Below is a list of these problems and
suggestions along with the responses from the RSX Development
group.

Q: A terminal is initially set up as full duplex. After a
   user logs in, it is set to half duplex.

A: There are 2 situations in which a terminal is reset to
   half duplex. When a user logs out, BYE resets the
   terminal to half duplex, and it remains that way when the
   next user logs in. If a LAT terminal is set to full
   duplex when no one is logged on, the initialization for
   the LAT terminal also resets the terminal to half duplex.
   This is the expected and desired behavior in the general
   case.

Q: The documentation for the logical name directives in the
   Executive Reference Manual is different from the
   documentation in the Mini-Reference Guide.

A: The documentation in the Mini-Reference Guide contains a
   number of technical mistakes. Please refer to the
   Version 4.0 RSX-11M-Plus and Micro/RSX Executive
   Reference Manual for the correct information.

Q: There are certain tasks on the M-Plus kit in [3,54] which
   are not documented such as MINUTL.TSK. There are other
   files which are unnecessary such as DCL.STB and
   F11MSG.STB.

A: Certain tasks which are specific to Micro/RSX were
   shipped with the V4.0 RSX-11M-Plus kits. These tasks are
   removed from the RSX-11M-Plus V4.1 kit. Other
   unnecessary files are also removed.

Q: Is there any possibility that loadable crash drivers will be supported under RSX-11M-PLUS? If so, which devices will be allowed?

A: Support for loadable crash drivers is included in RSX-11M-Plus V4.1. The devices supported are DU:, MU:, MS:, DL: and MM:.

Q: Why have the sources for Indirect been removed from the kits?

A: We have traditionally distributed the sources for privileged components on the kit. All sources for non-privileged utilities were (and are) available only from the source kit, purchased separately.

During the V4.0 development cycle, we moved around certain components on our master packs so there wouldn't be so much overlap in certain directories. The sources for Indirect were moved from UFD [12,10].

The reason they have been shipped in the past is simply because they resided in the same directory as MCR and the entire directory was copied from our master packs to the kit. These sources are now available with the rest of the non-privileged sources on the source kit. We will consider adding these back to the kit in a future release for your convenience.

Q: There are a number of new ELI switches which are undocumented. These are useful and provide an easier interface to error logging. Will these be documented for M-Plus?

A: There were a number of enhancements made for Micro/RSX to allow use of the error logging system by non-technical users. to common sources, these changes also became a part of the RSX-11M-Plus V3.0 kits.

However, there are a number of limitations with this support under RSX-11M-PLUS. The only devices currently supported are those which are supported under Micro/RSX. Also, there is additional overhead in logging an error when the new options for Micro/RSX are used. For these reasons, the support was not documented for RSX-11M-Plus and was turned off in RSX-11M-Plus V4.0. We will consider expanding and incorporating this support into a future release of RSX-11M-PLUS.

Q: EDTRES bombs out with a small (30 line) initialization files or when too many buffers are opened. EDT and EDTFSL seem to work fine. Please either fix EDTRES or

delete it from the kits.

A: Using the resident library configuration of EDT
(EDTRES.ODL) and having multiple DEFINE KEY commands
(usually associated with an EDT initialization file)
substantially reduces EDT's performance. In addition
multiple DEFINE KEY commands and/or numerous open buffers
may cause EDT to abort with a Memory Protect Violation
error message. This restriction is documented on page
A-5 of the EDT Editor Manual.

Even with these limitations, the EDTRES configuration is
still useful on limited memory systems. Therefore, it
should continue to be made available.

The non-resident configuration overlay structures improve
EDT's performance by enabling more DEFINE KEY commands to
be used. We recommend that users with large EDT
initialization files as mentioned above use a
non-resident EDT configuration. This configuration may
be installed as a separate EDT task.

Q: When building an RSX-11M-Plus Executive without support
for external headers, the symbol $HEADR comes up
undefined for DIR11M and DR211M.

A: This problem is cosmetic only and does not affect the
functioning of your system. It is corrected in V4.1 of
RSX-11M-Plus.

Q: How about changing the default extension on batch command
files from .CMD to .BAT?

A: We believe that this is useful, and plan to implement
this change in a future release. The new behavior will
look for .BAT files first, and if not found, will default
back to .CMD files. This ensures that applications that
depend on the .CMD default continue to work.

Q: There should be no HELP allowed for non-logged in
terminals. Or, a single help file should always be
forced.

A: We will look into a method for making HELP optionally
disable a user at a non-logged in terminal from obtaining
HELP text.

Q: You cannot specify a directory for the VFY command.

A: We are unable to reproduce this problem. VFY does accept
a device, directory and file specification for the output

list file. If this problem still exists under V4.0,
please send us an SPR with the appropriate command lines
which will reproduce the problem.

More Programming Techniques for User Written Drivers
(an addendum to "Timer Support for User Written Drivers")

Lawrence M. Baker
U. S. Geological Survey
345 Middlefield Road   MS977
Menlo Park, CA   94025

I would like to add some comments to Jim Bostwick's excellent
article in the April 1988 issue of the Multi-Tasker, "Timer
Support for User Written Drivers" [1].   It so happens that   I am
in the process of writing a communications device driver for
RSX-11M/M-Plus which makes use of the clock queue services he
describes.   He mentions that one must not unload a driver which
has a clock block in the clock queue.   This is normally true.
However, there are several ways to get around this problem, which
are described in the sections below.

Additionally, Jim stated that the driver service routine must
save and restore all registers. By my reading of the code (in
module TDSCH), no register saves and restores are required.   R4
is always refreshed from the clock queue listhead pointer,
$CLKHD, and R5 is picked up from the "request type" field (C.RQT)
in the clock block.   Any other values used in TDSCH are
calculated using these values. Furthermore, on a multi-processor
RSX-11M-Plus system, if the clock block specifies a Unibus-Run
Mask, TDSCH must conditionally fork to the correct processor,
which never saves anything but R4 and R5. To me, that's a dead
giveaway that the rest of us don't have to worry either.

Finally, while I've got your attention, I thought I would
describe another useful technique I have devised to protect an
RSX-11M system when a driver is unloaded with device interrupts
enabled.

I should warn you all up front, however, that I rarely have a
chance to test my code on an RSX-11M system. Usually, I enable
the code on an RSX-11M-Plus system and test it that way.   I try
to write all my drivers to be compatible with both RSX-11M and
RSX-11M-Plus, but since we no longer have any RSX-11M systems
under software maintenance, we no longer have a current RSX-11M
distribution kit.   I welcome your comments, either personally   or
through the Multi-Tasker, on on these ideas.

Unloading Drivers With Active Clock Queue Requests

1 RSX-11M-Plus Drivers

If a driver only needs a single clock block to service all
devices, e.g., for implementing I/O request timeouts in a driver
that maintains its own internal queues, the RSX-11M-Plus
Controller Table (CTB) may have an 8-word clock block* prepended
[3]. Setting LS.CLK ("clock block allocated") in the CTB status
byte (L.STS) informs LOAd and UNLoad of its presence. When the
driver is loaded, if LS.CLK is set, LS.CBL ("clock block linked
into clock queue") is clear, and the request type in the clock
block (C.RQT) is C.SYST ("internal system call"), LOAd will
relocate the driver entry point $xxTMO (which must be global) and
insert it at C.SUB in the clock block. When the driver is
unloaded, if both LS.CLK and LS.CBL are set, and the entry point
at C.SUB is located in the driver, then UNLoad will remove the
entry from the clock queue. Thus, the driver need only concern
itself with properly setting and clearing LS.CBL in L.STS to
indicate the correct state of the clock block.

RSX-11M-Plus offers two other features that drivers may use to
remove outstanding timer requests. If a clock block is
associated with a particular controller or unit, then the driver
can remove it in the controller or unit offline routines, xxKRB
and xxUCB, respectively. For clock blocks not associated with a
particular controller or unit, the driver can provide the $xxUNL
entry point to gain control during the unloading process to
remove them from the clock queue. (I have never tried the latter
approach, however.) These routines are documented in the
RSX-11M-Plus Guide to Writing an I/O Driver [3], sections 4.5.9
(Controller Status Change Entry Point), 4.5.10 (Unit Status
Change Entry Point), and 4.3.6 (Loadable Driver Entry Points for
LOAD and UNLOAD). The UNLoad module UNLCTL contains the code for
removing the clock block at label DOCTB, which may be used as a

---

* The ClkDf$ macro defines the length of a clock block to be 9
  words. However, since the last word is not used for a
  system-type request, it is safe to allocate only 8 words here.

template for a driver routine. (Source code for MCR commands is found in UFD [12,10] on the distribution kit.)

2 RSX-11M Drivers

Unfortunately, RSX-11M has no concept of taking a controller or a unit offline, nor is a driver informed when it is about to be unloaded [2]. Consequently, I have devised a "safe" clock queue transfer block, which is similar to the interrupt control block (ICB) used with loadable drivers. When the timer expires, the executive calls this small piece of code (residing in system pool) which is guaranteed to be there even if the driver has been unloaded. The "safe" entry point examines the Device Control Block (DCB) offset D.DSP ("address of driver dispatch table") to determine if the driver is still loaded. (D.DSP is non-zero only if the driver is still in memory -- the same test is applied in DRQIO.) If D.DSP is zero, control is returned immediately to the executive. If D.DSP is non-zero, then not only is the driver resident, but the driver has already been correctly mapped by the executive clock queue service routine (in executive module TDSCH) using the C.AR5 offset in the clock block. Therefore, it is safe to transfer control to the "real" entry point in the driver.

The clock block contains the address of it's service routine in C.SUB. Normally, this is where the "real" service routine address would be placed. However, since that must point instead to the "safe" entry point, some place is needed to store the "real" entry point for access by the "safe" service routine. Conveniently, the last word in the clock block is not used for the "system call"-type clock queue entry (C.SYST). (This is also true for RSX-11M-Plus.) Thus, the "real" timer service routine address can be safely placed there. A pleasant consequence is that a single "safe" service routine can then service any number of clock blocks, since the address of the clock block is included as an argument (in R4) when the executive clock queue service routine calls the driver service routine.

It is convenient to locate this code just after the DCB for easy access by the driver. A sample clock block and setup code is given below. The first case is the simplest and applies when the driver needs only a single clock block. It is similar to the CTB clock block scheme used by RSX-11M-Plus, described above. The second case includes an additional routine that allocates and initializes a clock block from system pool.

In general, this solution will not work under RSX-11M-Plus (aha! something RSX-11M can do that RSX-11M-Plus can't do!). This is because it requires executable code in an area that is usually mapped for Data access only (assuming a separate I- and D-space executive). (That's why RSX-11M-Plus has a separate pool for interrupt control blocks -- it has the same problem dealing with the interrupt entry points to loadable drivers. The ICB pool is mapped for both Instruction and Data access, so that one part of

the executive can load an ICB -- which requires Data access --
and an interrupt process can execute it -- which requires
Instruction access.) Of course, RSX-11M-Plus doesn't need these
kludges either, as I have described above.

3 RSX-11M-Plus Examples

3.1 Static Allocation

If the driver needs a single clock block, allocate 8 words at the head of the CTB in the driver data base and notify LOAd of its presence by setting LS.CLK in L.STS. If the driver code contains the global entry point $xxTMO, LOAd will place the relocated address in C.SUB and no additional driver initialization is required.

```
        .MCall  ClkDf$

        ClkDf$                          ; Define Clock Block Offsets

;
; RSX-11M-Plus Controller Table (CTB)
;

                                ; L.CLK   8-word clock block
        .Word   0               ; C.LNK   Clock queue thread word
        .Byte   C.SYST          ; C.RQT   Request type
        .Byte   0               ; C.EFN   Event flag number
        .Word   0               ; C.TCB   TCB addr/sys subr ident
        .Word   0,0             ; C.TIM   Absolute expiration time
        .Word   $xxTMO          ; C.SUB   Subroutine address
        .Word   0               ; C.AR5   Relocation base
        .Word   0               ; C.URM   URM to execute routine
        .Word   0               ; L.ICB   Link to first ICB
        .Word   0               ; L.LNK   Link to next CTB
        .ASCII  /xx/            ; L.NAM   Device name
        .Word   $xxDCB          ; L.DCB   Pointer to DCB
        .Byte   x$$x11          ; L.NUM   Number of KRB's
        .Byte   LS.CLK          ; L.STS   Controller status
$xxCTB::.Word   xxAKRB          ; L.KRB   Table of KRB addresses
           :          :
```

3.2 Clock Queue Insertion Routine

The code fragment below issues a one-shot request with a one second expiration. It assumes the address of the clock block to be inserted in the clock queue is in R0.

```
        BITB    #LS.CBL,L.STS-L.CLK(R0) ; Block already in queue?
        BNE     100$                    ; If NE, yes
        CLR     R1                      ; High order delta time
        MOV     $TKPS,R2                ; Low order delta time
        MOV     #C.SYST,R4              ; System call, any CPU
        BISB    #LS.CBL,L.STS-L.CLK(R0) ; Mark clock block in que
        CALL    $CLINS                  ; Insert into clock queue
100$:                                   ; Reference label
```

4 RSX-11M Examples

4.1 Static Allocation

To provide an equivalent feature for an RSX-11M driver to that
described above for an RSX-11M-Plus driver,

1. Insert the following code in the data base, just after the
   DCB:

```
;
; RSX-11M Device Control Block (DCB)
;

$xxDCB::.Word    0                    ; D.LNK   Link to next DCB
        :                             :


;
; RSX-11M periodic timer clock block (accessed off the DCB)
;

        .MCall  ClkDf$

        ClkDf$                        ; Define Clock Block Offsets

D.CLK==.-$xxDCB                       ; D.CLK   Clock block
        .BlkB   C.LGTH

; Clock entry really points here, which checks to see if the
; driver is still resident (M-Plus UNLoad does this for us)
; before attempting to enter driver.

C.REAL==C.LGTH-2                      ; C.REAL "Real" entry point addr

D.CODE==.-$xxDCB                      ; D.CODE "Safe" entry point
        TST     $XUDCB+D.DSP             ; Driver resident?
        BNE     10$                     ; If NE, yes
        RETURN                          ; Save us from crashing
10$:    JMP     @C.REAL(R4)             ; Do it for real now
```

2. To initialize the clock block, use the following code
   fragment in a powerfail recovery routine that has been
   entered as a result of a LOAd command (UC.PWF must be set in
   U.CTL in the UCB):

```
        MOV     U.DCB(R5),R3          ; Get DCB address
        MOV     R3,R0                 ; Copy DCB address
        ADD     #D.CLK,R0             ; Start of clock block
        TST     C.SUB(R0)             ; Already set up?
        BNE     20$                   ; If NE, yes
        MOV     R3,C.SUB(R0)          ; Copy DCB address
        ADD     #D.CODE,C.SUB(R0)     ; "Safe" entry point addr
        MOV     #$xxTMO,C.REAL(R0)    ; "Real" entry point addr
20$:                                  ; Reference label
```

If multiple clock blocks are needed, and the number is always fixed, then more than one clock block can be allocated by surrounding the allocation with a repeat directive.

```
.Rept   N$CLKB
.BlkB   C.LGTH
.EndR   ; N$CLKB
```

Then the code above could be replaced by something like:

```
TMOTBL: .Word   xxTMO1              ; Table of timeout entry points
        .Word   xxTMO2
          :       :

        MOV     U.DCB(R5),R3        ; Get DCB address
        MOV     R3,R0              ; Copy DCB address
        ADD     #D.CLK,R0          ; Addr of first clock blk
        MOV     #N$CLKB,R1         ; Number of clock blocks
        MOV     #TMOTBL,R2         ; Address of timeout tble
10$:    TST     C.SUB(R0)          ; Already set up?
        BNE     20$                ; If NE, yes
        MOV     R3,C.SUB(R0)       ; Copy DCB address
        ADD     #D.CODE,C.SUB(R0)  ; "Safe" entry point addr
        MOV     (R2),C.REAL(R0)    ; "Real" entry point addr
20$:    ADD     #C.LGTH,R0         ; Addr of next clock blk
        TST     (R2)+              ; Addr of next timeout
        SOB     R1,10$
30$:                               ; Reference label
```

4.2 Dynamic Allocation

If the driver requires a variable number of clock blocks, they must be allocated from system pool. (This is also true for RSX-11M-Plus.) In that case,

1. The clock block allocation may be omitted from the device data base given above:

```
D.CLK==.-$xxDCB                 ; D.CLK   Clock block
       .BlkB   C.LGTH
```

2. The following routine is called to allocate and initialize each clock block. (The clock block addresses returned should be stored permanently in such a way that they can be accessed through the resident data base if the driver is reloaded. Otherwise, system pool will slowly disappear each time the driver is reloaded.)

```
;+
; ALLCLK - Allocate and Initialize Clock Block from System Pool
;
; Inputs:        R2 = Address of timer service routine
```

```
;
; Outputs:        R0 = Address of clock block (if C=0)
;                      C.SUB  = Address of clock queue transfer block
;                      C.REAL = Address of timer service routine
;                 C  = 0 Clock block successfully allocated
;                 C  = 1 Allocation failure (R0 invalid)
;
; Registers R1-R5 preserved.
;-


ALLCLK: MOV      R1,-(SP)                  ; Save R1
        MOV      #C.LGTH,R1                ; Length of clock block
        CALL     $ALOCB                    ; Allocate clock block
        BCS      20$                       ; Sorry
        MOV      CLKCOD,C.SUB(R0)          ; "Safe" transfer address
        MOV      R2,C.REAL(R0)             ; "Real" transfer address
20$:    MOV      (SP)+,R1                  ; Restore R1
        RETURN
```

3. The following code fragment should be executed by the driver
    initialization routine, prior to any calls to ALLCLK:

```
        MOV      U.DCB(R5),R0              ; Get DCB address
        ADD      #D.CODE,R0                ; Address of "safe" code
        MOV      R0,CLKCOD                 ; Save for ALLCLK
```


4.3 Clock Queue Insertion Routine

The code fragment below issues a one-shot request with a one
second expiration. It assumes the address of the clock block to
be inserted in the clock queue is in R0. It also uses the C.AR5
field as an "in-use" flag. $CLINS stores the driver APR 5
mapping there every time it is called, thereby marking the clock
block "in-use". The "real" service routine must then clear C.AR5
upon entry to mark the clock block available again.

```
        TST      C.AR5(R0)                 ; Block already in queue?
        BNE      100$                      ; If NE, yes
        CLR      R1                        ; High order delta time
        MOV      $TKPS,R2                  ; Low order delta time
        MOV      #C.SYST,R4                ; System call
        CALL     $CLINS                    ; Insert into clock queue
100$:                                      ; Reference label
```


5 RSX-11M-Plus Drivers

The same techniques described above for removing clock blocks apply here. Interrupts are always associated with a particular controller, so the controller online/offline entry point, xxKRB, is the appropriate mechanism to use.

6 RSX-11M Drivers

The same priciples described above may be applied to protect an RSX-11M system from dangling interrupts. In this case, a "safe" interrupt transfer block (ITB) is appended to the Status Control Block (SCB) for each device to be protected. When entered, each ITB immediately transfers control to an ITB service routine which examines D.DSP in the DCB to determine if the driver is still loaded. If D.DSP is zero, further device interrupts are disabled and an RTI ("return from interrupt") is executed. (The executive's ICB never sees it.) If D.DSP is non-zero, then the saved contents of the original vector are placed on the stack, and an RTI is executed to pass control to the executuve's ICB -- as though it had never been intercepted.

A 4-word ITB is required: two words for a JSR R5,xxINT instruction, and two words for the saved vector contents. A single ITB service routine can thus service any number of controllers, since the transfer through the ITB loads the address of the vector save area into R5 as part of the subroutine linkage. (The JSR R5,xxINT in the ITB is not a true subroutine call, but a convenient shorthand for "save a register, load the address of the vector save area into the register, and branch to the ITB service routine", which can then obtain all the device-dependent information it needs by accessing locations relative to the address stored in the register.) This also frees R5 for use by the ITB service routine, which must restore it before executing the RTI instruction.

7 RSX-11M Examples

To protect an RSX-11M system from interrupts to an unloaded driver,

1. Insert the following code in the data base, just after each SCB:

; SCB Extension

i].IIf NDF S.ITB, S.ITB==.-xxcSCB

```
                                        ; S.ITB  Interrupt Transfer Block
            JSR        R5,xxINT         ;;; Make R5 into arg list
                                        ;;;    ptr, BR xxINT
.IIf NDF I.VSAV, I.VSAV==.-xxcSCB-S.ITB
        .BlkW   2                       ; I.VSAV Original vector contents
```

2. Insert the following interrupt transfer block service routine
    in the data base, just after the DCB. The routine given
    below disables further interrupts by clearing the CSR --
    substitute the appropriate method for your device.

```
;
; RSX-11M Interrupt Transfer Block Service Routine
;
; Interrupts transfer here using R5 as an argument list pointer
; to a three element argument list (original vector PS and PC,
; CSR address):
;
;                     | int PS  |
;                     | int PC  |                 |JSR R5,...| S.ITB
;            SP -> | saved R5 |        R5 ->  | vec PS  | I.VSAV
;                     |             |                 | vec PC  |
;
; The DCB is checked to see if the driver is still resident (on
; M-Plus, CON OFFLINE does this for us) before attempting to
; enter driver.

xxINT:   TST        $xxDCB+D.DSP                ;;; Driver resident?
         BNE        30$                         ;;; If NE, yes
         CLR        @<S.CSR-S.ITB-I.VSAV>(R5)   ;;; Disable interrupts
         BR         40$                         ;;; Join common code

30$:     SUB        #4,SP                       ;;; Make some stack space
         MOV        4(SP),(SP)                  ;;; Move saved R5
         MOV        (R5)+,4(SP)                 ;;; Put vec PS in place
         MOV        (R5),2(SP)                  ;;; Put vec PC in place
40$:     MOV        (SP)+,R5                    ;;; Restore R5
         RTI                                    ;;; Transfer to RSX ICB
```

3. The following code fragment should be executed by the device
    initialization routine, prior to enabling device interrupts.
    It assumes the normal driver convention that R4 contains the
    SCB address. The strange order of the last three
    instructions is to prevent a device interrupt that occurs
    while the ITB is being set up from crashing the system (just
    in case).

```
; Set up interrupt transfer block that RTIs if driver not loaded

            MOV        R4,R3             ; Copy SCB address
            ADD        #S.ITB,R3         ; ITB address
            MOV        S.VCT(R4),R0      ; Get interrupt vector/4
            ASH        #2,R0             ; Vector (base) address
            MOV        R3,R1             ; Copy ITB address
            ADD        #I.VSAV,R1        ; ITB vector save area
```

```
MOV     (R0)+,(R1)+              ; Get original vector PC
MOV     (R0),(R1)               ; Get original vector PS
MOV     R3,-2(R0)               ; Replace vector PC with
                                ;    ITB address
```

References

[1] Bostwick, J., 1988, "Timer Support for User Written Drivers":
    The RSX Multi-Tasker, in DECUS U.S. Chapter SIGs Newsletters,
    v. 3, no. 8, pp. RSX-11-RSX-16.

[2] Digital Equipment Corp., 1981, RSX-11M Guide to Writing an
    I/O Driver: Order no. AA-2600E-TC, Digital Equipment Corp.,
    Maynard, MA.

[3] Digital Equipment Corp., 1982, RSX-11M-Plus Guide to Writing
    an I/O Driver: Order no. AA-H267B-TC, Digital Equipment
    Corp., Maynard, MA.

Adding Powerfail Detection to RSX Device Drivers

Lawrence M. Baker
U. S. Geological Survey
345 Middlefield Road   MS977
Menlo Park, CA   94025

On PDP-11 systems equipped with non-volatile memory (e.g., MOS
memory with battery-backup or, on older PDP-11 systems, magnetic
core memory), RSX will recover the machine state following a
power failure. In fact, it does such a fine job that normally
the only noticeable signs of an outage are an entry in the error
log file and a lapse in the system clock equal to the duration of
the outage. Both device drivers and user tasks (if requested)
are informed of the occurrence when power is restored.

RSX powerfail recovery is implemented in Executive module POWER.
Support for powerfail recovery is a SYSGEN option in RSX-11M
systems; it is always included in RSX-11M-Plus systems. In the
discussion which follows, I will assume the system contains the
necessary hardware and Executive support for powerfail recovery.

The powerfail interrupt is one of the few non-maskable
interrupts, or traps, on a PDP-11 [1]. (The powerfail interrupt
is treated as a trap in the PDP-11 Architecture Handbook since it
is generated by the CPU, even though it does not result from the
execution of a machine instruction.) This means that the

powerfail interrupt service routine inside RSX will gain control if power fails, regardless of the current processor priority.

When power fails, the powerfail interrupt service routine saves the volatile machine state in non-volatile memory and reconfigures the powerfail vector to point to the powerup interrupt service routine. When power is restored, the powerup interrupt service routine restores the machine state and increments a flag in the Executive common area ($PWRFL) to request Executive powerfail recovery. It then returns to the interrupted process after requesting a redispatching of the processor. When the Executive dispatcher is eventually entered, the Executive powerfail recovery routine is executed. (This is a distinct routine which cooperates with the powerup interrupt service routine described above to provide Executive powerfail recovery support. I apologize for any confusion my terminology may cause.)

The Executive powerfail recovery routine calls device drivers at their powerfail recovery entry point, XXPWF (where XX is the device name), for each unit with either active I/O or the "unconditional call on power failure" bit (UC.PWF) set in U.CTL. The interface to this routine is described in the appropriate RSX-11M or RSX-11M-Plus Guide to Writing an I/O Driver [2, 3]. However, what is not explained is that the powerfail recovery entry point is not called until after all outstanding driver fork requests are serviced and an interrupted Executive process completes (e.g., CALL $SWSTK in a privileged task or the Directive service routine handling an Executive request). This means that once a driver successfully dequeues an I/O packet (CALL $GTPKT) or converts itself from an interrupt process to a fork process (CALL $FORK), it will not be notified of a power failure until after it returns to RSX.

This can lead to the faulty initiation of a data transfer before the driver powerfail recovery routine has had a chance to reinitialize a device. (Even if the device does not require any reinitialization, the powerdown/powerup sequence will normally cause the device logic to reset -- clearing the contents of any registers that had previously been set up.) Imagine, for example, that a power failure occurs between the time the transfer address is loaded and the word count register is loaded. It is entirely likely that a transfer starting at location 0 will be initiated if the driver does not detect the power failure before setting the CSR GO bit. The most likely effect will be an immediate corruption of the interrupt vectors followed by a system crash or a HALT.

A driver can easily defend against this situation using a code sequence such as that given below (R2 contains the CSR address; GO, IE and RESET are typical device CSR command bits with obvious effect). It simply tests the powerfail recovery request flag within RSX to determine whether a power failure has occurred (i.e., Executive powerfail recovery is pending). If the flag is non-zero, the driver returns to RSX, deferring any cleanup to its

powerfail recovery routine. (The code is similar to that recommended in Writing a Device Driver for VAX/VMS [4].) Since the current I/O request has not completed, the device will continue to look busy, which will block the initiation of any further I/O requests.

```
        TST     $PWRFL          ; Powerfail recovery pending?
        BNE     20$             ; If NE, yes
         :       :              ; --- finish device setup ---
        BIS     #GO!IE,(R2)      ; Start the transfer
        RETURN                  ; Wait for interrupt

; Power failed -- RETURN immediately so RSX can call XXPWF

20$:    MOV     #RESET,(R2)      ; Terminate xfers in progress;
                                ;     disable device interrupts
        RETURN
```

The current I/O request must be completed in either the driver powerfail recovery routine or in the driver timeout routine. The routine which completes the I/O request must exit by restarting the driver fork process (i.e., JMP XXINI) instead of returning to RSX. Otherwise, any I/O packets in the controller request queue will sit there until the next QIO to the device.

The following code fragment could be used in a driver unit-powerfail recovery routine (R4 contains the SCB address; R5 contains the UCB address):

```
        TSTB    S.STS(R4)        ; Was the driver/device busy?
        BEQ     10$              ; If EQ, no
        MOVB    #IE.DNR&377,R0   ; Device not ready
        CALL    $IOALT           ; Finish w/ 0 bytes transferred
10$:    JMP     XXINI            ; Restart driver process
```

RSX-11M-Plus drivers are also called at the same entry point for controller-powerfail recovery, which can usually be treated as a noop. If the driver completes the outstanding I/O in the powerfail recovery routine, it must distinguish between the two forms of entry (using the C-bit) to avoid multiple completion of the current I/O request.

Completion of the current I/O request in the driver timeout routine is even simpler, since there is no need to check for current driver/device activity (without activity, there can be no timeout) and RSX presets R0 to IE.DNR. The previous five lines of code reduce to:

```
        CALL    $IOALT           ; Finish w/ 0 bytes transferred
        JMP     XXINI            ; Restart driver process
```

That's the best a driver can do, given the current powerfail
recovery support in RSX. There is still a chance of a power
failure between the test of $PWRFL and the command to start the
device (before and after the BNE instruction), but that is a much
smaller window of vulnerability than is the case if the driver
fails to detect power failures at all.

To shrink the window even smaller requires modifications to the
Executive powerfail recovery module. This is more on the side of
"blue sky," so you can quit reading now if you don't care about
things that are not currently supported by RSX.

My first suggestion approximates the combination of BIT and BNE
provided by the BBS ("Branch on Bit Set") instruction on a VAX.
It uses two words in the per-CPU database to contain the driver
address for continuation if no power failure has occurred, and
the driver address for continuation following a power failure:


```
$PWRAD::.Word   0               ; Branch target if no powerfail
        .Word   0               ; Branch target if power failure
```


Since these are shared global variables, the driver must enter a
critical section (on the correct CPU) before manipulating them.
When the device driver gains exclusive access to the processor,
it sets up the target addresses for the power failure conditions,
checks $PWRFL for a powerfail recovery already pending, and then
JMPs indirectly through $PWRAD to start the transfer, followed by
a lowering of the processor priority back to 0. This leaves a
very small window of vulnerability between the JMP @$PWRAD and
the target of the branch (one hole vs. at least two holes in the
current system). For example,


```
; Once  we have entered the critical section, all maskable inter-
; rupts are blocked and we are  guaranteed  exclusive  access  to
; $PWRAD.   Thus,  the only possible way $PWRAD can be changed is
; by a non-maskable interrupt service routine, e.g., the  powerup
; interrupt service routine.
        MFPS    -(SP)           ; Save processor priority
        MTPS    #PR7            ; Enter critical section
        MOV     #10$,$PWRAD     ;;; Setup branch targets for
        MOV     #20$,$PWRAD+2   ;;;    the powerup ISR
        TST     $PWRFL          ;;; Powerfail recovery pending?
        BNE     20$             ;;; If NE, yes -- we lose already
        :       :               ;;; --- finish device setup ---
        JMP     @$PWRAD         ;;; Final test for powerfail
10$:    BIS     #GO!IE,(R2)     ;;; Start the transfer
        MTPS    (SP)+           ;;; End of critical section
        RETURN                  ; Wait for interrupt

; Power failed -- RETURN immediately so RSX can call XXPWF
```

```
20$:     MOV       #RESET,(R2)        ;;; Terminate xfers in progress;
                                      ;;;     disable device interrupts
         MTPS      (SP)+              ;;; End of critical section
         RETURN                      ; Return to RSX
```

One line of code must be added to the powerup interrupt service routine to alter the branch target to indicate a power failure:

```
         MOV       $PWRAD+2,$PWRAD ;;; Alter branch target to
                                   ;;;     indicate power failure
```

Note that no logic is required in the Executive powerfail recovery routine to determine whether this feature is being used by the current process -- either way the code may safely be executed. It also turns out that this solution is general enough to be useful to any code that manipulates devices, whether or not it is in a driver (i.e., "connect-to-interrupt" code in a privileged task).

To completely remove the window of vulnerability requires the active participation of RSX. The powerup interrupt service routine in module POWER must be modified to replace the saved PC from the powerfail interrupt with the address of a powerfail condition handler just before the RTI instruction that returns control to the interrupted process.

```
         TST       $ONPWF             ;;; Powerfail condition handler?
         BEQ       40$                ;;; If EQ, no
         MOV       $ONPWF,(SP)        ;;; Alter return PC
40$:     RTI                          ;;; Return from powerfail
```

$ONPWF contains the address of the driver powerfail condition handler (which must be mapped by the current process), or zero, if the current (driver) process does not use one. As before, $ONPWF must be located in the per-CPU database.

```
$ONPWF::.Word   0                 ; Powerfail cond'n handler addr
```

Since $ONPWF is a shared global variable, the driver must enter a critical section (on the correct CPU) before setting up the powerfail condition handler address.

```
         MFPS      -(SP)              ; Save processor priority
         MTPS      #PR7               ; Enter critical section
         MOV       #20$,$ONPWF        ;;; Set powerfail cond'n handler
         TST       $PWRFL             ;;; Powerfail recovery pending?
         BNE       20$                ;;; If NE, yes
```

RSX/IAS-23

```
        :         :              ;;; --- finish device setup ---
        BIS     #GO!IE,(R2)      ;;; Start the transfer
        BR      30$              ;;; Exit critical section

; Power failed -- RETURN immediately so RSX can call XXPWF

20$:    MOV     #RESET,(R2)      ;;; Terminate xfers in progress;
                                 ;;;     disable device interrupts

; Exit critical section protecting powerfail condition handler

30$:    CLR     $ONPWF           ;;; No powerfail cond'n handler
        MTPS    (SP)+            ;;; End of critical section
        RETURN                   ; Wait for inter'pt/return to RSX
```

This solution completely eliminates the window of vulnerability
for detecting a power failure during device setup. The danger
with this method is that the system is more susceptible to device
driver coding errors, since the driver must correctly set and
clear the powerfail condition handler address/flag. However,
this is no different than the precautions a driver must take to
avoid double forking or multiple completions of an I/O request.
In addition, the driver must be prepared for the interruption of
powerfail condition handler itself, should a rapid power
fluctuation occur.

To reduce the risk of a driver coding error, the code to set and
clear the powerfail condition handler could be incorporated into
a macro defined in RSXMC.mac:

```
;+
; **-ONPWF$-Set/Clear Driver Powerfail Condition Handler
;
; To establish ADDR as the powerfail condition handler,
;
;       ONPWF$  ADDR               ; Set powerfail condition handler
;
; The processor priority is raised to PR7 and execution either
; continues at the following statement, or, if powerfail recovery
; is already pending, branches immediately to ADDR (BNE ADDR).
;
; In either case, the powerfail condition handler must be cleared
; before returning to RSX,
;
;       ONPWF$                     ; Clear powerfail cond'n handler
;
; The processor priority is dropped to PR0 and execution
; continues at the following statement.
;-

        .MACRO  ONPWF$  ADDR

        .IF NB  ADDR
```

```
        MFPS    -(SP)               ; Save processor priority
        MTPS    #PR7                ; Enter critical section
        MOV     #ADDR,$ONPWF        ;;; Set powerfail cond'n handler
        TST     $PWRFL              ;;; Powerfail recovery pending?
        BNE     ADDR                ;;; If NE, yes -- BR to handler

        .IFF    ; NB ADDR

        CLR     $ONPWF              ;;; No powerfail cond'n handler
        MTPS    (SP)+               ;;; End of critical section

        .ENDM   ONPWF$
```

Using this macro, the driver code simplifies to:

```
        ONPWF$  20$                 ; Set powerfail condition handler
        :       :                   ;;; --- finish device setup ---
        BIS     #GO!IE,(R2)         ;;; Start the transfer
        BR      30$                 ;;; Exit critical section

; Power failed -- RETURN immediately so RSX can call XXPWF

20$:    MOV     #RESET,(R2)         ;;; Terminate xfers in progress;
                                    ;;;     disable device interrupts

; Exit critical section protecting powerfail condition handler

30$:    ONPWF$                      ;;; Clr powerfail cond'n handler
        RETURN                      ; Wait for inter'pt/return to RSX
```

The driver must carefully manage any temporary storage on the
stack to prevent corruption of the stack by the powerfail
condition handler. This usually means that the stack pointer
cannot be altered while the powerfail condition handler is armed.

For example, the following incorrect code will corrupt the stack
if power fails between the time the device is started and the
powerfail condition handler is disarmed (i.e., the return address
is erroneously removed by the TST (SP)+ instruction):

```
        MOV     #READ,-(SP)         ; Template CSR for READ function
        MOV     (SP),(R2)           ; Prepare the device
        :       :                   ; --- finish device setup ---
        BIS     #GO!IE,(SP)         ; Enable interrupt and start xfer
        ONPWF$  20$                 ; Set powerfail condition handler
        MOV     (SP)+,(R2)          ;;; Start the transfer
        BR      30$                 ;;; Exit critical section

; Power failed -- RETURN immediately so RSX can call XXPWF
```

```
20$:    MOV     #RESET,(R2)     ;;; Terminate xfers in progress;
                                ;;;     disable device interrupts
        TST     (SP)+           ;;; Cleanup stack

; Exit critical section protecting powerfail condition handler

30$:    ONPWF$                  ;;; Clr powerfail cond'n handler
        RETURN                  ; Wait for inter'pt/return to RSX
```

The correct method for managing the stack is to leave the template CSR on the stack until the powerfail condition handler is disarmed:

```
        MOV     #READ,-(SP)     ; Template CSR for READ function
        MOV     (SP),(R2)       ; Prepare the device
        :       :               ; --- finish device setup ---
        BIS     #GO!IE,(SP)     ; Enable interrupt and start xfer
        ONPWF$  20$             ; Set powerfail condition handler
        MOV     (SP),(R2)       ;;; Start the transfer
        BR      30$             ;;; Exit critical section

; Power failed -- RETURN immediately so RSX can call XXPWF

20$:    MOV     #RESET,(R2)     ;;; Terminate xfers in progress;
                                ;;;     disable device interrupts

; Exit critical section protecting powerfail condition handler

30$:    ONPWF$                  ;;; Clr powerfail cond'n handler
        TST     (SP)+           ; Cleanup stack
        RETURN                  ; Wait for inter'pt/return to RSX
```

This device driver powerfail detection mechanism is superior to that available in either RSX or VMS. I believe it is as airtight as is possible without the addition of specialized device hardware. It is reasonably simple to implement and requires no more programming discipline than that required to write a standard device driver. I encourage your comments and critiques, either personally or through the Multi-Tasker.


Disclaimer

No warranty, expressed or implied, is made by the United States Department of the Interior, Geological Survey, as to the accuracy and functioning of the program and related program material, nor shall the fact of distribution constitute any such warranty, and no responsibility is assumed by the Geological Survey in connection therewith.

References

[1] Digital Equipment Corp., 1983, PDP-11 Architecture Handbook: Order no. EB-23657-18, Digital Equipment Corp., Maynard, MA.

[2] Digital Equipment Corp., 1981, RSX-11M Guide to Writing an I/O Driver: Order no. AA-2600E-TC, Digital Equipment Corp., Maynard, MA.

[3] Digital Equipment Corp., 1982, RSX-11M-Plus Guide to Writing an I/O Driver: Order no. AA-H267B-TC, Digital Equipment Corp., Maynard, MA.

[4] Digital Equipment Corp., 1984, Writing a Device Driver for VAX/VMS: Order no. AA-Y511B-TE, Digital Equipment Corp., Maynard, MA.

SIG RT-11
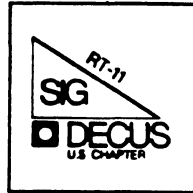
DECUS
U.S CHAPTER

RT

RT-11 MINITASKER
October, 1988

Contents

****

From the Editor:

By the time you read this, It'll be DECUS time again. To help you plan you week at Anaheim, Milton Campbell has submitted the schedule and abstracts of the sessions sponsored by the RT-11 SIG. This is our 15th anniversary. Come help us celebrate.

John Firestone at the University of Washington has created a code management program for RT-11 that works like the UNIX (harrumph!) MAKE. Several months ago I asked him to summarize some of the technical aspects of the program and how he solved some of the problems. This issue has the first installment - "What it is." The second part, "How it Works" will follow soon. (Won't it, John?)

All newsletter contributions will be gratefully accepted by:

John M. Crowell
RT-11 Newsletter Editor
Multiware, Inc.
2121-B Second St. Suite 107
Davis, CA 95616

MK: A UNIX-LIKE MAKE PROGRAM FOR RT-11

INTRODUCTION

When I work on a big programming project, I find it convenient to break my larger programs into smaller, more manageable pieces: I then don't have to page through enormous source files with my editor or recompile source code I have not changed. After breaking up a large program, though, I find it inconvenient remembering just which commands I must give to make my program agree with the changes I just made to its sources. For example, which of ten files do I need to recompile because I changed an "include" file? When I use UNIX I have an agreeable tool that takes care of this problem: MAKE and its MAKEFILEs. It seemed a pity that I did not have something similar when I used my favorite real time operating system. This inspired me to write MK.

MK'S BASIC OPERATION (IDENTICAL TO MAKE'S)

Like its long-named UNIX role model, MK lets you rebuild a program with a minimum of typing ("mk" usually suffices) and a minimum of compiling. MK follows a script, a MKFILE, that describes how a program's files are related to each other and how to rebuild them. Using this script, MK examines the "last-modified" times of the program's files by a "depth-first" search and issues the commands to rebuild those that have become out of date. To illustrate this, consider a simple example.

A program named MUNG.SAV is made by compiling three C-language files, MAIN.C, SUB1.C, SUB2.C with the commands:

```
CC MAIN.C/A
CC SUB1.C/A
CC SUB2.C/A
```

and then linking them with a library, LIB.OBJ, by the command:

```
LINK/EXE:MUNG.SAV (MAIN,SUB1,SUB2,LIB).OBJ
```

Suppose two of the files, MAIN.C and SUB1.C, pull in definitions contained in a common "include" file, DEF.H. A graph of what is built from what (their dependencies) would then look like the following:

```
                            MUNG.SAV
                               |
      --------------------------------------------------------
      |               |                 |              |
  MAIN.OBJ         SUB1.OBJ          SUB2.OBJ        LIB.OBJ
      |               |                 |
      |---------------|              SUB2.C
      |       |       |
  MAIN.C      |     SUB1.C
              |
            DEF.H
```

Notice how the common include file, DEF.H, is assigned to
the two object files, MAIN.OBJ and SUB1.OBJ, rather than the
two source files, MAIN.C and SUB1.C, that "include" it.
This actually makes sense if you ask: "If I change DEF.H,
what needs to be rebuilt?"

All this information might be represented in a MKFILE or
MAKEFILE as follows:

```
    MUNG.SAV: MAIN.OBJ SUB1.OBJ SUB2.OBJ LIB.OBJ
        LINK/EXE:MUNG.SAV (MAIN,SUB1,SUB2,LIB).OBJ
    MAIN.OBJ: MAIN.C DEF.H
        CC MAIN.C/A
    SUB1.OBJ: SUB1.C DEF.H
        CC SUB1.C/A
    SUB2.OBJ: SUB2.C
        CC SUB2.C/A
```

The first, third, fifth and seventh lines are "dependency"
lines that reproduce the graph I drew earlier; the files to
the left of the colons, the "targets", are the files to be
built, the files to the the right of the colon, their
"prerequisites", are the files the targets are built from.
Thus, the target MUNG.SAV is built from the prerequisites
MAIN.OBJ, SUB1.OBJ, SUB2.OBJ and LIB.OBJ; the target
MAIN.OBJ is built from MAIN.C and DEF.H and so forth. The
indented second, fourth, sixth and eighth lines are the
commands or "actions" that do the building. When a target
needs to be built these lines are passed to RT-11 in an
indirect command file.

If MK were supplied with this MKFILE and asked to build
MUNG.SAV it would issue whatever actions were necessary to
make each of the targets to the left of the colons newer
than its prerequisites to the right. If none of the files
to the right of the colons had changed since the last great
rebuilding, MK would announce this fact and stop. If DEF.H

had been changed, it would recompile MAIN.C and SUB1.C (but not SUB2.C) and link MUNG.SAV with the following sequence of actions:

```
CC MAIN/A
CC SUB1/A
LINK/EXE:MUNG.SAV (MAIN,SUB1,SUB2,LIB).OBJ
```

By default, MK builds the first target in the MKFILE and the prerequisites it depends on. In this example. typing "mk", would build MUNG.SAV. MK will build other targets, if you give their names when you run MK.

SOME DIFFERENCE BETWEEN MK AND MAKE

The above MKFILE is rather wordy and can be written more compactly. MAKE supports implicit rules that tell how to build one file type from another. Exploiting these, you can shorten the MAKEFILE to something like:

```
.c.obj:
    cc $</a
mung.sav: main.obj sub1.obj sub2.obj lib.obj
    link/exe:mung.sav (main,sub1,sub2,lib).obj
main.obj sub1.obj: def.h
```

or, if you are lucky, just:

```
mung.sav: main.obj sub1.obj sub2.obj lib.obj
    link/exe:mung.sav (main,sub1,sub2,lib).obj
main.obj sub1.obj: def.h
```

(The substring $< will be discussed shortly.) MAKE also allows macro definitions. To make the MAKEFILE easier to maintain, you might define a macro, OBJ, to represent its four object files and rewrite the MAKEFILE as something like:

```
OBJ = main.obj sub1.obj sub2.obj lib.obj

mung.sav: $(OBJ)
    link/exe:mung.sav $(OBJ)
main.obj sub1.obj: def.h
```

MK strives to be different. While it does support macro definitions it does not allow implicit rules. It allows something worse, as good as, or better (circle one), namely: RT-11 factoring. Using this you can write the original MKFILE as:

```
        mung.sav: (main,sub1,sub2,lib).obj
              link/exe:mung.sav (main,sub1,sub2,lib).obj
        (main,sub1,sub2).obj: $*.c
              cc $*/a
        (main,sub1).obj: def.h
```

or with a macro, SRC, as:

```
        SRC = main,sub1,sub2
        mung.sav: (SRC,lib).obj
              link/exe:mung.sav (SRC,lib).obj
        (SRC).obj: $*.c
              cc $*/a
        (main,sub1).obj: def.h
```

These last examples illustrate some differences in the way
MK and MAKE invoke macros. MK and MAKE both define a macro
in a command line with an equal signs. However, MAKE
invokes a macro whenever it finds the macro's name preceded
by a dollar sign and perhaps bracketed with parentheses
(e.g. $(OBJ) in the examples above) while MK invokes a macro
wherever and however it finds the macro's name (e.g. SRC in
the examples above). MAKE and MK both also recognize some
internal macros that are handy when you transform a bunch of
files of one type into one or more files of another:

        $@   which expands to the full name of the current
                  target
        $*   which expands to the name of the current target
                  without its extension
        $<   which expands to the complete list of
                  prerequisites for the current target
        $?   which expands to the list of prerequisites that
                  are younger than the current target

MK and MAKE, however, are quite different in the way they
expand lines containing these.

MAKE simply substitutes an appropriate list of file names
separated by spaces whenever it finds an internal macro.
For example, if it encountered the two lines:

        list: main.c sub1.c sub2.c
              print $<

it would output:

        print fee fi foo fum

This works well with MAKE's implicit rules and the command-line syntax of most UNIX programs. MK, on the other hand, lacks MAKE's implicit rules and has to deal with the different command-line style of some RT-11 programs; consequently, it behaves differently. When MK is given a dependency line with more than one target it repeats the dependency line and the action line(s) that follow, substituting each target, one at a time. For example, given the lines:

```
(main,sub1,sub2).obj: $*.c
      cc $*.c
      as $*.s
```

MK would produce:

```
main.obj: main.c
      cc main.c
      as main.s
sub1.obj: sub1.c
      cc sub1.c
      as sub1.s
sub2.obj: sub2.c
      cc sub2.c
      as sub2.s
```

This allows you to compactly specify a generic action that will convert a bunch of files of one type into another (in this example, turning C source files into object files). When one of those actions has an internal macro that expands to more than one prerequisite, MK repeats the action substituting each prerequisite, one at a time. For example, MK would expand:

```
list: (main,sub1,sub2).c
    print $<
main.sav: (main,sub1,sub2).obj
    r link
    main.sav/b:2000=//
    $<
    clib,suport//
    ^C
```

to:

```
list: (main,sub1,sub2).c
    print main.c
    print sub1.c
    print sub2.c
main.sav: (main,sub1,sub2).obj
    r link
    main.sav/b:2000=//
    main.obj
    sub1.obj
    sub2.obj
    clib,suport//
    ^C
```

This lets you to pass a long list of files to a program, in a natural, RT-11 fashion.

In other areas, MK supports some things MAKE lacks and lacks some things MAKE supports. MK allows file names in dependency lines to have colons. This means that MK requires that the colon separating the targets from their prerequisites must have a blank after it and that it will misinterpret bare device names in dependency lines. MK automatically includes a line like:

    TOUCH target

after each action it outputs for a target. This allows MK to work under operating systems, such as RT-11, that do not time-stamp their files. At present, MK does not support command-line macro definitions, double-colon dependency lines, or the fake targets .SILENT or .IGNORE.

SOME MORE LIMITATIONS OF MK

MK executes each action as an indirect command file called from an indirect control file; starting IND in an action probably will not work. Presently, MK requires that all

targets and prerequisite files reside on standard RT-11, file-structured devices. MK uses the unused, sixth RT-11 directory word to hold a file modification time; copying a file using PIP can cause MK to fail. MK has only been tested under RT-11FB V5.3.

CONCLUSION

MK has proved to be a valuable tool in its own development. It has relieved me of many minutes of repetitious typing and has often reminded me of forgotten relations between different parts of its programs. MK has probably made MK more reliable. I expect it will be useful in many other projects.

John Firestone
Geophysics Program, AK-50
University of Washington
Seattle, WA

Editor's note: John has suggested a natural second part to this article that would explain how MK works. I'll stay on his case.

RT-11 Sessions at Fall DECUS

Milton Campbell
RT-11 SIG Symposium Coordinator

To help you pre-plan your meandering through the mazes of the Fall Symposium at Anaheim, below is a schedule of the sessions sponsored by the RT-11 SIG. The abstracts are appended after the schedule.

Sessions with an asterisk for speaker indicate sessions that have been replaced due to speaker unavailability. The replacement session topics have not yet been determined.

Sessn.	Time	Title	Speaker(s)

Monday, October 17
Location: California C

RT035	9:00- 9:30	RT-11 SIG ROADMAP	John Rasted
RT034	9:30-10:00	RT-11 SIG BUSINESS MEETING	John Rasted
RT023	10:00-11:00	RT-11 PRODUCT PANEL	Connie Pawelczak
RT003	11:00-12:00	WORLD MAIL FOR RT-11	*
RT002	12:00- 1:00	REAL WORLD DISK COMPARISONS	Robert C Peckham

Tuesday, Octover 18
Location: Anaheim Room

RT005	9:00-10:00	GRAPHICS TOOLS FOR RT-11	David Evans
RT028	10:00-11:00	DECNET/RT-11 - THE USER'S VIEW	RT-11 Engineering

Location: California D

RT026	12:00- 1:00	RT-11 NEW FEATURES OF SYSLIB	RT-11 Engineering
RT037	1:00- 2:00	RT-11 MACRO-11 SYSMAC AND SYSLIB INTERACTIONS	George F. Mancini

Location: Anaheim Room

RT038	3:30- 4:00	RT-11 APPLICATIONS IN TRANS.	*
RT025	4:00- 5:00	RT-11 RUNNING ON THE KXJ TO AN IOP?	Richard Kromer & Tim McDonald

Wednesday, October 19
Location: California D

RT019	9:00-10:00 USING KERMIT WITH TSX-PLUS	Tim Clarke & John Rose
RT018	10:00-11:00 TSX-PLUS REALTIME PERFORMANCE MEASUREMENTS	Jim Crapuchettes & Tim Clarke
RT030	11:00-12:00 SHARED FORTRAN-77/RT OTS LIBRARY	Milton Campbell
RT021	12:00-1:00 TSX-PLUS MAGIC	Jim Crapuchettes & others

Thursday, October 20
Location: Anaheim Room

RT022	9:00-10:00 DECNET/RT-11	RT-11 Engineering
RT011	10:00-11:00 Why Should I Write A Handler?	William K. Walker
RT007	11:00-12:00 CONFESSIONS OF AN RT-11 HANDLERHOLIC	John M. Crowell

Location: California D

RT029	12:30- 1:30 REALTIME SYSTEM PERFORMANCE IN RT-11	RT-11 Engineering
RT032	1:30- 2:30 FORTRAN-77/RT PROGRAMMING STYLE	Robert Walraven
RT027	2:30- 3:30 EXTENDED UNIT CAPABILITY FOR RT-11	RT-11 Engineering

Location: Orange County 19

RT004	7:00- 8:00 RT-11 CONTROLLING WORLDS LARGEST COLOR DISPLAY	James Maloney
RT008	8:00- 9:00 REAL-TIME COMPUTING AT DISNEY	Nick Mansur
RT001	9:00-11:00 RT-11 15TH ANNIVERSARY AND RT-11 USERS SPEAK OUT	John M. Crowell & others

Friday, October 20
Location: California C

RT020	9:00-10:00 RT-11 APPLICATIONS WORKSHOP	Laura DeChellis & others
RT024	10:00-11:00 RT-11 FEEDBACK SESSION	RT-11 Engineering
RT036	11:00-11:30 RT-11 SIG SYMPOSIUM WRAP-UP	John Rasted

RT-11 Sessions at Fall DECUS

Milton Campbell
RT-11 SIG Symposium Coordinator

The following are the abstracts for the RT-11 SIG sponsored sessions for the Fall Symposium in Anaheim. All rooms are in the Anaheim Convention Center. Two sessions have been canceled due to the unavailability of the speakers. Replacement sessions, possibly with a different topics, will be given.

- -

RT001   RT-11 15th ANNIVERSARY & RT-11 USERS SPEAK OUT

Room: Orange County 19   Day: Thursday
Time: October 20, 1988   9:00-11:00 p.m.

Moderator: John M. Crowell
                  Multiware, Inc.

Traditionally, the RT-11 Special Interest Group conducts its "User Speakout" on Thursday evening of the Symposium. This tradition continues this year, except that the Speakout includes the culmination of the year long celebration of the Fifteenth Anniversary of the release of RT-11. Birthday party refreshments will be supplied. On hand are a number of RT-11 artifacts, as well as participants in the evolution of RT-11 from both Digital and customers.

No Speakout would be complete without the usual questions and comments from the audience. Questions about any aspect of RT-11 will generate answers. In many cases the answers are even useful. In this anniversary year, comments and statements about where RT-11 is headed are especially appropriate as RT-11 enters its mature years.

- -

RT002   REAL WORLD DISK COMPARISONS

Room: California C   Day: Monday
Time: October 17, 1988   12:00-1:00 p.m.

Speaker: Robert C Peckham   Chair: Robert Roddy
           Computer Programming   Naval Ship Research
           Services   Research Center

Many computer users are interested in the actual data transfer rates achieved when real controllers and disks operate with a real operating

system, doing real data transfers, as compared with the data transfer rates claimed in manufacturers' literature.

Test programs were written to exercise the various operational parameters of a disk, while doing the type of transfers that might be observed in real-world applications. These test programs were run on a wide variety of disks by approximately twenty DEC end user sites.

The test programs were run on disks ranging from RX01 through the more common cartridge disks, on to some relatively large and exotic Winchester and memory disks, and even on an Ethernet virtual disk.

The results are presented in tabular form so that direct comparison is possible. The results of this project are very interesting to those interested in real-world disk performance.

1987 and 1988 disk performance data is included.

- -

RT003   WORLD MAIL FOR RT-11

Room: California C   Day: Monday
Time: October 17, 1988   11:00-12:00 noon

Session Cancelled -- watch UPDATE.DAILY for replacement session.

- -

RT004   RT-11 CONTROLLING WORLDS LARGEST COLOR DISPLAY

Room: Orange County 19   Day: Thursday
Time: October 20, 1988   7:00-8:00 p.m.

Speaker: James Maloney   Chair:Milton Campbell
            Goodyear Tire and Rubber Co.   Talisman Systems

The world's largest color display is one of the most highly recognized objects. After sunset, several ships are launched with DEC equipment on board utilizing RT-11 to control the color display mounted on the port and starboard sides of the ships. The ships are actually airships and are commonly referred to as Goodyear Blimps.

This session addresses the interactive graphic hardware and software tools required to draw an animation for the very large display sign. The airborne hardware and software required to control the Super Skytacular sign in real time are also covered in this session. Future plans include a new generation of hardware and software with a fourfold increase in resolution and 127 different colors available. In addition, the next generation of equipment will provide a significant reduction of the fixed and carry-on weights. The new sign has been designated the Spectacular Skytacular sign.

RT005   GRAPHICS TOOLS FOR RT-11

Room: Anaheim Room   Day: Tuesday
Time: October 18, 1988   9:00-10:00 a.m.

Speaker: David Evans   Chair: Shal Farley
            Sandia National Laboratories   Cheshire Engineering Corp.

The choice of graphics tools for RT-11 involves both software and hardware
considerations.   The main hardware consideration is choosing between DEC REGIS or
Tektronix style terminals. This is not a difficult choice. Two classes of software are
also available: user written packages available from the DECUS library and
commercial packages. User supplied software obviously offers an excellent
price/performance ratio, but generally does not offer the many features or support
available from commercial packages. Several user written and commercial packages
are reviewed with an emphasis on features, flexibility and organizational framework
(stand alone program vs callable subroutines). Finally, options involving
DECNET/ETHERNET and other hardware enhancements are reviewed.

- -

RT006   IS YOUR REAL-TIME JOB DETACHED TO AN IOP?

Room: Anaheim Room   Day: Tuesday
Time: October 18, 1988   5:00-6:00 p.m.

Speakers: Richard Kromer   Chair:Robert Walraven
           Sandia National Laboratories   Multiware, Inc.

           Tim McDonald
           J & M Systems

Compute intensive I/O or continuous real-time tasks can be moved from a Micro-11 to
its KXJ-11 IOPs.   Sharing the processing load with the   IOPs increases the host system
capabilities.   This session covers the benefits and trade-offs of moving tasks to IOPs
and illustrates how to do it.   Specific applications including seismic event detection
and high speed barcode reading are covered.   Use of the RT-11 IOP software tool kit
and its implementation are discussed.

RT007 CONFESSIONS OF AN RT-11 HANDLERHOLIC

Room:Anaheim Room

Day: Thursday

Time: October 20, 1988

11:00-12:00 noon

Speaker: John M. Crowell
Multiware, Inc.

Chair: Robert Peckham
Computer Programming
Services

"My name is Jack, and I'm a handlerholic." I confess to writing overlaid SET code. I write handlers that refuse to install for no apparent reason. I once even wrote a handler in position-dependent code (gasp!), and linked it as a .REL file so that the load code could perform the relocation.

Along the way I've learned a few things about the "run-time" environment of RT-11 handlers (as distinguished from the I/O environment). In this session I share some of the nuggets of information I've gleaned about what goes on during handler installation, handler loading, and SET option execution. If time permits (and we hope it does not), I'll expound lightly upon the vagrancies of aborting I/O.

Some of the questions I hope to answer in this hour include:

o Do I really have to overlay large SET code?
o How do I go about it?
o Can/should SET code change the handler in memory as well as the handler file on the disk?
o How can my handler tell whether it's being installed by the bootstrap or as the result of an INSTALL command from KMON?
o Who cares?
o How can I use FETCH/LOAD code to reduce the size of my handler?

I hope that, after hearing my sad confessions, no one will stumble down the same dark paths that brought me to this place.

- -

RT008 REAL-TIME COMPUTING AT DISNEY

Room: Orange County 19

Day: Thursday

Time: October 20, 1988

8:00-9:00 p.m.

Speaker: Nick Mansur
Walt Disney Company

Chair:Milton Campbell
Talisman Systems

As part of the celebration of the Fifteenth Anniversary of RT-11, the RT-11 SIG presents this special session. Mr. Mansur discusses the use of realtime computing in the shows and attractions at the Disney amusement parks. The session concentrates on methods and techniques; no particular vendor's equipment is discussed.

- -

RT011   Why Should I Write A Handler?

Room: Anaheim Room   Day: Thursday
Time: October 20, 1988   10:00-11:00 a.m.

Speaker: William K. Walker   Chair:Laura DeChellis
          Monsanto Research Corp.   MDB Systems Inc.

While handlers have some unique advantages in many data acquisition applications, they also have their limitations. (Not the least of which being the effort necessary to learn how to write one.) Sometimes, a simple polling loop, or an in-line interrupt service routine, may be more appropriate.

This session takes a look at when you should write a handler, and when you shouldn't. It also examines some of the alternatives to handlers, and discusses when they should be used instead. The emphasis is on the "why" rather than the "how" -- a knowledge of internals and "bare-metal" programming is not necessary.

- -

RT018   TSX-PLUS REALTIME PERFORMANCE MEASUREMENTS

Room: California D   Day: Wednesday
Time: October 19, 1988   10:00-11:00 a.m.

Speakers: Jim Crapuchettes   Chair: David Billing
           Omnex Corp.   Monsanto Research Corp.

           Tim Clarke
           Omnex Corp.

This session presents and discusses performance measurements of realtime application software running under TSX-Plus.

RT019   USING KERMIT WITH TSX-PLUS

Room: California D   Day: Wednesday
Time: October 19, 1988   9:00-10:00 a.m.

Speakers: Tim Clarke   Chair:David Evans
           Omnex Corp.   Sandia National Labs.

           John Rose
           Omnex Corp.

This session is a tutorial on how to use Kermit with TSX-Plus. While the issues of modem, modem control and serial line interfaces are treated, the main emphasis is on how to set up the TSX-Plus environment.   Subjects included are:

* "CL" lines
* "DTR" control
* Use of command files to set up the TSX-Plus environment

There is a question and answer session at the end.   Users who currently have questions or problems that may require testing are invited to submit them to the speaker in advance.

- -

RT020   RT-11 APPLICATIONS WORKSHOP

Room: California C   Day: Friday
Time: October 21, 1988   9:00-10:00 a.m.

Moderator: Laura DeChellis
            MDB Systems, Inc.

The RT-11 Applications Workshop is an opportunity for Symposium   attendees to describe how they use their RT-11-based computer systems in their day-to-day jobs. The session consists of a number of 5 to 10 minute descriptions. This is an audience driven session and is the opportunity to tell the RT-11 community what you do with your system.

RT021 TSX-PLUS MAGIC

Room: California D
Time: October 19, 1988

Day: Wednesday
12:00-1:00 p.m.

Moderator: Jim Crapuchettes
Omnex Corp.

A panel of experienced TSX-Plus users, system managers and system programmers are on hand to assist users in making more effective use of their TSX-Plus systems. Some brief presentations of special techniques, utilities, handlers, command files, and programs may be made by panel members, but most of the session is oriented toward audience questions, problems, solutions, and wishlist items.

All TSX-Plus users are encouraged to attend. This is your chance to get an answer to that elusive problem, to learn how others have made their systems better, and to share the knowledge you have gained while using TSX-Plus.

- -

RT022 DECNET/RT-11

Room: Anaheim Room
Time: October 20, 1988

Day: Thursday
9:00-10:00 a.m.

Speaker: RT-11 Engineering
Digital Equipment Corp.

Chair: Dennis Jensen
AMES Labs.

This session provides an overview of the DECnet/RT-11 product. It describes product enhancements being added to the next version of DECnet/RT-11. The discussion includes DECnet/RT-11 utilities such as TLK, NFT, FAL, S, and RMT. Interfaces provided to MACRO-11 and FORTRAN-IV programmers are explained as well as the types of network programming which is available. Users are introduced to support tools available such as Network Management (NCP and NML), network generation and installation tools, and diagnostic aids.

- -

RT023 RT-11 PRODUCT PANEL

Room: California C
Time: October 17, 1988

Day: Monday
10:00-11:00 a.m.

Speaker: Connie Pawelczak
Digital Equipment Corp

Chair: Gary Sallee
Sallee Software

This session presents an overview of RT-11 Engineering, the current product releases, and the features and future directions of RT-11.

- -

RT024   RT-11 FEEDBACK SESSION

Room: California C

Time: October 21, 1988

Day: Friday

10:00-11:00 a.m.

Speaker: RT-11   Engineering
           Digital Equipment Corp.

Chair: Bradford Lubell
         L.A. Heart Lab, UCLA

In this session RT-11 Engineering reviews the customer wishlist items which accrue during the week and since the previous DECUS. Wishlist items can be deposited in the wishlist box located in the booth area next to the RT-11 Demo System or can be given to an RT-11 SIG member or a representative from RT-11 Engineering.

- -

RT025   RT-11 RUNNING ON THE KXJ

Room: Anaheim Room

Time: October 18, 1988

Day: Tuesday

4:00-5:00 p.m.

Speaker: RT-11   Engineering
           Digital Equipment Corp.

Chair: John Crowell
         Multiware, Inc.

This session discusses, in general terms, the support of KXJ11-CAs by RT-11. RT-11 will run on one or more KXJs on a single QBUS system. The RT-11 systems can be run effectively "stand-alone" on the KXJs, after "booting" from the host processor, or can be run in an "RTEM-like" (closely coupled) relationship with the host system. The types of services available among KXJs and between a KXJ and the host processor are discussed. These include file, mailbox and event flag services. Some support for peripherals local to the KXJ is discussed. These peripherals include the parallel port, the serial ports and the DMA engine. Possible applications for the KXJ11 environment are also discussed.

RT026   RT-11 NEW FEATURES OF SYSLIB

Room: California D   Day: Tuesday
Time: October 18, 1988   12:00-1:00 p.m.

Speaker: RT-11 Engineering   Chair: Jim Crapuchettes
          Digital Equipment Corp.   Omnex Corp

This session discusses new SYSLIB routines which will be added in the next major
release of RT-11. Several new SYSLIB functions simplify real-time application
development by providing often needed directory manipulation and date/time
functions. These functions allow inspection and modification of directory entries,
and perform wildcard directory searches. New and modified date routines operate
over an extended date range that takes RT-11 date capabilities through the 21st
century. Other new routines include a default trap handling function, a new CLOSE
routines, and a new virtual "no-overlay" handler.

- -

RT027   EXTENDED UNIT CAPABILITY FOR RT-11

Room: California D   Day: Thursday
Time: October 20, 1988   2:30-3:30 p.m.

Speaker: RT-11 Engineering   Chair: David Evans
          Digital Equipment Corp.   Sandia National Labs.

This session discusses the new extended unit capability which is being added in the
next major release of RT-11. Previously, device handlers could support up to a
maximum of eight device units, with each device unit representing a maximum of
65535 blocks. Today, it is possible to have a single large disk storage device that
exceeds the total storage usable with eight device units. This makes the eight device
unit limit a serious constraint. Further, having only eight logical disk (LD) units
with which to organize files hampers the user's efforts at file organization.

Topics covered in this presentation include:

      o What extended unit support is (DU, LD, and user handlers)
      o Compatibility constraints on the implementation
      o How to select extended unit support
      o How to use extended unit support
      o Ramifications of extended unit support on utilities and programmed requests
      o A look at .DRxxx (handler) macros involved
      o LOAD/.FETCH code requirements for handlers with extended unit support
      o Special considerations of $OWNER table.

RT028   DECNET/RT-11 -- THE USER'S VIEW

Room: Anaheim Room   Day: Tuesday
Time: October 18, 1988   10:00-11:00 a.m.

Speaker: RT-11   Engineering   Chair: Dennis Jensen
            Digital Equipment Corp.   AMES Labs.

This session provides a user's view of DECnet/RT-11. It covers such topics as how DECnet/RT-11 is generated and installed, how the various DECnet/RT-11 utilities are used, what the network calls look like in a user's program and why one programming interface might be chosen over another. Other relevant topics to users include the types of network management functions which are commonly performed and how to do system tuning and network trouble shooting.

- -

RT029   REALTIME SYSTEM PERFORMANCE IN RT-11

Room: California D   Day: Thursday
Time: October 20, 1988   12:30-1:30 p.m.

Speaker: RT-11   Engineering   Chair: Robert Roddy
            Digital Equipment Corp.   Naval Ship Research
                                                Center

Developing   realtime   applications   requires   an   understanding   of   a   system's responsiveness to external events. This is one of several sessions which have the combined goal of discussing strategies for configuring particular systems based on application needs. This session examines realtime performance capability using the RT-11 operating system. Performance testing has been conducted using various PDP-11 processors with and without CPU cache, using differing memory modules, and testing under all three RT-11 monitors. Various system loads have been tested using a variety of hardware and software configurations. This session presents the analysis of the testing results.

- -

RT030   SHARED FORTRAN-77/RT OTS LIBRARY

Room: California D   Day: Wednesday
Time: October 19, 1988   11:00-12:00 noon

Speaker: Milton Campbell   Chair: John Rose
            Talisman Systems   Omnex Corp.

RT-11 (and TSX-Plus) provide methods to share memory between programs. While the most common use of this capability is to share data, it can be profitably used to share code. The specific example discussed in this session is constructing the FORTRAN-77/RT Object Time System (OTS) so that it can be shared between several programs. Besides the reduction
in overall memory used, the method also results in the OTS consuming less virtual (i.e. program) memory.

- -

RT032   FORTRAN-77/RT PROGRAMMING STYLE

Room: California D   Day: Thursday
Time: October 20, 1988   1:30-2:30 p.m.

Speaker: Robert Walraven   Chair: William Walker
           Multiware, Inc.   Monsanto Research
                                                     Corp.

Examples of good FORTRAN programming style are presented. Ample time is allowed for audience participation.

- -

RT034   RT-11 SIG BUSINESS MEETING

Room: California C   Day: Monday
Time: October 17, 1988   9:30-10:00 a.m.

Moderator: John Rasted
            JTR Associates

This session begins with an overview of the RT-11 Special Interest Group (SIG), followed by SIG activity at the symposium and those areas of SIG activity which are not related to the symposium. These areas include:

   o   Minitasker (the SIG Newsletter);
   o   SIG tape copy;
   o   SIG DECUS Library activity; and
   o   Local User Groups (LUGs).

In this session, the SIG also begins the planning for the next DECUS symposium.

- -

RT035   RT-11 SIG ROADMAP

Room: California C   Day: Monday
Time: October 17, 1988   9:00-9:30 a.m.

Moderator: John Rasted
            JTR Associates

This session is designed to help the attendee obtain the most benefit from the symposium. Veteran attendees discuss the tried and true techniques that new attendees can use to make the most of the week and still survive the experience. There is a brief description of those sessions which are relevant to RT-11 users. Schedule changes and possible session repeats are also discussed. Plan to attend so you can avoid the disappointment of
missing an important session.

- -

RT036   RT-11 SIG SYMPOSIUM WRAP-UP

Room: California C   Day: Friday
Time: October 21, 1988   11:00-11:30 a.m.

Moderator: John Rasted
              JTR Associates

This is your chance to respond to the SIG and Digital presentations at the Symposium and to influence future plans. The SIG is looking for input from the attendees to aid in selecting desirable sessions for the next Symposium. At this session you have the opportunity to have questions answered that may have arisen during the Symposium. Representatives from Digital are also present.

Topics include:

        o SIG activities;
        o RT-11 and layered products;
        o   Pre-Symposia Seminars; and
        o Future DECUS Symposia.

RT037   RT-11 MACRO-11 SYSMAC AND SYSLIB INTERACTIONS

Room: California D   Day: Tuesday
Time: October 18, 1988   1:00-2:00 p.m.

Speaker: George F. Mancini   Chair: Tim Clarke
            R.T.C. Associates Inc.   Omnex Corp.

This session introduces the use of the RT-11 System Libraries, SYSMAC.SML &
SYSLIB.OBJ, and how they can be used with MACRO-11.   An example program is
presented to demonstrate console I/O, hardcopy I/O, disk file I/O, mark time routines,
time/date routines, and chaining to other routines.   Both the SYSMAC and SYSLIB
library calls are presented.

- -

RT038   RT-11 APPLICATIONS IN TRANSPORTATION

Room: Anaheim Room   Day: Tuesday
Time: October 18, 1988   3:30-4:00 p.m.

Session Cancelled -- watch UPDATE.DAILY for replacement session.

UNISIG

Editor's Corner

You are probably reading this in September (or maybe even October), but I am writing it in August. And if you've ever been in Los Angeles in August, you understand why this month's newsletter is on the "brief" side. This month we have the second in Jim Livingston's series on the Bourne Shell and we also have a bit of Unix humor from the Usenet.

I am looking forward to cooler weather in the next couple of months, so my brain can start functioning again. Until then, your comments, suggestions, and articles are encouraged. Please send hardcopy to :

<div align="center">

Sharon Gates-Fishman
NDC Systems
730 E. Cypress Ave.
Monrovia CA 91016

</div>

or e-mail to:

<div align="center">

amdahl!cit-vax!ndc!sgf

</div>
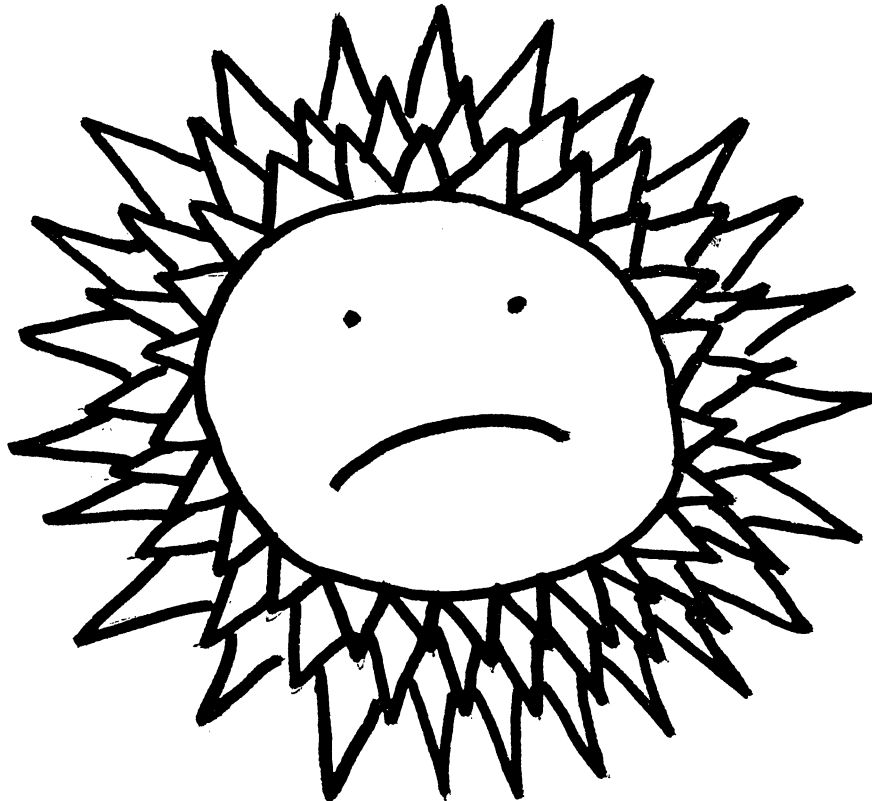
Making the Bourne Shell Your Own

by Jim Livingston

In "So There is Another ULTRIX Shell" in the last issue of the newsletter, I talked in generalities, mostly, about the differences between *sh* and *csh*, the Bourne and C shells, respectively. This month and next I'd like to get into some useful specifics about *sh*. In particular, I'll discuss how *sh* works, and how you might go about customizing your environment, assuming *sh* to be your preferred (or only available) shell.

The environment of a *sh* really has three components: its own syntax, i.e., what you can and cannot type to it; a list of things called "variables", which operate much like VMS process logical variables; and a collection of shell "scripts", or programs, the analogue of DCL command procedures in VMS.

The first thing to note is that *sh* is case-sensitive, as is the rest of UNIX. Thus, "ls" is seen as different from "LS", "Ls", and "lS". This characteristic will be one of the more noticeable changes for the VMS user, who is accustomed to having all input folded to upper case by the terminal handler.

There are a few characters that say very special things to *sh*: "<", ">", "&", ";" "[", " "]". I'll talk about each one in the paragraphs below. As a general comment, these characters are often combined on a *sh* command line to cause some change in the default behavior of the command.

I'll start with the last four characters, since two of them have familiar functions: "*" and "?" are the field and single-character metacharacters (wild cards), respectively. The brackets extend this capability somewhat, in that they enclose lists of target characters. As an example, look at

$ ls [0-9]*

which implicitly lists the digits "0" through "9" between the brackets. This command will cause all the files in the current directory having a digit as the first character of their names to be listed on the standard output, which is normally your terminal screen. We can alter the output behavior on the command line by using the redirect operator, ">". Thus,

$ ls [0-9]* > leading.digit

will cause the result of the command to be sent to the file "leading.digit". This is akin to using ASSIGN/USER LEADING.DIGIT SYS$OUTPUT before issuing a DIR command on VMS. In a similar way, "<" causes input to be taken from a specified file, rather than the keyboard. Combining two ">" symbols appends the command's output to the specified file, rather than overwriting it. Thus,

$ who >> users

will cause the output of *who* to be appended to whatever's in the file "users".

In case you want to execute more than one command on a single command line, ";" is used as a separator. It terminates the input of one command and permits you to specify another; the commands will be executed in left-to-right order. For example, the following (somewhat contrived) command line

$ ruptime > junk; grep down < junk; rm junk

would result in the display of the TCP/IP nodes that were down at the moment the command line was issued. You'll see in a little bit why I call the command contrived. As is, *ruptime* outputs its listing to the file "junk"; *grep* searches "junk" for the string "down", and prints the result on the screen; *rm* deletes the temporary file. Note that it's the guarantee of sequential execution of each part that makes this command work; if the execution order were other than left-to-right and strictly sequential, we wouldn't get the desired result.

Using another of the special characters, I'll show you how the above command would really be issued. The '|', i.e., vertical bar, is the "pipe" symbol. Since UNIX processes read from standard input and write to standard output, by default (just as VMS processes do), you can specify that the input of one command becomes the output of another, running simultaneously with it. The command above should be

$ ruptime | grep down

which causes the output of *ruptime* to become the input of *grep*, and to be searched for "down". Only lines containing "down" will be displayed, just as in the earlier command line, as *grep* writes to its standard output. This is the most efficient way to accomplish the desired result.

As I feared, I've come to the end of the column and still have special characters left; I'll finish those off next time, and push on to the shell variables and scripts.

Fun With Unix

Lately the Usenet has been host to a continuing discussion under the heading "Fun With Unix." People post their favorite unix commands with arguments that result in humorous error messages. If you've been following this on the Usenet and are sick of it, my apologies for reprinting it here. But for those of you who either missed it on the net, or aren't on the net, what follows is just a very few of the "Fun With Unix" suggestions. The response that you are *supposed* to get is shown upside-down at the bottom of this page. I have tried all of these on my Ultrix 2.2 system, and I *do* get the "correct" response.

Disclaimer: Just because it is under the cshell section doesn't mean that it won't work under bourne shell, and conversely.

csh (cshell):

1. % make love
2. % got a light?
3. % make 'heads or tails of all this'
4. % test without warning
5. % make sense
6. % make mistake
7. % \(-
8. % [Where is Jimmy Hoffa?
9. % awk "Polly, the ship is sinking"

And from the Bourne shell (sh):

10. $ "Amelia Earhart"
11. $ PATH=pretending! /usr/ucb/which sense
12. $ man -kisses dog
13. $ lost
14. $ found

The Response You *Should* Get:

1. Make: Don't know how to make love. Stop.
2. No match.
3. Make: Don't know how to make heads or tails of all this. Stop.
4. test: argument expected
5. Make: Don't know how to make sense. Stop.
6. Make: Don't know how to make mistake. Stop.
7. :-) Command not found. (if you don't get this, turn your head 90 degrees to the right and look again)
8. Missing].
9. awk: syntax error near line 1
   awk: bailing out near line
10. Amelia Earhart: not found
11. no sense in pretending!
12. dog: nothing appropriate
13. lost: not found
14. found: not found

NEWSLETTER OF THE VAX SYSTEMS SIG
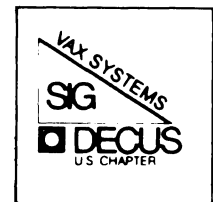
Pageswapper

Our Mascot

VAX

Contributions

Contributions and suggestions for this newsletter are constantly needed. Articles, letters, technical tips, or anything of interest to our SIG are greatly appreciated. The editor prefers submissions be made electronically. Magnetic tape and hard copy will be accepted, but may delay publication.

Please do not submit program source. It is difficult to typeset and is better distributed on the VAX SIG tape. Please do not submit "slides" from DECUS Symposia presentations or other meetings. They are generally a very incomplete treatment for those readers of the *Pageswapper* who are not so fortunate as to be able to travel to Symposia. Please DO write articles based on such slides. Please do not embed "mark up language" (TeX, SCRIBE, RUNOFF) commands in your submission. Plain ASCII text is preferred.

Send your contributions via one of the following networks:

ARPAnet/CSnet/NSFnet: ctp@cs.utexas.edu
UUCP: ctp@cs.utexas.edu.uucp ({harvard,ihnp4,uunet}!cs.utexas.edu!ctp)
BITNET: CTP@UTADNX
CompuServe: 75226,3135
DECUServe/DCS: POOLE

or you can use a facsimile machine connected to the following number:

(512) 471-8885

or, if you must, use the U. S. Mails:

Clyde T. Poole
The University of Texas at Austin
Department of Computer Sciences
Taylor Hall 2.124
Austin, Texas 78712-1188

Weekday Relative Time

Bob De Wolf, Fullerton, CA

Often the need arises to schedule a batch job at a particular time on a particular day of the week. It's easy to do this with VMS because you can use weekday relative time format to reschedule the procedure every time it runs. Suppose you want to schedule the job to run on Sundays at noon. Just put a SUBMIT command at the beginning of the batch procedure and add an appropriate "/AFTER" qualifier such as:

```
/AFTER="SUNDAY+12:00"
```

With this qualifier, the job will be scheduled to run at noon next Sunday, regardless when the job is currently running. If you add the following statement to the job after the SUBMIT command the job can be properly scheduled the first time by simply running the procedure:

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

WHOOPS!

Forget it. None of the above will work until VMS supports weekday relative time. The relative time feature that exists in VMS does this job clumsily at best. If you use it you must use an /AFTER qualifier such as:

/AFTER=TODAY+7-12:00

The "-" character is read "dash" and is a necessary separator. This reschedules the job to run as desired, provided that it is already running on the correct day of the week. Naturally it won't be if your system was down on Sunday and you're booting it on Monday. It also can't be used to initially schedule the job, unless you happen to be starting it on Sunday which is unlikely.

There must be a better way, and there is. You can't use weekday relative time as described at the beginning of the article, but you can use the NEXT procedure listed at the end of this article to obtain the date and time for the next run using weekday relative time. It returns the date and time in the global symbol "NEXT_WEEK". In this case, you procedure should look like this:

```
$ @NEXT SUNDAY 12:00
$ SUBMIT procedure-file-spec/AFTER="''NEXT_WEEK'"
$ DELETE/SYMBOL/GLOBAL NEXT_WEEK
$ IF F$MODE() .EQS. "INTERACTIVE" THEN EXIT
```

This isn't as pretty as as it would be with weekday relative time format built into VMS, but for now it'll have to do.

```
$ DAYS_OF_THE_WEEK="MONTUEWEDTHUFRISATSUN"
$ REQUEST_DAY_NUM=-
F$LOCATE(F$EXTRACT(0,3,P1),DAYS_OF_THE_WEEK)/3 + 1
$ DATE=F$TIME()
$ WEEKDAY=-
F$EDIT(F$EXTRACT(0,3,F$CVTIME(DATE,,"WEEKDAY")),"UPCASE")
$ TODAYS_DAY_NUM=-
F$LOCATE(WEEKDAY,DAYS_OF_THE_WEEK)/3 + 1
$ DAYS_FROM_TODAY=REQUEST_DAY_NUM - TODAYS_DAY_NUM
$ IF DAYS_FROM_TODAY .LT. 0 THEN -
DAYS_FROM_TODAY=DAYS_FROM_TODAY + 7
$ IF DAYS_FROM_TODAY .EQ. 0 THEN GOTO THIS_DAY
$ NEXT_WEEK==-
"TODAY+''DAYS_FROM_TODAY'-''F$CVTIME(P2,,"TIME")'"
$ EXIT
$ !
$ THIS_DAY:
$ IF F$CVTIME(DATE,,"TIME") .LES. F$CVTIME(P2,,"TIME") -
THEN GOTO DOIT_TODAY
$ NEXT_WEEK=="TODAY+7-''F$CVTIME(P2,,"TIME")'"
$ EXIT
$ !
$ DOIT_TODAY:
$ NEXT_WEEK=="TODAY+''F$CVTIME(P2,,"TIME")'"
$ EXIT
```

A Few Modest Proposals (with apologies to Jonathan Swift)

Glenn Everhart

One of the things working groups do well in DECUS is to direct energy to solving problems. To get people in the right frame of mind, I'd like to suggest a few problems that need solving. If anyone is interested, I suggest coming to the working group formation Open Meeting which will be scheduled in Anaheim, and/or considering some of these within the existing working groups. It would be useful to many of us if any or all of these problems could be solved by a few people cooperating to get it done.

1. Wide area DECnet doesn't give fine enough granularity of user classes.

This is something people in DECUS can solve for themselves. There is no need to wait (possibly a LONG time) for DEC to do this one.

Problem: "WORLD" in the context of a large DECnet is too broad. On a single machine, files available to the world are available to people on YOUR machine (and you can use ACLs to limit which of those can get at them). On a large DECnet, if FAL is left open, files open to world are open to EVERYBODY. I'd like to be able to have files open to large classes of people, say company employees, but not generally available to others, and I'd like that to be possible network wide. The problem is, DECnet doesn't propagate identifiers. From what I've heard thus far, Phase V won't either (though this may - and I hope will - change). Proxies to thousands of users in each class aren't an answer; too hard to maintain.

However, the following scenario might provide a reasonable level of functionality.

A. Suppose we have every machine on a network agree on a FEW standard identifiers. Examples: Non_Citizen, Non-Employee, Short_Timer, Twit.

B. Now, on each node, accounts have these identifiers applied where appropriate. (Most accounts would need none). Notice these identifiers are suitable for EXCLUDING access; ability to give yourself identifiers gains nothing.)

C. In SYLOGIN.COM, FAL.COM and similar places, before anything useful can be done, run an image which must be concocted. This image first obtains the originating process' PID and the node name from which the request comes. [If the node is in a local list of "untrusted nodes", it just flags all "standard identifiers" as present.] Now the image accesses (via nontransparent DECnet) a special object on the originating node, and sends the originating process information it has obtained back to this server. [If the server is unavailable, all "standard identifiers" are flagged as present.] The server obtains (via CMKRNL ?) information on which "standard identifiers" are present for the process being queried about. It sends a message back to the image that asked it telling which of the "standard identifiers" is present. NO OTHER identifiers are dealt with, to avoid general screwups.

Now the image we added to SYLOGIN.COM, or FAL.COM or wherever forces those identifiers into the process it's running in. CMKRNL may be needed here also.

As a result, the process that came from remote file access, or SET HOST, has in effect inherited the identifiers from the source process. These standard identifiers can now be used to set up ACLs to refine/restrict "world" file access. Because of the "untrusted node" list, a site

can control further which other sites it trusts, so that sites where system management is weak or doesn't put identifiers on accounts, gets treated as totally untrusted.

This looks to me like something that's workable. Has anybody done all or part of it already?

In support of this, I received the following message from Matthew Hull:

```
From:        Matthew Hull <HULL%CTSTATEU.BITNET@MITVMA.MIT.EDU>
Subject:     Getting Process IDs
To:          EVERHART%ARISIA.decnet@GE-CRD.ARPA

>  The server obtains (via CMKRNL ?) information on which "standard
>identifiers" are present for the process being queried about. It sends a
>  [...]
>This looks to me like something that's workable. Anybody done it already?
>Or part of it?
>     Code donations gratefully accepted!

Part of it.  We have several situations where Identifiers are added to
a process but not via the UAF. This would not cause any problems if the
system service Sys$Find_Held worked as we think it should; that is, return
all identifiers in the current process, and NOT just the ones in the UAF.
Towards this problem I wrote a Macro routine which was written to be
implemented as a Site System Service routine.  In the end, it was decided
that our applications did not warrant a Site-specific System Service, and the
routine was shelved.  However, I'd be pleased if someone could benefit
from my efforts, if not ourselves.  The routine accepts an array of longwords
specified by the caller, and fills this array with all the current process'
identifiers.  It enters Kernel mode, copies the critical memory, and exits.
The routine was written for VMS V4.5, and was researched by using
ANALYZE/SYSTEM to examine the process memory to determine the structure of
process identifiers in VMS. Since it makes assumptions on a VMS critical
structure (the process Rights list), some changes may be needed for different
versions of VMS 4, and I wouldn't expect it to work for VMS 5 (without some
modification, that is).  Nevertheless, if you decide to write your own
system to handle your Identifiers/Access application, this code may be of
some use.

        .PAGE
        .SBTTL  Return Process Identifiers
        .LIBRARY   "SYS$LIBRARY:LIB.MLB"
;++
; Functional Description:
;       This routine returns the longword values of the current process'
;       Identifiers into a longword array.  Written for VMS V4.5 at
;       the Connecticut State University by Matthew G. Hull.
;
; Input Parameters:
;       04(AP) - Number of longwords in longword array - Longword (Byte) value
;       08(AP) - Address of longword array              - Longword address
;       R4     - PCB address of current process
;
; Output Parameters:
;       R0     - Completion Status code
;       @08(AP) - Written with Process Identifiers
;
```

```
; Register Use:
;        R4 - PCB address of current process
;        R5 - Temp; Addr of next Rights List descriptor
;        R6 - Address of current ID list
;        R7 - Counter of current ID list
;        R8 - Current pointer into user longword array
;        R9 - Count of User Space IDs stored
;        R10- Address just beyond the 4 Rights List Descr. Addresses
;
; A picture is worth a thousand words:
;        The 4 Rights List pointers each point to a Rights List descriptor.
;        The Descriptors describe a block of Identifiers.
;        An identifier is a longword value, followed by an attributes longword.
;
;        |--------------|          |-------------|
;        | RL Pointer #1| ---->     | Descrip #1  |
;        |--------------|          |-- (Size) ---|
;                                  |  (Pointer)  | ----> |------------------------|
;                                  |-------------|       |ID array #1|            |
;                                                        |-----------|------------|
;                                                        | (ID val)  | (Attrib.)  |
;                                                        |     .     |     .      |
;                                                        |     .     |     .      |
;                                                        |     .     |     .      |
;---
         $PCBDEF                ; Define Process Control Block offsets
         $ARBDEF                ; Define Access Rights Block offsets
         $SSDEF                 ; Define System Services constants
         .ENTRY   USS_GET_IDS,^M<R4,R5,R6,R7,R8,R9,R10>
         CMPL     #2, (AP)                   ; Check number of arguments
         BNEQ     40$                        ; Error if not 2
         MOVL     8(AP), R8                  ; Get address to store Identifiers
         BEQL     50$                        ; Branch if null address
         MOVZBL   4(AP), R9                  ; Load User longword counter
         MULL3    #4, R9, R5                 ; Determine # bytes to be written
         PROBEW   #0, R5, (R8)               ; Check if array is writeable
         BEQL     50$                        ; Error if not
         MOVL     PCB$L_ARB(R4), R5          ; Base Address of Access Rights Block
         ADDL2    #ARB$L_RIGHTSLIST, R5      ; Plus offset to 1st Rights List Descrip
         ADDL3    #16, R5, R10               ; Plus 4 longwords
         BRB      10$                        ; Branch beyond Exits
40$:     MOVL     #SS$_INSFARG, R0           ; Indicate insufficient arguments
         RET                                 ;    and return
50$:     MOVL     #SS$_ACCVIO, R0            ; Indicate access violation
         RET                                 ;    and return
60$:     MOVL     #SS$_NORMAL, R0            ; Set normal completion status
         RET                                 ;    and return
10$:     MOVL     (R5)+, R6                  ; Load addr of next Rights List descrip
         BEQL     30$                        ; If empty, try next Rights List
         MOVZWL   (R6), R7                   ; Load Identifier bytes counter
         ASHL     #-3, R7, R7                ; Divide by 8 (8 bytes per ID)
         MOVL     4(R6), R6                  ; Load Address of ID List quadwords
20$:     TSTL     (R6)                       ; See if this ID is blank (zero)
         BEQL     30$                        ; If so, move to next list
         DECL     R9                         ; Decrement User Longword counter
```
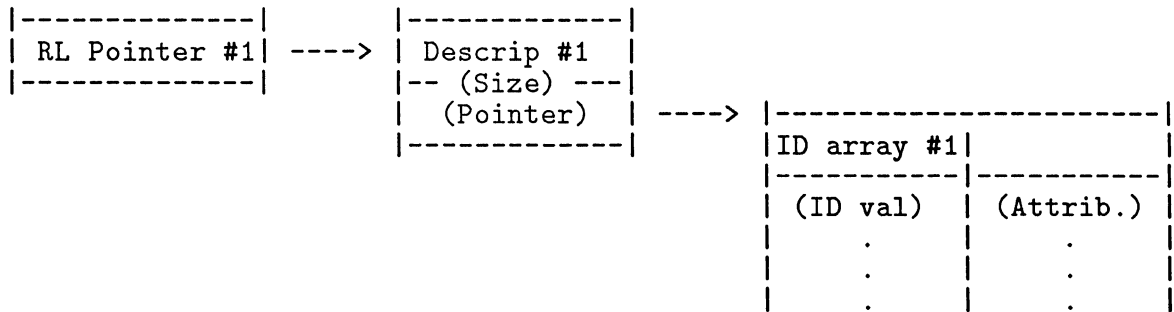
```
        BLSS    60$                     ; Exit if no more memory
        MOVL    (R6)+, (R8)+            ; Copy Identifier into user space
        ADDL2   #4, R6                  ; Skip attribute flags longword
        SOBGTR  R7, 20$                 ; Decrement ID count, branch if more
30$:    CMPL    R5, R10                 ; Are we at the last Descr address?
        BLSS    10$                     ; If not, process next Rights list
        BRB     60$                     ; Exit normally
        .END
; =====================================================================
; =====================================================================
;
;  Test program to simulate the process memory area in order to test
;    the USS_Get_Ids routine.
;
;
; Simulated Process space
;
        .PSECT PROCESS_SPACE  RD,NOWRT,NOEXE
PCB:    .BLKB 140                       ; 140 bytes of process info
        .ADDRESS ARB                    ; Base addr of ARB
ARB:    .BLKB 32                        ; 32 bytes of ARB info
RLDP1:  .ADDRESS RLD1                   ; Pointer to 1st Rights List descriptor
RLDP2:  .ADDRESS RLD2                   ; Pointer to 2nd Rights List descriptor
RLDP3:  .ADDRESS RLD3                   ; Pointer to 3rd Rights List descriptor
RLDP4:  .LONG    0                      ; Rights list 4 is empty
;       .ADDRESS RLD4
        .BLKB 40                        ; More ARB
        .BLKB 100                       ; More PCB
                                        ; Descriptors
RLD1:   .LONG ^X40                      ; 64 bytes of memory to a Rights List
        .ADDRESS RL1                    ; Address of 1st Rights list
RLD2:   .LONG ^X40
        .ADDRESS RL2                    ; Address of 2nd Rights list
RLD3:   .LONG ^X40
        .ADDRESS RL3                    ; Address of 3rd Rights list
RLD4:   .LONG ^X40
        .ADDRESS RL4                    ; Address of 4th Rights list
                                        ; Rights Lists
RL1:    .QUAD ^X00030023                ; [3,43]
        .QUAD ^X80010052                ; DIALOUT_ACCESS
        .QUAD ^X80010002                ; STUDENT_HELP
        .QUAD ^X80172172                ; PPS_ACCESS
        .BLKQ 4                         ; The rest are zero
RL2:    .QUAD ^X80000004                ; LOCAL
        .QUAD ^X80000003                ; INTERACTIVE
        .BLKQ 6                         ; The rest zero
RL3:    .QUAD ^X8001004D                ; SYS$NODE_CCSUB
        .QUAD ^X8001004C                ; SAS_ACCESS
        .BLKQ 6                         ; The rest zero
RL4:    .BLKQ 8                         ; Empty
```

```
;
; Calling Program Data
;
          .PSECT MY_DATA   RD,WRT,NOEXE
IDENTS: .BLKL 20                    ; Max supported = 32 IDs (VMS V4.6)
                                    ; Output Longword Array (will receive IDs)
ARR_SIZE:
          .LONG 20                  ; Size of Output array
          .LONG   0                 ; Buffer
          .PSECT MY_CODE RD,NOWRT,EXE
          BEGIN TEST_GET_IDS
          MOVAL   PCB, R4           ; Addr of PCB into R4
          PUSHAL  IDENTS            ; Push address of Longword array
          PUSHL   ARR_SIZE          ; Push number of longwords in array
          CALLS   #2, USS_GET_IDS ; Get the Identifiers in "Process Space"
          MOVL    R0, R8
          $IF R8, NEQ, #SS$_NORMAL, L -
            THEN <PRINTBOMB ^| Error from USS_GET_IDS.|, R8>
          MOVAL   IDENTS, R6
DUMP:     $IF (R6) EQL #0 L THEN <BRW DONE> ELSE <DUMPMEM (R6)+, #4>
          BRB DUMP
DONE:
          EXIT
          .END TEST_GET_IDS
```

2. VD: for VMS V5 (and other similar jobs):

The VMS virtual disk driver VD: allows a contiguous file to be treated as a VMS volume. This has many uses (multiple file structures, volume ACLs for file protections, multiple cluster factors, and so on) but the driver needs to be converted to run under VMS V5.x. There are many other highly useful pieces of software that do privileged things, but don't run under VMS V5. Remember P (the process internals inspector from VMS V3)? How about OBSERVE? The list is long, but many of these useful utilities should be converted to run under the new VMS release to assure their continued usefulness to the VAX community.

3. Ethernet security designs.

Ethernet systems are becoming widely used as paths between VAXen and terminals these days. Unfortunately, traffic on Ethernet is easy to tap anywhere one can put an Ethernet adapter into "promiscuous mode". Such programs are used for diagnostics (find out why your Ethernet is so slow today...) but can easily be used to watch for passwords and the like. VAX based Ethernet monitor programs have appeared on various national networks, and PC based programs are supplied with some board makers' products. They are surprisingly small and simple, and appear easily modified to capture selected traffic. (They run a few hundred lines of C, with modification perhaps under 50 lines of C to select, say, LAT traffic going TO a VAX, which gets what people type in.) As a result, Ethernet terminals need to be thought of as inherently tapped, with the sole consolation being that the tap goes no further than one's building walls most of the time. This is an unhealthy situation, and needs to be fixed. Fixing it, however, requires some changes only DEC is in a position to make. A working group could be helpful in this by thrashing out a workable design to help speed up the engineering effort at correcting the problems.

One possible way to address the problem would be to add a few passes of XOR-with-keystring encryption to traffic between the LAT boxes and the VAX. By arranging modest keys, relatively prime string sizes, and changing the keys frequently (and transparently to users), a fairly efficient and quick encoding could be devised where the key would always be essentially longer than the messages, making decipherment harder. It would also mean that a much longer

session watching net traffic would be needed to even hope to capture sensitive information. If such an approach is decided upon, many questions remain. Should traffic from terminal to VAX only be encrypted, or both ways? Should a non-translated "transparent" mode be added for file transfers? How should keys be managed? Should this be one more SHOW TERMINAL option? If there are any REALLY high powered code hackers interested in the problem, perhaps they could devise patches to DEC code for VMS and for the LATs to accomplish this more quickly.

The result of a working group on this issue would be an article to be presented to Digital as a suggestion, and to the VAX SIG via some channel such as publication in Pageswapper for discussion. The partial approach outlined above is for illustration. An actual working group panel should approach the issue freshly.

4. A Concordance for the VAX SIG tapes.

How many of you noticed that a Concordance building tool was included on the Spring 1988 VAX tapes? A few people willing to work could perform a useful service to the SIG by building a concordance of the VAX SIG tapes, or as many of them as can be readily acquired. Digging deeper into the tapes than just the abstracts, to find what is and is not of special interest, is desirable. I'd like to work with such a group myself, but don't have time to do it all.

The above four topics will illustrate a few of the problems facing VMS users which we can help one another with. I hope there will be many more such topics raised. I also hope some of these topics will inspire some of the readers to solve some of the problems which are posed.

VMS V5.0 Upgrade: "Gotcha" for NON-cluster Systems

Richard D. Piccard, Kalamazoo College, Kalamazoo, MI

ABSTRACT:

The VMS V4.7 to V5.0 upgrade process creates the on-disk directory structure of a cluster common system disk, even if you are NOT in a cluster! This fact may not be noticed in advance by the system manager, and the resulting structure is not documented where a non-cluster system manager is likely to browse. I discuss briefly some of the consequences of this restructuring of the system files, and the ways to deal with it.

INTRODUCTION:

The VMS V4.7 to V5.0 upgrade process creates a uniform system file directory structure for all sites that are not heterogeneous clusters. Stand-alone systems will thus find it simpler to upgrade to local area-, CI-, or mixed-clusters at any future time. The Colorado Customer Support Center will also find it easier to provide support to all customers, since we will all share the same file directory structure, aliases, and logical name environment.

Unfortunately, the documentation for the upgrade process does not adequately warn the system manager of this change and its consequences (I, at least, did not notice it in advance). Furthermore, the resulting structure is not documented in any of the manuals a non-cluster system manager is likely to browse in. The people in the Colorado CSC quickly directed me to the "VAXcluster Manual", section 2.1, which does indeed present the essential information.

Sharing Files:

The key idea is that SYS$SYSROOT is a logical name search list with two translations: SYS$SPECIFIC and SYS$COMMON. The files that are intrinsically specific to a single processor are located in the subdirectories within SYS$SPECIFIC, (such are [SYSMGR], [SYSEXE], etc.). The files that can be shared amongst the several systems in a cluster are located in the corresponding directories within SYS$COMMON. The search list for SYS$SYSROOT is set up to look first at SYS$SPECIFIC, and then in SYS$COMMON.

The first point of confusion for non-cluster system managers is that if you have SET DEFAULT to SYS$MANAGER:, as will be the case upon login as SYSTEM, and give the DIRECTORY command, VMS will talk to you about the files it finds in SYS$SYSROOT:[SYSMGR] and SYS$COMMON:[SYSMGR], with no indication that the former is really SYS$SPECIFIC. This enhances the likely confusion, because, as we saw, SYS$SYSROOT is really a logical name search list, one of whose translations is SYS$COMMON.

Upgrade Relics:

Just to keep us all on our toes, the upgrade procedure itself dumps nearly all of the system files into SYS$COMMON, including files that clearly do not belong there, but rather in SYS$SPECIFIC. The first one we stumbled across was SYSTARTUP_V5.COM, which we had prepared in advance on a scratch pack while we were experimentally running V5.0 in the dead of night. We managed to end up with copies in both SYS$SPECIFIC and SYS$COMMON, and made some modifications to one and some to another. Such joy!

At the end of our regular accounting period, some time after the upgrade, we hit another beaut: SYS$MANAGER:ACCOUNTNG.DAT is left in SYS$COMMON after the upgrade, but when you SET ACCOUNTING/NEW_FILE, it creates the replacement in SYS$SPECIFIC, and there is NO SUCH FILE as SYS$MANAGER:ACCOUNTNG.DAT;-1 for your data harvesting command procedure to preserve and use!

ACL Propagation:

In order to provide a secure system environment while using student operators, (see "System Security and Student Operators DO Mix." John D. Hoinville and Richard D. Piccard; Pageswapper for August, 1987, pages VAX-3 through VAX-18), we perform all routine BACKUP operations from batch jobs, whose log files have Access Control Lists that permit only the appropriate inspections by the student operators (to determine softwrite error counts, and to determine what other errors, if any, have occurred). All of these log files were left in SYS$COMMON by the upgrade procedure, but the new versions created by running those procedures since then are of course in SYS$SPECIFIC, since it finds that directory first! The access control lists, unfortunately, only propagate within a directory, so the student operators could not inspect the new log files.

Conclusion:

I am sure that we will discover other instances of file misplacement, and it is my understanding that there will be a DSIN article providing further advice as to which files ought to be located in SYS$SPECIFIC, and which in SYS$COMMON, but it seems to me that a few prompt words on this subject in the Pageswapper might be helpful to others, so I will leave it at this.

VAX Systems SIG Advanced Software Q & A

1988 DECUS U.S. Spring Symposium - Cincinnati, Ohio

Sue Reese, Lockheed Missiles and Space Co., Sunnyvale, CA, Session Chairperson

[Editor's Note: The following is an edited transcription of the VAX Systems SIG Advanced Q&A. The transcription was performed by a professional service that does not employ computer experts. The original unedited transcription was amusing to read and completely useless. I have spent a great deal of time attempting to correct as many of the errors and omissions as I could identify. Many errors still exist. Notes form the editor are enclosed in square brackets. Questions are preceded by a "Q", answers are preceded by and "A" and followups from the audience are preceded by "F". Some undecipherable information is indicated by ????.]

Sue Reese – We won't go into all of the reasons we changed the format. We will just describe the format. Throughout this week forms have been available in the camp ground and in the opening session hand outs, the "Boot Block" and people have, throughout the week, been filling out questions for this session. This is the first time we've tried this and we will be having members of the executive committee and I think a helper or two reading out the question and then we will get an answer. Our every intent is that over the next few months these questions and answers will be printed in the *Pageswapper*. Some of you may not be aware of the fact that the lead time for the *Pageswapper* is two months. So, it will be about two months down the road before they appear, but we will begin printing these in the *Pageswapper*. We're going to begin the session by introducing the DEC people that we have on the stage and before we start I'd like everyone to give a real big round of applause to the DEC people, they've been here all week, they've been down on the exhibit floor, they've been in the camp ground, they've been in the suite, they've been giving sessions, they are just as tired as we are and we want to let them know how much we appreciate their coming. [applause] I'd like to introduce Harriet Cohen, Harriet is our counterpart from the SIG.

Harriet Cohen – I'd like to thank you for coming to the session late in the day, I hope we can answer all your questions. We tried very hard during the week to address the questions and hear your concerns and take back your input for future releases of VMS. What I'd like to do now is to thank Sue Reese and the other members of the Steering Committee, they've all worked very hard. They've put a great deal of effort into it and for us too it's been a very successful DECUS and in large part thanks to Sue and the rest of the executive council of the SIG. [applause]

Sue Resse – One of the things that makes this work for you and for us is the cooperation and the spirit that exist between the members of the Steering Committee and DEC and thank you very much, this means you like us... [laughter]... We're going to have each of the developers up here identify themselves and perhaps their area of expertise and why don't we just start down at the end....

> I'm Stu Davidson and I work on VAX RMS.
>
> I'm Paul Beck, DECnet VAX.
>
> Ralph Webber, File Systems, some other random stuff.
>
> Stu Farnum, General Exec and Multi-processing.
>
> Dave Tiel, VAX Clusters.

Jake Benoit, Mice and menus.

Tom Cafferello, VMS Performance.

Lee Lahey, SET HOST.

Sharon Armode, DECnet VAX.

Michelle ????, the VAX clusters group and tapes specifically.

Steve Furro, UMS Exec.

Ann ????, DCL and file manipulation utilities.

Ray Gosselin, Batch Print.

Bill Fisher, Batch Print.

Sue Reese – Okay, we're going to launch into this. A member of the executive committee will read the question. We will read the name of the person asking the question, if that person is in the room, we're not going to delay a lot, but if you're in the room go towards the microphone in the middle here in case there is an additional piece of information that's necessary [or] we didn't quite understand what you were saying, if there's no one there we'll just answer as best we can and move on.

Jeff Jalbot – Just let me explained briefly the format of these questions. We've sorted these into topic areas and so that we don't get instant burn out from somebody with a particular specialty, we're going to rotate through the topic areas, so the first topic area is A and we're going to go B, C, D and these have intrinsic meanings.

Q. This first questions is from David Lobers from Real Share and he asks....On a LAVC, I set the time ahead on one node and job execution queues on all nodes fired up, what's going on?

A. Well, batch print is a cluster wide utility and when any job controller thinks it's time to execute queued batch jobs it will execute all of the jobs in the timer list. It will execute all the jobs that are ready to be executed. Basically it's a feature. The thing you want to remember here is to keep your cluster time consistent. There's a SET TIME CLUSTER command in 4.6 and 4.7 and SYSMAN V5 will provide the capability for keeping the cluster time consistent.

Q. Okay, this is category B, question number 1 and the question comes from Bill or Ed Davidson at Texas Instrument and it deals with 750 and MicroVAX 2000 and an LAVC. The question is, when a 750 goes through a power fail restart the MicroVAX 2000 satellite member of the LAVC crashes, reboots, crashes, reboots and so on until the 750 is rebooted. As rumor has it is the PE driver does not support power failure recovery until VMS 5.0. Is the crashing problem fixed in VMS 5.0?

A. PE driver, that's the driver that's used to support cluster communications over the Ethernet does not support power fail in releases before 5.0. It does support power fail in 5.0.

Q. This is category C, question 1. The question was submitted by Joseph Crumb an Independent Consultant. The CPUs were the VAX 8700 and 8550 under VMS Version 4.5 and this is a rather long question so bear with me. For months one of my clients experienced performance degradations that seemed to go away after booting and come back after a day or so of operation. After reading an article on page file fragmentation, I suspected this might be the problem. I copied the program from the article and it showed that after

several hours of operation the largest available segment of the page file on one system was about twenty five pages and the allocation factor was around 16 or so. The system used a lot of global buffering and did transaction processing and COBOL with FMS, about a 100 users for one application. The page file was about 40% full at this point and no "page file fragmentations continuing" message was ever received on the console. Performance and fragmentation improved significantly when the system was booted for a period of about eight hours. Increasing the size of the page file to a point where it was only about 10 to 15% full seems to have helped, but needs too much disk space. Is this fixed in Version 5 and is there anything else I can look at or do until then?

A. Okay, this doesn't sound to us like something that really is a amenable to fixing by VMS. It's our guess that you're seeing a set of application characteristics which simply run counter to some of the page file allocation policies that VMS enforces. We suspect the application needs lots and lots permanently allocated pages scattered through the page file. One way this could happen would be a processes ran in image forever and have lots of data that was modified once, for example, by initialization and then use read only thereafter, although there are probably other scenarios that would cause the same behavior. Transaction processing certainly seems like a work load that would have those characteristics. The solution that we see is effectively the one that you don't like and that's the create a larger page file to reduce the fragmentation. For most people's work loads a page file which is only 40% full is going to be just fine, but that obviously does not seem to be the case here. There's a further point which is that if you use a lot of global buffers, RMS global buffers there's a short coming in V4 in the page file space for global buffers is limited to the primary page file. In V5 the page file space for global buffers can be allocated on a secondary page file. So, when you see V5 you'll see some relief to your disk space problem in that you can move the majority of your paging space off the system disk and that way you won't feel the severe crunch of having a considerably larger than average page file to support your application.

Q. Okay, the next question is a category D question number 1. The question is from Dale Whygan at NRC. What is the difference between full function and end node DECnet when terminals are directly connected. There is no terminal server.

A. DECnet functions and how terminals are connected aren't related at all and there seems to be just some misunderstanding in this question relating terminal servers and the protocols they use and DECnet routing and the protocols it uses. End node and full functional licenses deal with the DECnet routing layer where an end node is as node is going to have exactly one circuit active and a router is a node which can have more than one circuit active routing DECnet traffic through it, whereas terminal servers employ a totally different protocol which can coexist with DECnet but does not go through DECnet routing. So, how terminals are connected to a node, whether directly or through terminal servers has no bearing whatsoever on what kind of DECnet license is being used.

Q. Category E question 1. John Irving from Armco. We have in house software that uses port IDs, i.e. TXA1. We installed the session support utility SSU. SSU changes the ports to the form TBxxx or VTxxx. This has caused us to write some DCL procedures to get around this that are let's just say, in quote "not clean". The question boils down to, is there anyway to retrieve the physical port ID from ports using SSU?

A. The answer to this is that there are a couple of ways it can be done. One is to go through and initially log into the port and then use the MCR command, MCR SSU ENABLE and then when you're done with the session MCR SSU DISABLE and this will allow you to get to the logical name for the port ID. In addition to this we want to thank you for this suggestion. There is a field that's been added to the UCB and that's something that we should be able to do at some point later. Thank you.

Q. Question F 1 comes to us from John Reynolds at Eastman Kodak. The 8700 occasionally hangs. Apparently there's an output buffer in the PRO 380 console that overflows. How can I flush this buffer while the system is running? Second part is, how do I flush it when the system hangs?

A. This actually would have been a good question to ask at the hardware Q&A, but we went and found the hardware developer who is responsible for it and while I have here his answers to the question. What he says is, we've been working hard since last DECUS to correct this problem. Part of the answer lies in Version 8 of the console software. The other part lies in some VAX VMS patches that will be available from Field Service with console Version V8 and that's all I know about the problem.

Q. Question G1 is brought to us from A. Sirrell of Westernhouse Electric. After Version 4.6 modified UAF records went to...I think the number is 1412 bytes....$SETUAI is not being used by any locally written software, nor is BPR or other high level software. This is a problem due to the DCL read buffer size limitations.

A. Other than the bug that was discussed in the VMS Notes conference, I assume here..this is the first time we've ever heard of this problem, okay. Is it possible that you've installed some software of a third party product which might be using SETUAI. Please note that the only supported method for modifying UAF is through AUTHORIZE or to the SETUAI system service. The one exception is if you're using the user data area of the UAF.

F. As far as we know nothing was done with a third party product. Of course it's always possible, but nothing that we're aware of and we saw it on, for example, on one of our own accounts immediately after modifying our password through AUTHORIZE.

A. The final part of the written response is, if you can isolate this problem to a particular component within VMS please submit an SPR and we'll investigate further.

Q. Question H1 is from Donna Dickerson from USAA. It's actually three parts. It says, in Version VMS 5.0 can a node spec be used in the FROM parameter of BACKUP commands to backup disks in remote nodes to tape drives on the cluster? The second is in VMS 5.0 can SYSMAN execute command procedures on specific nodes? And, the third part is, will there be a last read indicator for read only files aside from using the expiration date?

A. I'll be able to answer the second one. SYSMAN is quite happy to execute command procedures on any node you wish. It will take any single line of DCL, including an "@" command procedure, followed by a set of parameters.

A. All right we have a note from Andy Goldstein in response to the FROM parameter, he was unable to determine for sure what this question was really asking for and came up with four possibilities. In the interest of brevity, I think that it probably is the fourth alternative that the user had in mind which was backing up from some remote disk, backing up some disk from some remote node to local tape drive. That seems to match. The fact is that DECnet file access does not provide enough of the file characteristics that are needed to backup a disk at this point in time and therefore it is not possible using the current file access protocols to do this function, so it cannot be done and probably will not be done in the near future.

F. There were three parts to the question. The last one was will there be a last read indicator for read only files, so you can track, evidently file usage?

A. We're aware of this issue and it is being investigated for a future release. No commitment. Paul pointed out that there is a new product DFS which allows disks on remote systems to be locally mounted. I have no direct information about whether or not BACKUP can be used in that case. If we're going to talk about products there's also the RSM Remote

System Manager product which has the capability of doing backups over the network on a scheduled basis. At least most of BACKUP, probably not an image backup, but...yeah. DFS or RSM.

Q. Okay, this is category I Question 1 from Mike Mier of Shuttleworth and he asks, from DCL, i.e. in a lexical function or however, how can I determine what identifiers the current process holds?

A. Okay, the answer we have here is, right now there no easy way to do that however we expect to provide this capability at some point in the future. In extremes there is a work around to do this. What you can do is a SHOW PROCESS/PRIVILEGE sending your output to a file and then you can read from that file. [boos] I didnt' write it. [laughter]

Q. Question J1, Paul Plume, Luken Steel. [There is] a problem with LINK beginning in Version 4 when putting logical names for /SHAREABLE/NOEXE images and tables. SYS$TRMLOG doesn't know about, on second pass processing of the OPTIONS line LINK uses SYS$TRMLOG2, process this file, the status of course is not checked since RMS$PARSE worked in past one and in further processing an access violation occurs. Did you change this to use SYS$TRMLNM for VMS Version 5.0? This does not occur when linking the sharable image, but when linking against the sharable image.

A. This is in fact a message from back home since I really don't know much about the linker. We are aware of this problem. It was not..... [The tape ended and part of the answer was lost -ctp] ... one of the maintenance releases following shortly after V5, the fix soon.

Q. This is category A Question number 2 from Porsia Shale, Hughes Aircraft Company. Is there a way to make a non clustered system disk structure clustered without having to reinstall VMS?

A. This question appears to be asking about the shared system disk structure. The good news is in Version 5 all VMS system disk will have that structure whether you need it or not. So, you'll never have to worry about converting it. Prior to Version 5 if you do not have a disk of this structure, the way to get VMS to create it for you is to reinstall an upgrade, such as, if you're running 4.6, if you reinstall 4.6 you'll be asked the question do you want this structure, answer it yes and allow the upgrade to complete.

F. But you have to install 4.6 not 4.7, right? I mean, if I have 4.7 right can I do it, can I reinstall 4.6?

A. Does anyone up here know?

F. There is an easier way of....I would have to preface it with the remark, that I don't think they will tell you that it is supported...

A. You're right.

F. KITINSTALL or actually VMSKITBUILD will do that. There is another .COM file. We were just discussing this in the camp ground, but there is another .COM file on SYS$UPDATE that will allow you to add the common directories, MAKEROOT or...

A. No, I don't think it MAKEROOT works...

F. ...SYSBUILD with the COMMON option will do that for you.

A. Just the comment on the upgrade, you'd have to use 4.6. 4.6 is a remastered release, that is an upgrade. 4.7 seven is just maintenance update. Reapplying 4.7 won't help you with this.

F. Well, my problem was right now I'm running 4.7, can I...can I do an upgrade 4.6.

A. You want to downgrade...

F. I don't know.

A. Let us know if it works. [laughter]

Q. This is Question B-2 from Scott Morgan at EGS and the question is, what is the performance hit on 3000 and 6000 series processors for BACKUP (because of lack of CRC instruction)? What is Digital doing to fix this and when do you expect a fix?

A. I don't really know what the performance hit is and by the way this one doesn't have an answer from back East, so I'm making it up on the fly. I heard at earlier sessions here that it was between 2 and 6 times, I'm not sure if I want to believe that or not. I do believe there's a significant performance hit. We are looking at possible solutions to make this better, probably, not as good as if it were an instruction in the processor. The fact that next year is not enough is difficult, because I don't think we can fix it that fast.

A. Ralph, I can add something to that. There actually was an answer from back East which I just checked on a few minutes ago. Keith Wall put something in for this. His point is that, in fact, for most applications the CRC instruction is not really the bottleneck and the bottleneck is really the speed with which we can read the input disk and in the case of some of the smaller systems, namely the MicroVAX I, emulation of CRC is an issue because it's processor is slow enough that you really are CPU bound there, but in the 6000 series the speed of the emulated CRC is not expected to be a factor at all and only perhaps minimally on the 3000 series. Keith also talked about some things that are coming down the line. He's working on getting the bottleneck for the input disks removed and when that happens then you'll start the CRC become a bottleneck. [laughter] You remove one bottleneck and you open up another one, but he said there's also some work to see about off loading the CRC to hardware and this requires some MSCP software changes and those are further in the future.

A. Definitely won't happen next year.

Q. This is Question C2 submitted by Thomas Holts of Carolina Power and Light. What can I do to increase performance when the data base I'm using requires several million files and has very large directories, many of which are full?

A. I've got an answer from Andy back East. This is the last question that you'd expect a guy who's favorite command line interpreter is XDELTA, to answer it, so I'm completely at Andy's mercy here. And, he says that you'd be well advised to restructure your files into a directory tree instead of using the single level of directories. He goes on to explain that there are a couple of problems in the design of the file systems buffer manager in VMS Version 4 that makes performance on very large directories such as the ones that you'd have by implementing this application at a single level, they make that performance much worst than it should be. They show up mainly when the directories are being modified as well as being read, that is additions and deletions to the directories, not modification of the data in the files. The problem exist as well in VMS Version 5. We understand them, but we're going slowly and carefully making modifications to the file system buffer manager, because the one we have now may work slowly, but it works correctly. There's some more detail here. Depending on the length of your file names and the patterns of file usage, you can expect 10 to 20 files per directory block. Since the maximum supported directory size is 1024 blocks a million files in one level of directory is excluded, not possible. You can factor the structure into two levels giving you about a thousand directories with a thousand files in each, each directory would would be 50 to 100 blocks in length. The file system searches directory reading up to 32 blocks at a time, but because of the above

mentioned problem the size read is actually often much smaller. Thus, you might actually be better off factoring the structure into three levels with the 100 way branch at each level. Then each directory would only be five to 10 blocks long, would be read with one I/O in most cases and the overhead of the additional levels of directory would be optimized out by RMS's directory path cache, when their repeated references to the same directory. He goes on to answer the question of how one might factor the directories. He suggest partitioning the files by any simple hash or partition on the file name that's likely to give you a reasonably smooth distribution. Finally there's one other consideration if you're using mechanically generated file names, that is names generated by software algorithms rather than by users, don't generate file names who's lexical value is monotonically ascending in time. For example, don't use a generic prefix plus the time of day in hex. Because of the way directory space management works, the long term results of constructing such file names is a very sparcely populated directory file giving you very poor performance and eventually giving you directory allocation failures. If you need to put the date into your file name, byte reverse it or use some other non monotonic representation.

Q. The next Question is category D Question 2 from Jeffrey Hines at NASA Marshall Space Flight Center. The ET driver only allows 43 receive buffers. This is not the way the XE driver handles Ethernet receives. We have a patched driver from DEC allowing our 8250s to run like our MicroVAX IIs. Has the driver been fixed for VMS Version 5.0?

A. Okay. In VMS Version 5 ET driver has been modified to offer greater user buffer allocation and it can now handle up to 255 receive buffers. In addition, ET driver allows up to 31 Ethernet users which is up from 14 in Version 4. So, customers that were hitting the 14 user limits will now have a little bit more head room.

Q. This is Question E2 and it's a fairly long one. Richard Colletis from Lawrence Institute of Technology. We have a Fortran program in which we SPAWN a subprocess with NOWAIT. Program passes data to the subprocess via a mail box, then the program can go on or it can exit. When the subprocess is finished it sets a common event flag and waits for another flag to be set. The main program, when the user selects the appropriate option, sees (via a CEF cluster) the set flag, then sets the event flag that the subprocess is waiting for, then the subprocess sends its data back to the main program (process) via the same mail box and goes away. While the subprocess is waiting it is in CEF state. The problem and it's stated in two and I'll think I'll split them as they're both fairly long. One, occasionally when the parent process presses control T, a status comes back for both the parent process and the subprocess. This is intermittent, but not really a problem, why does this happen? And, the second part is the major problem.

A. All right the answer for the first part is that there is a field in their terminal driver's UCB which is UCB$L_TL_CPLPID. By default this value is zero and there is no controlling owner of the terminal UCB and under these circumstances the control T AST is delivered to the process and all subprocesses. As soon as you do a SPAWN that field is set to a non zero value and the ASTs are delivered only to the process who's PID is that field.

Q. Number two. The real problem is, when this subprocess is running and we are at DCL we can usually type "STOP /ID=x" where x is the ID of an unrelated user (different UIC, disk, etc.) and it works fine. However, once in awhile I type STOP /ID=x and the subprocess also goes away. This can happen whether the subprocess is in LEF or CEF or any other state (the process x does go away). To reiterate this is intermittent. Why does this happen?

A. And, here we really don't have an answer to the best that we can tell you maybe sharing a mail box by mistake and that somehow those two processing are walking on one another. We recommend running accounting and taking a look at the final status for each of the processes and see if the UICs are somehow related, such as in the same group.

Q. Question F2 is from David Hubert at Pacer Software. David asks, why were the Ethernet common routines removed from the VMS fiche? They were in VMS 4.4 fiche but are missing from the 4.6 version. Are they going to be in Version 5.0?

A. This is actually a more specific instance of the follow up question to this so...I'd like to do a more general question first.

Q. I'm going to ask F3 which is from John Saunders at same place. Could you state the policy on exclusion of VMS components from the source listing fiche. In particular I need to know when to complain that something has been accidentally omitted and when to complain that policy details should be changed? [laughter, applause]

A. To help you make sure that your complaints fall into the right category, let me give you some guidelines for the sorts of materials we omit from the micro fiche. In general we omit anything that has to do with premature disclosure of an unannounced product, anything that has to do with how licensing is done or how keys are made. We omit anything that might be a possible violation of export regulations, regarding encription or other security features. We also omit items that we believe are proprietary, that disclose possibly patternable technology or are trade secrets to Digital and we avoid anything as far as we can that's a security hole, in fact we generally omit most security related software from the fiche. That's some of the general guidelines we use when we look at source code and try and figure out what materials should be on the fiche and what shouldn't. There was a specific question about the Ethernet common routines, why were they on 4.4 fiche and not on the 4.6 fiche. In the 4.6 Ethernet common routines there were portions of the ECRs that are in fact proprietary to Digital and changes were in 4.6. What we are looking doing in the 5.0 fiche is shipping as much of the Ethernet common routines as we can without including the proprietary information within those routines on the kit.

Q. The second appearance is category G comes from Jim LaQuare of Pioneer High Bred International. Need the ability to INSTALL or LINK an image with an ACL identifier. We could then easily grant "write access" to the production RMS data from production images only.

A. Okay. There's a parenthetical expression at the end that says we currently grant and ID on the fly and attemp to revoke it before the user gets to $ prompt. The response is, thank you for you suggestion. We recognize the need for so called protected sub systems and we expect to provide such a capability at some point in the future.

Q. Question H2 comes from Michael Tang of PAFC. If during a multi volume stand alone backup a physical IO, I believe, error occurs on the backup media one has to restart his backup from the beginning. If one is on the first volume it is aggravating. If one is on diskette 92 of a 105 it's darn frustrating. Could you possibly provide a solution for this?

A. I'm getting more and more in the dark on this. There is no answer from back East and I think that the only response is, thank you for your suggestion. I think we realize that this is a problem and hopefully we'll be able to find time to do something about it.

Q. Category I..Question number 2 from Louise Kwee of Measurex Corporation. She asked, how do I get the command line entered by a user in order to use CLI$DCL_PARSE to do further command processing? Assume that I have begun execution by entering an image defined in a .CLD file. I want the whole line entered rather then taking pieces with CLI$PRESENT and CLI$GET_VALUE.

A. Okay, the answer to this is fairly straight forward. Call CLI$GET_VALUE with the $ line as the label requested. This will give you the command line after having been processed somewhat as in upper casing, removing of multiple spaces, conversion of the date key-

word, such as yesterday, etc. and again more of a description can found about this in the CLI$GET_VALUE utility description.

Q. Question J2. Michael S. Pumne, Martin Marietta Energy Systems. We are running a detached MONITOR process (MON SYS /ALL) on each of the three nodes in the cluster, if one CPU is either shut down or crashes the binary MONITOR files on the other two nodes appear to get corrupted. By this I mean, when we run the MONITOR to convert the binary format to ASCII format we get the following error: `MONITOR-F-CLASMISS Requested Class Record Missing From Input File`. What am I doing wrong?

A. It doesn't sound like you're doing anything wrong. This looks like a bug. That error message should never occur and if it does it is indicative of a bug. The three processes running MONITORs should be totally independent and if one of those nodes goes down or gets shut down the other MONITORs should continue to write into their binary files. So, please submit an SPR on this one and when you do, show us the command line that created the detached process, also the command line that issued the MONITOR command and if possible send along the corrupt reporting file.

Q. This is category A Question 3 and comes to us from Joseph Crumb, an independent consultant. On a cluster we have an application written in COBOL which seems to have more occurences of locked record contention when users are running it on both nodes then when running on one node only. Are there conditions where locking granularity is different for a cluster than for a single node? The files are very large, some with multiple keys.

A. Okay, I believe I discussed this problem with the author this morning and he provided some additional information. The contention that he's talking about is not a performance problem, so this is not a performance question, it's really that the application is seeing some form of error messages, perhaps, record locked, we're not sure, when the application is run on multiple nodes and not on one node. We're somewhat mystified by this. The granularity of record locking is the same in one node or across multiple nodes. So, either there is a bug in VMS of which we're not aware or there is some application behavior that's not evident from the description of the problem. We'd encourage you to report this as a SPR with all the supporting information that you can.

Q. This is category C Question 3 submitted by Neil Schmidt of Inland Steel. What is the most efficient disk hardware configuration to use to house your page and swap files for an 8800 based VAX cluster in terms of speed, availability and so forth?

A. I have a couple of answers from back East. I think I'll start with the one that seems to contain the most detail. We have observed that the average profile for paging IO under VMS is small. That is we can do IOs which are 8 blocks and downward. With this type of load it will be very difficult to saturate a KDB50 disk controller on an 8800 when you've got say 4 spindles on that KDB. So, in regard to the fastest mechanism for paging and swapping we think KDB or an HSC base for a pager swap file are probably equivalent in terms of speed. However, the availability characteristics are very different. A KDB under VMS Version 5 will allow dual pass failover between two nodes in a cluster provided that the total disk name, the fully specified disk name, is the same on both systems. Dual HSCs will provide fail over for clusters as small as a single VAX. There are lots of other more important questions to answer in figuring out the configuration you want. For example, HSC are considerably more expensive, however, they're considerably more expandable. They scale well on a cluster where the local disk space solution doesn't. They provide you with tape controllers and so forth. On the other hand the KDB is a considerably cheaper solution, but you have to bear in mind that it is also a bounded solution. Given these guidelines we recommend that you analyze your requirements and after looking at your application make a reasoned decision as to which configuration best fits your application, your budget and so on.

Q. Okay, the next Question is Category D Question number 3 from Ron Frederick at Niper in Oklahoma. My users have two LANs of micro computers which they wish to link into the DECnet. I do not want them listening/analyzing DECnet traffic and eventually posing as SYSMGR. How to put them onto DECnet without giving them the opportunity to observe privileged processes, a bridge between the secure segment and their segment perhaps. Yeah, the parenthetical statement is user mode and the question came with a diagram which I'll hold up so you can all see. The diagram actually contains some topological impossibilities as well.

A. There's a relatively lengthy answer here which I will filter somewhat myself which came back from the East Coast and he says, I don't know what you mean by "link into DECnet" in this case, let me guess what you want. You want to link to LANs and some other nodes together and let them communicate with each other using DECnet. You do not want one node to be able to look at DECnet packets for and from nodes not on its own local LAN. The diagram in particular shows two LANS connected via bridge to a DELNI which presumably represents the other LAN and the point is the picture shows three connections into a bridge and only two connections are allowed into a bridge. If you have 3 LANS you need at least two bridges. A bridge can be used to filter packets on address, but not on protocol type. Another thing you may want to know is that bridges can be used to filter multicast addresses. They can be used to prevent some Ethernet applications, for example, DECnet, LAT and LAVC, from traversing a bridge. The reason this works is because most Ethernet applications used specific multicast addresses to find their counterparts on the Ethernet. If you prevent an application's multicast address from traversing a bridge you've prevented that application from traversing the bridge. We use this in our lab to keep Ethernet cluster traffic isolated, to keep boot request isolated. We have one extended LAN that actually has, I believe, something in the vacinty of 90 LAVCs on it. So that turns out to be very useful. However, now to amend the answer that came back a little bit, from looking at the diagram. Bridges of...at least my understanding is can probably do some of what the requester wants.

[A question and parts of two answers were lost here. The partial answer with no question was deleted -ctp]

Q. Question H-3, submitted by Jack Peters from RCA Advanced Technology Labs. An annoying property of the present version of the BACKUP utility is that it leaves tapes positioned at EOF and allocated while it does its record pass. We have a 4 RA-81 volume bound disk set, and the record pass can take hours. Has this been corrected in V5 and if not, is it planned?

A. Yeah, we heard this same problem in the SIR's, and since I don't have a copy of that, the developers are aware of this. The problem is being addressed, it is not fixed in 5.0, and we're not going to commit to a date. But it is a known problem, and something will be done.

Q. This is question I category 3 submitted by Terry Mish, from Naval Surface Warfare Center, who asks, there is a problem in MAIL in VMS 4.7. If you're sending to a distribution list and a member of the list has set forwarding to user before him in the list, the process hangs in MAIL. Has this been fixed in version 5.0?

A. O.K. the very simple answer is, no it has not been fixed in version 5.0. We are aware of the problem and it will be fixed in a future update. I'll go into just a little bit of detail. It's not specifically a problem of distribution list and MAIL, it is a problem with the fan in of VMS MAIL addresses. If a person on node A attempts to send a message to two people on node B, one of which has forwarding set to the other, then VMS MAIL will optimize the addresses and only send a message to one of them since they're both forwarded to the same place. The problem is that nobody tells node A that this is going on, and kind of

sits around waiting for a reply. And as I said, this is a known problem, it's not in Version 5.0, but it will be fixed in a future update.

Q. This is question A-4 that comes from Victor Shanes, Banker Trust Company. How can I insure that my system disk and quarum disk are dual ported (2-HSC's)? The only way that I know is to bring the system down and check the HSC's. SHOW DEVICE/FAULT does NOT pick up the second drive. Also, if an HSC drops, the DEVICE/FAULT may not pick this up.

A. Is the proposer of this question present? SHOW DEVICE/FAULT almost always will tell you if there is a secondary path. It shows up in the display under the label of alternate host names. There are a few circumstances that might prevent this from happening. Could you elaborate?

F. The system disk never shows it on our clusters. Nor the quarum disk.

A. O.K. the ultimate way to discover if the disk is dual ported or not is actually much easier than describing the question. It's to go to the disk and release the lighted port button, cause some activity on the disk... [laughter]

F. Not on a production system. I'm not going to try to find out.

A. Excuse me.

F. I'm not going to try to find out on a production system if it is dual ported.

A. Well, there's very little risk in doing this. You release one button, if there's activity to the disk, the other light will immediately come on. If it doesn't, push the button back in.

F. But what happens if in fact there is a requestor slot on the other HSC or the drive itself is bad or something. I mean the HSC, the other HSC is down. I'm going to have a production problem.

A. The notion is that you have the disk operating on one path. The most you're going to do is deny service on that path for a few seconds. In less that path breaks at the point where you push the button it should restore operation the minute you restore it..

F. But I'd like to be able to go to one of the consoles, and effectively we check all the drives, rather than going to the drives. I've got 120 drives out there.

A. The SHOW DEVICE/FAULT should give you the information, we don't understand if you've consistently do not see it through SHOW DEVICE/FAULT on the second path and the second path exists, we don't understand why and perhaps a SPR with supporting information would be helpful. One additional caveat is if you have shadow sets, don't do the button popping.

Q. This is question C-4, submitted by Louise Cooley, of Measurex. Why would a process hang up in an RWMPB state when the number of pages on the modified page list is slightly less than MPW_WAITLIM SYSGEN parameter. The hanging process prevents completion of the initialization of a real time process control system.

A. O.K. process enters the resource way for modified page busy. when it attempts to remove a page from its working set and place it on the modified page list, and the current size of the list is greater than or equal to the SYSGEN parameter MPW_WAITLIM. In V4 systems a process leaves the RWMPB state, when the modified page writing tread terminates and the current size of the modified list is less than the size specified by the parameter MPW_LOWLIMIT. Actually less than or equal too. So, given that there are a couple of possibilities that might explain your problem. One is that the modified page writer has

terminated with the number of modified pages still above the low limit. This can happen in, cases for example, where one or more of the page files becomes full, there's no place to write the pages. The other possibility that occurs to us is that the SYSGEN parameters I mentioned earlier don't have the proper relation. In particular, if MPW_WAITLIM is less than MPW_HIGHLIMIT it is possible for a process to enter modified page busy wait, but not to trigger a modified page write. Because the number of pages on the list hasn't exceeded the trigger point as described by MPW_HIGHLIMIT. The solution in this case is to insure that wait limit is greater than or equal too high limit and given some other discussion I had with Louise earlier about this, I suspect that the latter is the answer to her problem. V5 provides some hope in this area in that we at least decreased the amount of time that a process stays in modified page busy state. Once the modified page writer has been triggered, the process is waited only until modified page list gets down to a level below that indicated by a new SYSGEN parameter, MPW_LOWWAITLIMIT. So we recommend setting this new parameter equal to high limit minus MPW_ WRITECLUSTERS, the first time that modified page write completes it's possible for processes to begin coming out of the modified page busy state.

Q. This is question D-4, and I'm going to read the configuration it's from Bill Norton, Harnish, Stager Engineers, Inc., Milwaukee, Wisonsin. The configuration are 2-730's on a DELNI and DEUNA's running VMS, version 3.5. Using SET HOST to connect from one 730 to another 730. Runs a dialogue program that eventually does a report to the terminal and then (this works fine on directly attached terminals) produces the following messages, %REM-F-Net Error DECnet Channel Error On Remote Terminal Link. Next line %REM-S-N Control Return To Node XYZ, %SYSTEM-W-Data Over Run Using IBT From An 1173 and then (M plus version V4 produces the same result). And I assume that they want a comment on this. There's no question, if that's the situation.

A. O.K. Is Mr. Norton here? Because there were some questions came back, but basically, our action basically is having successfully carbon dated the problem. It appears as though there might be a case of process resource exhaustion on the remote node. However, it's a little hard to gauge you know, this far after the 3.5 exactly where the problem might be and we'd be very interested in seeing it reproduced on a more recent system.

F. Granted it's a good argument for getting to upgrade but it's a fairly static turnkey system.

A. O.K. I mean we're not aware of any specific problem, certainly currently that causes this problem. The questions here that came back out since you're here, this seems to have been answered, but I'll ask you again anyway. Does the program run correctly on both nodes when using locally connected terminal, or is it, does it fail equally on each of your 730's, I mean does it succeed equally on both 730's when directly connected?

F. Yes. It's essentially a single machine system, and the other one is just being used to provide a very low cost way of getting up a few more terminals running on the active one.

A. When the link dies have you looked at the accounting log to see what the exit reason for the program is in the remote end? That kind of information can sometimes be very useful ... one question, there was a major revision if my memory goes that far back, on the protocol used by SET HOST about that time, wasn't there? ...I believe that was in C-4 that we went.... That's why I'm asking... Yeah 3.5 was already CTERM... I remember some incompatibility with our SET SYSTEMS, and SET HOST at that time... Basically, phase 4 DECnet VAX appeared in version 3.4. My mind barely goes back that far too. I think there is one other possibility. I had the same problem I don't think it was that long ago, but more recently where if you're doing QIO's to the terminal and you do a SET HOST and then your application does the QIO outputs, if the block size or the buffer size you are trying to write is greater than the DECnet buffer size you will get data over runs. That was only with the protocol that was used before we went C terminal. We went to C

terminal, we actually broke the QIO up into whatever size was usable on the remote end or on the RT pads sites, so that the QIO's would succeed. Another suggestion from the audience, check the MAXBUF parameter.

F. In this case MAXBUF is up around 2K so we can do full screen SMG writes. The QIO may be bigger than MAXBUF even? What I was looking for was a hint of what parameters I might start trying to make bigger to see if I could affect this.

A. The version number. [applause, laughter]

Q. This is question F-5, Scott Mogan from ENS, states that four years ago we were almost totally to DEC shop. Now maybe only 25% to DEC. Management seems to view DEC as non competitive like IBM, so it's difficult to justify new purchases, what is DEC doing to be more competitive, to help us help you?

A. This question is inappropriate for this particular forum.

Q. The 4th Question in the G category is from Doke Bain of Kelsa Inc. When security auditing is enabled, for any item, OPCOM transfers, i.e. broadcast, the alarm messages to all nodes in the cluster. This results in expensive and unnecessary redundancy particularly for file audit trails. A comment on the back is, I believe these messages need only to be logged on the single node where they originated. Please comment on what is being done in this regard.

A. Well you've got two comments, from two different developers. The first is from the person who does security, and that says, official quote from me, Mark Pilean. As I have mentioned at several DECUS conferences the security auditing in VMS is slated for overhaul adding additional features and capabilities. A second comment from the owner of OP-COM essentially. The design of the cluster wide OPCOM was intended to insure that the operator domain was consistent across the cluster. In particular the intent was that any event should be available to any operator, anywhere in the cluster. If this were not the case, one would have to search all the operator log files in the cluster for an event, rather than being able to look in a single log file. To eliminate the redundant information we suggest that you disable log files on all but one or two of the nodes in the cluster. This can be done by inserting the following lines in the SYSTARTUP.COM file, to DEFINE/USER SYS$COMMAND OPA0 REPLY/NOLOG. A similar command can be used to disable the operator console terminal on some of those nodes in the cluster. In V5, many of the normal connections manager messages are no longer displayed. This removes some of the cluster clutter on consoles and in the files. We have no intention to change the operator messages in general, to be specific to a single node. And then the owner of OPCOM says, a question for the audience. Another OPCOM enhancement has opened up the possibility of enabling or disabling specific operator classes for the log file. For example, if you do not want cluster messages in the log file you could stop them by saying something like, REPLY/NOLOG=CLUSTER. Question is would this be a useful feature or just additional clutter in the command dictionary. [Applause] I take that to be about a half useful feature.

F. Could we hear the votes for the, additional clutter. [Applause]

A. Sorry about that, it's more like a 80% useful feature.

Q. Question H-4 is from Kevin Angley from Memorex Atelex. When using BACKUP to copy the directory spec [A.B]*.* to [X.Y]*.*, the directory X is created with the attributes of directory B. Well obviously it should map from directory A. Note that this behavior is different if directory A contains files prior to the appearance of directory B. This inconsistency pretty much says this is a bug. He has a note saying, I filed an SPR, and was told that this is the intended behavior.

A. I have a note from the developer responsible for BACKUP who says, his is indeed a bug, BACKUP does it own file scanning and most of its own directory parsing. It keeps a stack for the directory level. At first guess I would say it is looking at the wrong level in the stack to create the first directory. We will fix this and SPR marked, Wells says this is a bug. Comment not written in ???? has been read in a public forum. He doesn't live in my office, should expedite attention, to try and get and insure that we will fix it. Keith Wells. There is another note from a file system guru that says, I'll bet this problem is another side effect of the fact that top level directory is not selected as a file. As the result its attributes are never made officially available instead it gets them from the first trial, we selected as a file, when it is implicitly created. So there may be a heat available here for Keith who will attempt to find your problem. There is no information here as to who it was that sent back your SPR with the coded answer that is was intended behavior.

F. It was the screening group that gave me that answer and they also told me that it had been reported as bug all the way back to 4.1, I think. But they...

A. They said it was included as a bug all the way back to 4.1 and it is intended behavior?

F. That's exactly what they said. [laughter/applause]

Q. This is question 4 - category I, submitted by Francis Recolin from Oracle. It's a rather lengthy question. The requirement is for the MAIL utility to remail messages with explicit specification of the firm mail address. We have to bridge several incompatible mail protocols and currently re-mail with MAIL, and it says, /FROM= the MAIL utility. But the re-mailed message then is addressed from the process that the MAIL utility ran under. Got that, now if MAIL utility had a /FROM= option we could extract the firm address out of the mail message header and re-mail with the explicit definition of the firm address. Preliminary VMS V5.0 documentation mentioned, /FROM= as a new MAIL option but it seems it was disabled, for security reasons. We need the /FROM=, option for our very high privileged application which is under the control of the VAX VMS system management.

A. Want to guess the answer? This is directly from the MAIL developer back in Spitbrook. There is no supported way to send a message on behalf of another user. This is kind of clarification here, what the user is looking for is something like, SEND/FROM=userA to make the message look like it came from userA when it really came from userB. The /FROM qualifier you are getting it confused with was a SELECT /FROM= command, which allows you to select messages that are in a particular folder from userA. And the final comment is, it is very unlikely that SEND/FROM= will ever be implemented in VMS MAIL, even for privileged accounts for security type reasons.

Q. The next question is, category A, number 5 from David Plante with MA Com. Draft VMS 5.0 release notes as a restriction on multiple Ethernet interfaces for any cluster node in a LAVC or in a mixed cluster. Does this apply even if the second Ethernet adapter is used only for non-DEC protocols?

A. The protocol that is used for cluster communications only supports one Ethernet adapter and that is the first Ethernet adapter that's state CONFIG finds. It is only the cluster protocol that is restricted to a single Ethernet adapter. It's quite possible to have nodes participate in a cluster that have multiple Ethernet adapters. However, since the question mentioned non-DEC protocol, there's something that I should add to this. In looking for Ethernet adapters, stand alone configure connects the standard VMS Ethernet driver to all controllers that it finds. So if the non-DEC protocol requires it own unique different Ethernet driver then there is a problem. But if the non-DEC protocol works through the standard Ethernet driver then it will co-exist just fine with clusters regardless of how many Ethernet adapters there are.

Q. This is category C, question 5, submitted by Court Reason of Promise Systems Corp. and it's relatively lengthy. We've experienced page file fragmentation that is strongly correlated with the use of RMS global buffers. It appears that RMS VMS is losing track of what is and is not free in the page file. When the application gets into trouble with respect to page file fragmentation, even if we delete via LOGINOUT, all processes connected with the application and hence, with the global buffers, the page file remains heavily subscribed to and fragmented. The evidence for this is no other processes or applications are left running, permanent global sections are removed, and then a re-start of the application immediately reencounters the page file fragmentation. There is some evidence that other sites experience this. Is this a known problem? Is it fixed? And are there any suggestions?

A. We've talked this over both here and we've also sent it back east and nobody has any advice for you on this other than to submit an SPR to us on it. We're really stumped. We'll obviously take it away and look at it in advance of receiving your SPR.

Q. Question from George Pandelios of GTE, Government Systems. Want to be able to check audit criteria from an ADA subprogram. Potential solution, SPAWN DCL, SHOW AUDIT/ [unintelligable] file and parts file contents. Alternative solution, input macro code to change mode to kernel and check appropriate bit. Question: Is there a better way? We don't want to give out change mode to kernel. I think that's appropriate, please comment on alternatives. What is the best way.

A. This is part of the planned overhaul of the security auditing. Noted: Aren't you sorry you read all of that Dave? Wait a minute, did you want to discuss this some more, I'll take a shot at discussing it with you if you want too. The general idea is we recognize huge deficiencies in the security auditing features in VMS. And we're working to make them better.

F. You can't comment as to time frame, whatever?

A. Not being the person trying to write the code, I wouldn't dare do that. Not even random guesses. I hate to commit other people on their work.

Q. Question H-6 is from Pamela Fare from Portsmouth Naval Shipyard. Is there a problem with RMS journaling in VMS 4.7. I heard there is a potential for data file corruption. I will not be physically present at this Q&A. If you can mail me an answer it would be much appreciated. We are planning to use RMS Journaling to maintain integrity of a million engineering drawing index file?

A. As far as index files there are no known problems of this sort at all. And in fact, up until a, shall we call it an informal problem report submission Monday morning, at the presentation of status of VMS, we didn't know about this problem in terms of RMS Journaling. There are some known fixes for our RMS journaling in particular some having to do with dealing with tapes. They are available for the technical support center and in fact, this problem, now has since we have talked to the person who suggested their was a problem, talked to him this morning and we do know what that problem is and a patch for that problem will also be available. However, that problem had to do specifically with sequential files and a fairly unusual access pattern, namely a whole slew of FIND operations to skip to the end of file. So there are no known problems with respect to index files at all, which is apparently this person's concern, I don't think that he needs to be concerned. Also, you can get in touch with the technical support people to get some set of patches for RMS Journaling that address kind of unusual problems.

Q. This is category I - question #5, by Howard Holcohm from RCA. He wants to know how can I call the DCL function from a program without performing a SPAWN. Specifically, one, DCL with all switches and number two, SET HOSTS with account password and log in as per the DCL commands. Run second image within first image?

A. O.K. I have several answers to this questions and basically they all say the same thing, which is, you can't. A lot of our utilities are not callable and you just can't access them from within a program.

Q. This is category A, question 6 from E. Mark Unger from Software Techniques. Is there anyway within a cluster for a program to determine if a disk is being used on a remote node as a system disk? I would like to do that without querying each node for information.

A. A direct interpretation of the question has as its answer, No, there's no way to point at a disk and say, is this being used as the system disk by a remote node that either I or wisdom from back east could identify. There are some indirect ways that you could determine this if you and your system management procedures cooperated a bit. For example, you could do a scheme along the lines of one A node boots, have a process that opens the file and keeps it open for the life of the machine. And a process on another node could attempt to do a conflicting open on the file and the failure of the conflicting open would indicate that, the remote node was booted and that this disk was its system disk. I believe there are some ways you can contrive to discover this information, but there is no direct answer that we can think of. If the poser of the question is present, perhaps he could tell us what the problem, being solved by this request is. While we're waiting for him to make his way to the microphone under version 5 the page, swap and dump files are open to the file system, unlike under version 4. So those are three instances of files against which you could check for conflicting open.

F. O.K. my main concern is that there's a symbol, EXE$GLSYSWCB which has windows on the system disk that they're actually not opened by the execute piece so they don't have FCB's. Now I do not want to play, I do not want to move, or touch the LBN's on the system disk, so if I'm going to do something with this disk, I want to make sure that the portion of the disk pointed to by this is not messed with. That's ..

A. Now I understand. You're writing a another disk compressing utility. You're asking about precisely that set of files which were under version 4, not open to the file system, but rather had been opened by SYSBOOT and other version 5 are open to the file system as well.

F. O.K. So if I could just know for sure that the only things that do not have FCB's are the things that are only used during system boot up and they're not accessed after system boot up at all.

A. That's correct. Everything that was formally in this, opened by the system but not opened by the file system state during steady state operation is now also open to the file system to the best of my knowledge. The quorum file maybe an exception to that. Also a file called I believe, CLUSTER.DAT. Don't go away thinking that the files that I mentioned are not open under version 4, they are open but the fact that they are open and being referenced isn't known to the file system. It's that inconsistency which we corrected in version 5, except for the couple of exceptions that Dave just mentioned.

Q. Question C-6,of Seth Stearns from Reliance Electric Corp. How can I control where my process dumps are written, parenthetically he has RUN/DETACH/DUMP.

A. Logical name SYS$PROCDMP. SYS$PROCDMP redirects the process dump file.

Q. This is category I, question 6, on the INSTALL utility from Bill Wade at Polaroid. When installing a user written .EXE, if there are not enough global pages and INSTALL is unsuccessful, but if the available number was 40 and 50 were needed, those 40 do not show up as available. We need to re-boot to get them back. Is this a known problem?

A. Not to us. There's no wisdom from back east. We've talked about it up here, we don't have answer for it, but we noted it and would ask that you SPR'd as well.

Q. This question is also from Bill Wade of Polaroid and the question is, has the disk quota out of synchronization problem been fixed in version 5. If so, please explain the cause and solution?

A. This answer is from back east, you can send all your rotten apples back via UPS. The answer is no, it hasn't been fixed in V5, we're still looking into it.

Q. Question G-7 from R. McClinton of Mitre Corporation. We're using the security features of VMS to determine if anyone is attempting to open files on a disk that he has no permission to open. We normally receive a security alarm if he is unsuccessful in opening the file. However, if the file is presently locked by a legitimate user, no security alarm is generated. And the illegitimate user is told that the file is locked by another user. I feel that the correct interpretation of the orange book should have this file access attempt be logged. Colorado rejected my attempts to SPR this problem, should I SPR this problem through the national computer security center? [laughter, applause]

A. This is an interesting response. There's a large paragraph at the bottom that has been covered over in invisible ink, and another one above that that simply says, noted. I have a feeling I'm not supposed to read the other paragraph, so, noted.

Q. This is Question category I, number 7 from Louise Ruley from Measurix Corporation. Does LIB$FIND_FILE_END free up memory allocated by LIB$FIND_FILE, comment; a process doing a thousand calls to each in a loop had a monatomically increasing working set. P.S. I no longer have fiche to look at, and TSC did not know.

A. O.K. I'll answer this in two parts. First of all the question, does LIB$FIND_FILE_END clean up the memory allocated by LIB$FILE_FIND, and the answer is yes, it does. The second comment here on her process is that, there's a question, is she here, she's not here. Basically the question asks, are dynamic descripters being used in your program and if so, your file requires that the only, only the string manipulational routines be used to alter the length or address of dynamic strings. Do not use LIB$GET_VM, but $GET LIB$FREE_VM for this purpose. And if you're still having a problem, as stated in the question here, please submit an SPR, because you haven't covered above.

Q. The eight entrance in category G, Stump the Security Team, is Sarah Kalick for the Institute for Defense Analysis. After the last symposium we were told as a security check, to check the checksums on various files, i.e. LOGINOUT, as a means of detecting an infection by a virus. Comment, I believe this was a specific case propagated by the chaos computer club, but are not sure. The problem, with image backups /RECORD changes the backup date of the file, and hereby change the checksum. The question, where do they get the check sum numbers they gave us to check against, and how do I keep them from changing while maintaining backups of my system disks?

A. Unfortunately, the CHECKSUM utilitie included with the VMS kit is not robust enough to be of much use. Remember I didn't say this. In several instances, the various hackers have included the necessary constants in their hacks to defeat checksum. As for the backup date, I just looked at the code for the CHECKSUM utility and there is nothing that looks at the backup date.

F. I had this question answered previously in this security clinic. And basically the answer was, I don't know who said it, but who ever said to use the DUMP/HEADER to get the checksum was wrong. And that's what we were doing to get the checksum. Rather than running the checksum program.

A. I think that the bottom line, even beyond that if I recall the CHECKSUM utility correctly the idea was run it once, get a number, run it again, and if the number has changed you know you are in trouble. That kind of thing. And that's not good enough, because

knowing of the existence of the utility, and it's algorithm those folks who hack LOGINOUT particular have become smart enough to put there hack in is such away that the checksum will check any how. Which leaves us with the challenge of finding a better mouse trap.

F. Right! I was also told that and the answer to that was it is better than nothing.

A. Certainly you will catch anybody not smart enough to a fix their hack. That basically only leaves you with bring LOGINOUT off the original tape, and DIFFERENCE on them. Or something along those lines. You know if you are really going to look for that stuff, compare it against the real file.

Q. This is question I8, from Harrison Spain at McDonald Douglas in Sypherist California. The new utility SYSMAN allows us to change SYSGEN parameters cluster wide. This seems to conflict with the "approved" practice of editing MODPARMS.DAT and running AUTOGEN. Comments?

A. The SYSMAN parameter functions allow you to preform the same thing that the SYSGEN parameter functions did. We still recommend that for permanent changes especially performance parameter changes, you still use the old method of editing MODPARAMS.DAT and running AUTOGEN. SYSMAN should really be used for looking at these parameters, and modifying that limited set of parameters which are not performance related things such as DUMPFILE, and other similar parameters.

RT-11 DUCM/DYC GRAF11 DEL DIR PLOT-10 IMAGE LIBED FSTATS MS/DOS TIC-TAC-TOE QIX VAX-LIB-DATMAN/VAX EDTPlus SPICE2 TREEDUPL LISPEX MCLS TYPE_PLUS MINC DISK_USE FRAG EDTEX PORT LOCATOR TECO CHPLOT NANNY DIR11-W WATCHDOG INACTIVE ACCOUNTS IMGSPL ICE TEX EDITOR VAX-LIB-4 GRAPHIC UTILITIES SETAUX.ARC STATPK FIGure KERMIT Distribution TENBACKU JUICER VTEDIT 2022 VAX-LIB-3 VISTA EDITOR MTU TDE RSTSOPEN DRAWTREE WATCHDOG PRM-1 SMARTMAILER TEN SPELL DECPoint of Sale JUICER PARALLEL Library V2 RTMULTI and Addo SMARTMAILER for RSTS/E CU FILTRA Spring 86 RT-11 SIG CP/M KERMIT S Invasion for PRO Bonner La SPLICE RUNOFF VAX-LIB-3 VAX-LIB-2 IMAGE SPELL TURBOCOM FNDFIL PC-8088 Collection #10 VT20 TOOLKIT PLATOOLS SMARTMAILER DEPROC LaTex KERMIT-11 FANCY FONTS XMIT CU ReGis to HPG CED International RUNITOFF JP5-JP6 FODT PASCAL-OS/8 ANISMT WPSIM PARALLEL LIBRARY DECSYSTEM-20 SIG Spring 85 CAMERA DELPHIN HACK BIBENTRY APFELN DIGITIZING Acid Docume Generator VAX-LIB-2 AMAR-10 AMAR-20 DATMAN/VAX IMAGE RT-11 DUCM/DYC GRAF11 DEL DIR PLOT-1 IMAGE LIBED FSTATS MS/DOS TIC-TAC-TOE QIX VAX-LIB-DATMAN/VAX EDTPlus SPICE2 TREEDUP LISPEX MCLS TYPE_PLUS AMAR-20 DISK_USE FRAG EDTEX PORT LOCATOR TECO CHPLOT NANN DIR11-W WATCHDOG INACTIVE ACCOUNTS IMGSPL ICE TEXT EDITOR VAX-LIB-4 GRAPHIC UTILITIE SETAUX.ARC STATPK FIGure KERMIT Distribution TENBACKUP JUICER VTEDIT 2022 VAX-LIB-3 VIST EDITOR MTU TDE RSTSOPEN DRAWTREE WATCHDOG PRM-11 SMARTMAILER TEN SPELL DECPoint of Sa JUICER PARALLEL Library V2 RTMULTI and Addons SMARTMAILER for RSTS/E CU FILTRA Spring 86 RT-1 SIG CP/M KERMIT S Invasion for PRO Bonner Labs APFELN RUNOFF VAX-LIB-3 VAX-LIB-2 IMAGE SPEL TURBOCOM FNDFIL PC-8088 Collection #10 VT200 TOOLKIT PLATOOLS SMARTMAILER DEPROC LaTe KERMIT-11 FANCY FONTS XMIT CU ReGis to HPGL CED International RUNITOFF JP5-JP6 FODT PASCAL-OS/ ANISMT TECO WPSIM DECSYSTEM-20 SIG Spring 85 CAMERA DELPHIN HACK BIBENTRY APFELN KERMI S DIGITIZING Acid Document Generator VAX-LIB-2 AMAR-1 AMAR-20 DATMAN/VAX IMAGE VT200 TOOLKI COMPRO EVENTS PC8088 Collection #9 TECO Cherry Tree Workstation Bookings System EXPORT Data Inputt Generator CMSBROWSE PERSONNEL INVENTORY MS/DOS COMMS Selection Electronic Grade Book CP/ KERMIT LaTex JUICER SPELL PORTACALC DPRINT DUNGEON MINC BUDGET BUG CALC C Langua System DPROC "DEP" DECENC DECmate II OS/278 DIAL DTC GAMMA-11 GDADL LISP for RSX-11 MEM KERMIT S VAX-LIB-6 SPICE 3A6 VT200 TOOLKIT RUNNOFF SPLICE SPY:RSX TCOPY SPELL VT-200 COMPR EVENTS CMSBROWSE UNDELETE DIAL BLOCKER SCAN CODER BITMAP DTC/PC ADDRESS BOO LaserWriter PORTACALC SPICE 3A6 PRO/Smart Mailer CBASIC2 Accts JP5-JP6 Payable/Receivable McGraw-Hi Payroll SEDT: EDT/WPS Screen CLNDRS:A Calendar Program INDEX AKCOUNT CORPHONE E-Systems Grab Ba RGT RDG PLTXSMB ICON DEVICS DATATRIEVE Library Collection CMSBROWSE EXPERT FPaint IMAG DBMS/Spreadsheet for MS/DOS AMAR-1 AMAR-20 RDIR/SQMAP PC-8088 Collection #11 UP TIME REPORTE RT-11 DUCM/DYC GRAF11 DEL DIR PLOT-10 IMAGE LIBED FSTATS MS/DOS TIC-TAC-TOE QIX VAX-LIB DATMAN/VAX EDTPlus SPICE2 TREEDUPL LISPEX MCLS TYPE_PLUS EXPORT DISK_USE FRAG EDTEX PORT LOCATOR TECO CHPLOT NANNY DIR11-W WATCHDOG INACTIVE ACCOUNTS IMGSPL ICE TEX EDITOR VAX-LIB-4 GRAPHIC UTILITIES SETAUX.ARC STATPK FIGure KERMIT Distribution TENBACKU JUICER VTEDIT 2022 VAX-LIB-3 VISTA EDITOR MTU TDE RSTSOPEN DRAWTREE WATCHDOG PRM-1 SMARTMAILER TEN SPELL DECPoint of Sale JUICER PARALLEL Library V2 RTMULTI and Ad SMARTMAILER for RSTS/E CU GRAPHKIT FILTRA Spring 86 RT-11 SIG CP/M KERMIT S Invasion for Bonner Labs RUNOFF VAX-LIB-3 VAX-LIB-2 IMAGE SPELL TURBOCOM FNDFIL PC-8088 Collection #10 V TOOLKIT PLATOOLS SMARTMAILER DEPROC LaTex KERMIT-11 FANCY FONTS XMIT MEMO ReGis to H CED International RUNITOFF JP5-JP6 FODT PASCAL-OS/8 ANISMT CODER WPSIM DECSYSTEM-20 SIG Sp 85 CAMERA DELPHIN HACK BIBENTRY APFELN REPORTER DIGITIZING Acid Document Generator VAX-LI AMAR-10 AMAR-20 DATMAN/VAX IMAGE VT200 TOOLKIT COMPRO EVENTS PC8088 Collection #9 TECO Cher Tree Workstation Bookings System EXPORT Data Inputter Generator CMSBROWSE PERSONNEL INVENTOR MS/DOS COMMS Selection Electronic Grade Book CP/M KERMIT LaTex JUICER SPELL PORTACALC DPRIN DUNGEON MINC BUDGET BUG CALC C Language System DPROC "DEP" DECENC DECmate II OS/278 DIA DTC GAMMA-11 GDADL LISP for RSX-11 MEMO PORTACALC VAX-LIB-6 SPICE 3A6 VT200 TOOLKI RUNNOFF SPLICE SPY:RSX TCOPY SPELL VT-200 COMPRO EVENTS CMSBROWSE UNDELETE DIA BLOCKER SCAN CODER BITMAP DTC/PC ADDRESS BOOK LaserWriter PORTACALC SPICE 3A6 PRO/Sma Mailer CBASIC2 Accts TREEDUPL Payable/Receivable McGraw-Hill Payroll SEDT: EDT/WPS Screen CLNDRS: Calendar Program INDEX AKCOUNT CORPHONE E-Systems Grab Ba RGT RDG PLTXSMB ICON DEVIC DATATRIEVE Library Collection CMSBROWSE EXPERT FPaint IMAGE DBMS/Spreadsheet for MS/DOS AMAR-1 AMAR-20 RDIR/SQMAP PC-8088 Collection #11 UP TIME REPORTER RT-11 DUCM/DYC GRAF11 DEL DIR PLO 10 IMAGE LIBED FSTATS MS/DOS TIC-TAC-TOE QIX VAX-LIB-5 DATMAN/VAX SPICE2 RT-11 DUCM/DYC G

DECUS

Program Library

SOFTWARE ABSTRACTS

"A Solution with a Future"

LIB

DECUS PROGRAM LIBRARY

These corrections are to be made to the 1988/1989 Software Catalog. **DECUS No. VAX-LIB-1, Title:** The VAX Library Collection 1, contains programs from VAX-1 through VAX-6, VAX-9, and VAX-12 through VAX-21.

DECUS No. VAX-LIB-2, Title: The VAX Library Collection 2, contains programs from VAX-22 through VAX-24, VAX-26 through VAX-33, VAX-37, VAX-39, VAX-41, VAX-43 throughVAX-45, VAX-47, VAX-48, VAX-51 through VAX-54, VAX-57 through VAX-61, and VAX-63 through VAX-74.

DECUS No. 11-829, Title: KERMIT-11 for P/OS and Micro/RSX, delete the paragraph, "Associated Documentation:......".

DECUS No. 11-830, Title: KERMIT-11 for Micro/RSTS/E and RT-11, delete the paragraph, "Associated Documentation:.........".

NEW LIBRARY PROGRAMS AVAILABLE FOR THE VAX/VMS FAMILY OF COMPUTERS

DECUS No: VAX-352 Title: MENU **Version:** 2.1, April 1988

Submitted by: Heino Bruecher, Feldmuehle AG Werk Reisholz

Operating System: VAX/VMS V4.6 **Source Language:** PASCALx**Hardware Required:** Video terminal supported by Screen Management Facility **Keywords:** Menu Control, Tools – Applications Development

Abstract: The program allows a user to execute program images, DCL command procedures,batch jobs, or DCL commands. The desired action is performed by selection of a key from the menu presented on the terminal. The selection can be done by using the up and down arrow keys or by pressing a number key with the number of the item. Actions can be performed by means of subprocesses (returns to the menu when the action is over) or by execution in the same process (MENU exits before starting the action). The menu bases on one or more text files. A menu can also have submenus (recursive algorithm). Command lines can take up to nine variable substitutions, prompts can be specified in the menu file. Based on qualifiers the menu can be made to exit due to timeout and/or to force the user to be logged off when it exits.

Notes: Operating system VAX/VMS V4.4 or higher is required.

Media (Service Charge Code): 600' Magnetic Tape (MA) **Format:** VMS/BACKUP

DECUS No: VAX-353 Title: WINDOW **Version:** 1.0, July 1988

Submitted by: Joachim Bromet, University of California at Davis, Davis, CA

Operating System: VAX/VMS V4.6 **Source Language:** FORTRAN 77 **Keywords:** Menu Control

Abstract: WINDOW is an interactive menu-driven program by which certain users may issue DCL commands on all VMS nodes via maneuverable windows without leaving the utility. Since heavy use is made of screen management routines, this program will only run using VT52, VT100 and VT200 terminals. It allows the presentation of choices in a pulldown menu format.

WINDOW is an integrated package of routines that chooses a VMS node, allows windows to be positioned (left, right, up and down), increases or decreases the window size two dimensionally and provides an on-line help library.

Notes: Operating System VAX/VMS V4.X or higher is required.

Media (Service Charge Code): User's Manual (EA), 2400' Magnetic Tape (PA) **Format:** VMS/BACKUP

DECUS No: VAX-354 Title: LJ250 DEColorwriter Demonstration Package **Version:** May 1988

Submitted by: Digital Equipment Corporation

Operating System: VAX/VMS **Hardware Required:** LJ250/LJ252 Companion Color Printer. **Keywords:** Graphics

Abstract: The files in this package demonstrate the capability of the LJ250/LJ252 Companion Color Printer to print color images from a sixel file. These demonstration files have different images such as birds, boats, street scenes, etc.

Notes: Operating System VAX/VMS V4.2 or higher is required.

Media (Service Charge Code): 600' Magnetic Tape (MA) **Format:** VMS/BACKUP

DECUS No: VAX-355 Title: CHOPS: Call Handling for Operations **Version:** 2.0, July 1988

Submitted by: Digital Equipment Corporation

Operating System: MicroVMS V4.5, VAX/VMS V4.5 **Source Language:** PASCAL **Software Required:** VAX TDMS V1.6 or higher is required. VAX DECgraph V1.5 is optional. **Keywords:** Utilities – VMS

Abstract: CHOPS is a call handling tool which was originally designed to help to improve Information Call Handling activities.

Its functionality is based on user requirements from IS Operation Support group and Application Development Support group.

In addition to that, CHOPS can take advantage of the experience and usage of other Call Handling Systems.

CHOPS main qualities are simplicity, performance and flexibility. It allows the Operation Secretary (or Call Handling desk) to follow various calls through different stages such as logging, closing, assignment, escalation or transfer. Various display, list, reports are available.

CHOPS keeps users' and callers' informations as well as skills and supported products. Those informations are easy to maintain and report.

CHOPS uses a "Queue Logic" to log a call, that is, calls can be stored into a public queue and then dispatched to appropriate expert, or calls can be allocated to a public queue as well as a

"Product Queues" allocated to some experts.

Notes: Operating System VAX/VMS V4.5 or higher is required.

Media (Service Charge Code): 600' Magnetic Tape (MC) **Format:** VMS/BACKUP

DECUS No: VAX-356 Title: LATUSER **Version:** 2.0, August 1988

Submitted by: Richard E. Cox, Jr., Kollsman, Merrimack, NH

Operating System: MicroVMS V4.6, VAX/VMS V4.7 **Source Language:** MACRO-32, VAX FORTRAN **Keywords:** Networking, System Management – VMS, Utilities – VMS

Abstract: Like "show user", LATUSER displays the terminal name, username and process identification (PID). However, LATUSER also displays the LAT terminal server and the terminal server port of all interactive users on the system.

System Managers, — do you have a problem with a terminal and the LAT terminal number will not do? LATUSER gives you the server and port it is attached to. Do you have to reboot the server? LATUSER can sort its output by server name, grouping all users on the same server together; now you know who is using that server. Need to know who is logged in from another node? LATUSER will display the remote user and node name where that user is logged in from.

LATUSER can sort the output by various fields, or direct the output to a file.

Media (Service Charge Code): 600' Magnetic Tape (MA) **Format:** VMS/BACKUP

DECUS No: VAX-357 Title: REMINDER **Version:** 2.0, August 1988

Submitted by: Richard E. Cox, Jr., Kollsman, Merrimack, NH

Operating System: MicroVMS V4.6, VAX/VMS V4.7 **Source Language:** VAX FORTRAN **Keywords:** Calendars, Scheduling, Utilities - VMS

Abstract: This package is used to send messages to one's self, to users with the same UIC, or to users with the same username up to an underscore.

It is not one of those programs that just displays information at login or whenever you request it. It actually sends your message to you at the time you tell the message to be sent. If you have a meeting at 10:30, this package will remind you at 10:30 even if you have logged in at 8:00. If you are not logged in when a reminder message is scheduled to be sent, it will send that message to you when you do log in; therefore, you never loose a message.

This package will continue to send a reminder message until the message has been acknowledged, or expired. The time delay interval used by this package continues to double starting at one minute until it has reached twenty-four hours. After a twenty-four hour period has been reached, a reminder message will be issued each day until the message expires. Reminder messages, by default, expire one week after the first scheduled broadcast.

Media (Service Charge Code): 600' Magnetic Tape (MA) **Format:** VMS/BACKUP

DECUS No: VAX-358 Title: MAINT **Version:** 1, August 1988

Submitted by: Leonard J. Peirce, Western Michigan Univ. Academic Comp Ctr, Kalamazoo, MI

Operating System: MicroVMS V4.5, VAX/VMS V4.5 **Source Language:** C **Memory Required:** 107KB **Keywords:** File Management, Utilities –VMS

Abstract: MAINT is a full-screen Directory/File Maintenance utility. Directories are presented to the user in a series of one or more screens, allowing the user to work with an entire directory at one time instead of working with a few files and having to do a DIRECT to see the current state of the directory. Run-time switches provide the user the opportunity to tailor what information is included on the screen and the option of including user-defined extended textual descriptors for individual files/directories.

The following functions are available in MAINT:

. Delete files/directories.
. Copy files.
. Rename files/directories.
. Protect files/directories.

. Edit an Access Control List (ACL) for a file/directory.
. List a file's contents to the screen.
. Change to a subdirectory.
. Get full directory information on a file.
. Search for a specific file in a directory.
. Suspend MAINT and return to DCL level, either indefinitely or just to execute one command.
. Create/access extended textual descriptors for files and directories.
. Access on-line help.

By combining the above capabilities with a full-screen interface and some added functionality, the user can work with entire directory structures quickly, easily, and efficiently just by pressing a few keys.

One very important feature of MAINT is that execution of the operations on files is NOT done until you tell it to go ahead and perform them. In other words, you can work with all of the files, specifying the operations, and then tell MAINT to execute them all at once. This means that you have time to change your mind and perhaps undo the operations on one or more of the files.

Media (Service Charge Code): 600' Magnetic Tape (MA) **Format:** VMS/BACKUP

DECUS No: VAX-359 Title: CDUTIL Version: 1.0, July 1988

Submitted by: John T. Carroll III, Columbus, IN

Operating System: MicroVMS V4.6 **Source Language:** VAX FORTRAN **Keywords:** File Management, Utilities – VMS

Abstract: CDUTIL is a FORTRAN program that performs text file compression and decompression operations. The compression algorithm that is employed is most effective when long strings of repeated characters are present.

Once invoked, CDUTIL prompts the user to request (C)ompression, (D)ecompression, or (E)xit. Either of the first two selections generate additional prompts for input and output files. The requested operation is then performed without further operator intervention and several lines of summarizing information are displayed. Any number of compression and decompression operations can be performed before exiting the program.

Media (Service Charge Code): One RX50 Diskette (JA) **Format:** VAX/ANSI, 600' Magnetic Tape (MA) **Format:** VAX/ANSI

NEW LIBRARY PROGRAMS AVAILABLE FOR THE PDP-11 COMPUTER FAMILY

DECUS No: 11-910 Title: MAIL Version: 1.14, January 1988

Submitted by: Mike Marak, Concordia Univ., EMC Lab., Loyola Campus, Montreal, Quebec, Canada H4B 1R6

Operating System: RT-11 V5.4, TSX-PLUS V6.2 **Source Language:** FORTRAN IV **Memory Required:** 32KB **Software Required:** FORTRAN IV **Keywords:** Data Communications, Mail, System Management – RT-11, Utilities – Terminal

Abstract: MAIL is a message handling system for use under TSX-PLUS. It allows users registered with the mail system to read their messages or send messages to other registered users. The messages are stored in a file, and users can only read messages that are sent to them.

The creation of the mail file and registering users is done by the POSTMN.TSX program.

Messages are limited to 1000 bytes maximum, and each user has a total of 2500 bytes of message storage.

Documentation is included, as well as a log of a sample session, and a command file to build the mail system. A pre-built mail system is also included, having the mail file as SY:MAIL.XXX.

Notes: Operating system RT-11 V5.4 or operating system TSX V6.2 is required because system calls are required.

Media (Service Charge Code): One RX01 Diskette (KA) **Format:** RT-11, 600' Magnetic Tape (MA) **Format:** RT-11

DECUS No: 11-911 Title: VSET Version: 1.2, August 1988

Submitted by: John M. Crowell, Multiware, Inc.

Operating System: RT-11 V5.4 **Source Language:** MACRO-11 **Memory Required:** 16KB **Keywords:** Device Handlers

Abstract: VSET performs SET options on RT-11 device handler files. The handler's SET code is executed as if a normal SET command had been issued, but the handler file need not be that of a currently installed device, and need not have the .SYS extension. It may reside on a disk other than the system disk. VSET will also, optionally, display all the possible SET options of a handler.

Notes: Operating System RT-11 V5.4 or later is required.

Media (Service Charge Code): One RX50 Diskette (JA) **Format:** RT-11, 600' Magnetic Tape (MA) **Format:** RT-11

NEW LIBRARY PROGRAMS AVAILABLE FOR THE CP/M PERSONAL COMPUTER FAMILY

DECUS No: CPM-273 Title: Vancouver Area Rainbow Users Group Newsletters **Version:** 1/87 through 8/88

Submitted by: David P. Maroun, Chilliwack, B.C., Canada V2P 6C5

Operating System: CP/M-86/80 **Source Language:** ASSEMBLY, BASIC, PASCAL **Memory Required:** 64KB **Keywords:** Utilities — CP/M

Abstract: This package contains much information of general interest, and are read in various parts of the North American continent. The newsletters contain a number of programs in ASSEMBLY, BASIC, and PASCAL languages, reviews of software and hardware and answers to readers' questions on computer problems.

The newsletters are in ASCII form but archived to save space. A de-archiving program is provided, as well as a program to aid viewing on the screen. Documentation for these programs is included.

Notes: The newsletters are in archived format. The de-archiving and viewing programs supplied are designed for CP/M-80.

Media (Service Charge Code): One RX50 Diskette (JA)

REVISIONS TO LIBRARY PROGRAMS

DECUS No: VAX-288 Title: REPORT WRITER Version: 1.1, July 1988

Submitted by: David Cohen, Security Pacific Automation Company, Los Angeles, CA

Operating System: VAX/VMS V4.5 **Source Language:** DCL, VAX COBOL **Keywords:** Tools – Applications Development

Abstract: REPORT WRITER generates a COBOL program, using as input four user-supplied files which define the report and the data file record. Handles up to eight levels of control breaks, with totals available for each level. Each control group can have the following options:

. "At Top of Control Group"
. "At Bottom of Control Group"
. "At Top of Page"
. "At Bottom of Report"
. "New Page" (All quoted terms in this abstract have the same meaning as in DATATRIEVE). Grand totals and "At Bottom of Report" are in addition to the eight allowable control breaks. Report column positions are computed automatically, from Layout Chart created by the user, in any editor. Output program can be edited and modified, if desired.

Notes: Operating System VAX/VMS V4.0 or later is required because file names are greater than nine characters in length.

Changes and Improvements: Additional control breaks, error handling and bug fixes.

Media (Service Charge Code): 600' Magnetic Tape (MA) **Format:** VMS/BACKUP, or order VAX-LIB-8

STEERING COMMITTEE LISTS

ARTIFICIAL INTELLIGENCE SIG

CHAIR
Cheryl Jalbert
JCC
128 West Broadway
Granville, OH 43023
(614) 587-0157

VICE-CHAIR
OPS5 WORKING GROUP CHAIR
Don Rosenthal
Space Telescope Science Inst.
Homewood Campus
Baltimore, MD 21218
(301) 338-4844

NEWSLETTER TASK FORCE CHAIR
ADMINISTRATIVE ASSISTANCE
Becky Wise
Amdahl CSD
2200 North Greenville Ave.
Richardson, TX 75081
(214) 699-9500 x 272

NEWSLETTER EDITOR
Terry Shannon
Computer Info. Sys., Inc.
Technical Consultant
165 Bay State Drive
Braintree, MA 02184
(617) 848-7515

SYMPOSIA COORDINATOR
Pam Vavra
Hughes Aircraft EDSG
P.O. Box 902 E52/D220
El Segundo, CA 90245-0902
(213) 616-7071

MEMBERSHIP COORDINATOR
SUITE COORDINATOR
Chris Goddard
Simpact Associates
9210 Skypark Court
San Diego, CA 92123
(619) 565-1865

SESSION NOTE EDITOR
George Humfeld
Naval Sea Systems Command
PMS 350 ED Dept of the Navy
Washington, DC 20362-5101
(202) 692-0137

ASS'T SESSION NOTES EDITOR
David Frydenlund

STORE REPRESENTATIVE
Sally Townsend
Inst. Defense Analysis
1801 N. Beauregard St.
Alexandria, VA 22311
(703) 845-2122

PUBLIC DOMAIN SOFTWARE TF CHAIR
LIBRARY REPRESENTATIVE
Jim Sims

Space Telescope Science Ins.
3700 San Martin Drive
Baltimore, MD 21218
(301) 338-4949

AI LUG COORDINATOR
ASSISTANT STORE REP.
Dennis Clark

RT2 Box 264
Kingston, TN 37763
(615) 576-7384

REPORTER TO THE UPDATE.DAILY
Bill Lennon

SEMINAR UNIT REP.
CAMPGROUND COORDINATOR
Leona Fluck
Educational Testing Service
Rosedale Road
Princeton, NJ 08540
(609) 734-1243

DEC COUNTERPART
Art Beane
Hudson, MA

MEMBERS-AT-LARGE
David Slater
George Winkler
Jeff Fox
John Williamson
Wayne Graves
Matt Mathews
Dave Campbell
Shirley Bockstahler-Brandt
Barry Breen
Tom Viana

BUSINESS APPLICATIONS SIG

CHAIRMAN
George Dyer
Gallaudet University
800 Florida Ave, NE-EMG Bldg
Washington, DC 20002
(202) 651-5300

COMMUNICATIONS COORDINATOR
Steve Lacativa
Price Waterhouse
153 East 53rd Street
New York, NY 10022
(212) 371-2000 x 3107

SYMPOSIA COORDINATOR
Mark Hults
USSA Administrative Systems
USSA Bldg. B01E
San Antonio, TX 78288
(512) 498-8725

LUG COORDINATOR
Patrick LeSesne
U.S. Coast Guard
Room 1416E 2100 2nd St. SW
Washington, DC 20593
(202) 267-0354

MARKETING COORDINATOR
Tom Byrne
L. Karp & Sons
1301 Estes
Elk Grove Village, IL 60007
(312) 593-5706

PROGRAM PLANNING COORDINATOR
Stuart Lewis
Douglas Furniture Corp.
P.O. Box 97
Bedford Park, IL 60499
(312) 458-1505

SEMINARS COORDINATOR
Daniel Esbensen
Touch Technologies, Inc.
9990 Mesa Rm, Rd. #220
San Diego, CA 92121
(619) 455-7404

LRP COORDINATOR
Arnold I. Epstein
D-M Computer Consultants
Rolling Meadows, IL 60008
(312) 394-8889

NEWSLETTER EDITOR
Dave Levenberg
Credit Suisse
Dept OA1 15th floor
100 Wall Street
New York, NY 10005
(212) 612-8372

SESSION NOTE EDITOR
Marty Schmitt
Harris Publishing
3 Barker Avenue
White Plains, NY 10601
(914) 946-7500 x 287

LIBRARY REPRESENTATIVE
David Hittner
Projects Unlimited
3680 Wyse Road
Dayton, OH 45414
(513) 890-1800

CL SIG LIAISON
Becky Burkes-Ham

DMS SIG LIAISON Joe Sciuto MEMBERS-AT-LARGE
Robert D. Lazenby
Dixie Beer Dist., Inc.
Louisville, KY

Robert Kayne
Gallaudet College
Washington, DC

Ray Evanson
Paragon Data Systems
Winona, MN

DEC COUNTERPARTS
Sue Yarger
Digital Equipment Corporation
Merrimack, NH 03054-0430

Paula Daley
Digital Equipment Corporation
Merrimack, NH 03054-0430

Pam Kukla
Digital Equipment Corporation
Maynard, MA 01754

Wombat Magic

DATATRIEVE/4GL SIG

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

SYMPOSIA COORDINATOR
Bernadette reynolds
City of Ontario Police
200 N. Cherry Ave.
Ontario, CA 91764
(714) 988-6481

ASS'T SYMPOSIA REPRESENTATIVES
T.C. Wool
E.I. duPont DeNemours & Co.
Engineering Dept.
P.O. BOX 6090.
Newark, DE 19714-6090
(302) 366-4610

Janet G. Banks
Weyerhaeuser Info. Sys.
Mail Stop CCB-2E
Tacoma, WA 98477
(206) 924-4082

COMMUNICATIONS REPRESENTATIVE
NEWSLETTER EDITOR
   Joe H. Gallagher
   Research Medical Center
   2316 East Meyer Blvd.
   Kansas City, MO 64132
   (816) 276-4235

COMMUNICATION REPRESENTATIVE
PRODUCTION EDITOR
   Steve Cordiviola
   Kentucky Geological Survey
   228 Mining & Mineral Resources Bldg.
   Lexington, KY 40506-0107
   (606) 257-5863

ASSOCIATE NEWSLETTER EDITOR
   Pasquale (Pat) F. Scopelliti
   Corning Glass Works
   Mail Stop MP-RO-01-1
   Corning, New York 14831
   (607) 974-4496

   Lorey B. Kimmel
   Digital Equipment Corp.
   6707 Whitestone Road
   Baltimore, MD 21207
   (301) 298-1500

   Herbert G. Reines
   Reznick Feddder & Silverman
   4520 East West Highway
   Suite 300
   Bethesda, MD 20814
   (301) 652-9100

   Alan Winston
   Stanford Synchrotron Radiation Lab.
   SLAL BIN 69
   P.O. Box 4349
   Stanford, CA 94305
   (415) 854-3300 x2874

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

ASSISTANT VOLUNTEER COORD.
   Harry Miller
   City of Ontario Police
   200 N. Cherry Avenue
   Ontario, CA 91764
   (714) 988-6481

SEMINARS COMMITTEE REP.
   Dana Schwartz
   15719 Millbrook Lane
   Laurel, MD 20707
   (301) 859-6277

SESSION NOTES EDITOR
   Mary E. Roberts
   City of Ontario Police Dept.
   200 N. Cherry Avenue
   Ontario, CA 91764
   (714) 988-6481

SUITE COORDINATOR
   Bert Roseberry
   Commandant (G-APA-1)
   2100 2nd Street, S.W.
   Washington, DC 20593-0001
   (202) 267-2629

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

DEC COUNTERPARTS
   John Carpenter
   110 Spit Brook Road
   2K02-2/Q21
   Nashua, NH 03062
   (605) 881-0043

ARTIST & LIBRARY REP.
   Bart Z. Lederman
   WU World Communications
   67 Broad Street (28th Floor)
   New York, NY 10004
   (212) 607-2657

RALLY WORKING GROUP CHAIR
   Steven G. Fredrickson
   Fredrickson Consulting Service
   107 1st Avenue N.
   Seattle, WA 98109
   (206)283-0273

POWERHOUSE W/G CHAIR
   David Nagurney
   TSO Financial Corp.
   Five TSO Financial Center
   Three Hundred Welsh Road
   Horsham, PA 19044-2009
   (215) 657-4000

DMS SIG LIAISON
   William Tabor
   W.I. Tabor, Inc.
   12018 Royal Palm Blvd.
   Coral Springs, FL 33065
   (305) 755-7895

SMARTSTAR WORKING GROUP CHAIR
   Thomas Colati
   Time Enterprises
   301 North Harrison
   Suite 101
   Princeton, NJ 08540
   (800) 548-6865

ACCENT-R USER GROUP LIAISON
   Winston Tellis
   Fairfield University
   North Benson Road
   Fairfield, CT 06430
   (203) 255-5411

FOCUS WORKING GROUP CHAIR
   Les Hulse
   The Gillette Company
   Prudential Tower Bldg.
   Boston, MA 02199
   (617) 421-7910

DAARC SIG

CHAIRMAN
   James Deck
   Inland Steel Research Lab.
   3001 East Columbus Drive
   East Chicago, IL 46312
   (219) 392-5613

SYMPOSIA COORDINATOR
   Mack Overton
   FDA
   Chicago, IL

COMMUNICATIONS REPRESENTATIVE
NEWSLETTER EDITOR
   Dale Hutchison
   Cummins Engine Co.
   4720 Baker St., Ext.
   Lakewood, NY 14750
   (716) 456-2191

DEC COUNTERPART
   Bill Forbes
   Marlboro, MA

HARDWARE & INTERFACING
   Peter Clout
   Los Alamos National Lab
   Los Alamos, NM

MATH STATISTICS & ANALYSIS
   Herbert J. Gould
   C.C.F.A. Univ. of Ill. Medical Ctr.
   Chicago, IL

PROCESS CONTROL-INDUSTRIAL AUTOMATION
   Bill Tippie
   Kinetic Systems Corp.
   Lockport, IL

RS-1
   George Winkler
   CPC International
   Argo, IL

DATA MANAGEMENT SYSTEMS SIG

CHAIRMAN
   Doug Dickey
   GTE Government Systems
   1700 Research Blvd.
   Rockville, MD 20850
   (301) 294-8462

MEMBER AT LARGE
   Alan Schultz
   Southwestern Bell Yellow Pages
   12800 Publication Dr., Suite 108
   St. Louis, MO 63131
   (314) 957-2029

SYMPOSIA COORDINATOR
SQL Standards Rep.
   Keith Hare
   JCC
   P.O. Box 463
   Granville, OH 43023
   (614) 587-0157

COMMUNICATIONS REP.
   Debbie Kennedy Coleman
   Shane Co.
   2 W Washington St., Suite 600
   Indianapolis, IN 46204
   (317) 635-9100

NEWSLETTER EDITOR
   William Packard
   Mass Mutual Life Ins.
   1296 State Street B-391
   Springfield, MA)1111
   (413) 788-8411

SESSION NOTE EDITOR
   Mark Morgan
   Farm Credit Banks
   P.O. Box 141
   Springfield, MA 01102
   (413) 786-7600

MEMBERSHIP COORDINATOR
MEMBER AT LARGE
   Rocky Hayden
   Userware International
   2235 Meyer Avenue
   Escondido, CA
   (619) 745-6006

MEMBER AT LARGE
PAST SIG CHAIRMAN
   Steve Pacheco
   Ship Analytics
   North Stonington, CT 06359
   (203) 535-3092

PAST SIG CHAIR
MEMBER AT LARGE
   Joe Sciuto
   U.S. Army
   5910 Westchester Street
   Alexandria, VA 22310
   (202) 692-6903

SEMINAR REP.
Stephen Gomez
Signal Technology
1700 Montgomery St.
San Francisco, CA 94111
(415) 954-8532

WORKING GROUP COORDINATOR/
Jim Perkins
PSC, Inc.
20 Kimball Ave., Suite 305
Shelburne, VT 05401
(802) 863-8825

CAMPGROUND COORDINATOR
Rosemary O'Mahony
Arthur Anderson & co.
33 West Monroe Street
Chicago, IL 60603
(312) 507-6510

SESSION CHAIR COORDINATOR
Andy Menezes
AD & E
29-B Montvale Avenue
Woburn, MA 01801
(617) 938-1979

Rdb WORKING GROUP Coordinator
Howard Cheng
Bechtel Western Power Corp.
12440 East Imperial Highway
Norwalk, CA 90650
(213) 807-4077

ROADMAP COORDINATOR
Elizabeth Irving
Dupont
P.O. Box 1089
Orange, Texas 77630-1089
(409) 886-9427

DBMS COORDINATORS
Bryan Holland
1006 Pleasant St., #20
Weymouth, MA 02189
(617) 335-7573

Paul E. Reese
Aetna, Systems Dept
Financial Division
CityPlace
Hartford, Connecticut 06156
(203) 275-2012

MEMBER AT LARGE
Jim Redfield
BDM
9020 N. Cap. of Texas Highway
Austin, TX 78759
(512) 346-6760

STORE REPRESENTATIVE
FIMS STANDARDS REP.
Paul W. Plum, Jr
Lukens Steel Company
Coatesville, PA 19320
(215) 383-2024

RMS WORKING GROUP COORDINATOR
Allen Jay Bennett
Logisticon, Inc.
5035 Whitneyville Road
Ada, MI 49301
(616) 452-1823

DEC COUNTERPART
Wendy Herman
John Wood
Nashua, NH

EDUSIG
CHAIRMAN
Robert A.Shive, Jr.
Associate Dean of the College
Millsaps College
Jackson, MS 39210-0001
(601) 354-5201

VICE CHAIR AND UNIV. COORD.
Ardoth A. Hassler
Assistant Director
for Academic Services
Computer Center
Catholic University of America
Washington, DC 20064
(202) 635-5373
HASSLER@CUA.BITNET

SYMPOSIUM COORDINATOR
Mary Jac Reed
VAX Project Manager
Office of Instructional Tech.
University of Delaware
Newark, DE 19716
(302) 451-8161

COMMUNICATIONS COORDINATOR
Paula Barnes
Computer Center Operations Manager
North Carolina School of Science & Math.
1219 Broad Street
P.O. Box 2418
Durham NC 27705
(919) 286-3366
plb@ecsvax.bitnet

SEMINARS COORDINATOR
Donald C. Fuhr
Director of Computer Services
Tuskegee University
Tuskegee, AL 36088
(205) 727-8242

NEWSLETTER EDITOR
Fred Bell
Professor of Computer Science and
Coordinator of Computer Based Education
Taft College
29 Emmons Park Drive
P.O. Box 1437
Taft, CA 93268
(805) 763-4282

SESSION NOTES EDITOR
Jim Gerland
SUNY/Buffalo
Academic Computing, Univ. Computing Serv.
Computer Center
Buffalo, NY 14260
(716) 636-3557
gerland@ubvms.bitnet
gerland@ubvmsc.cc.buffalo.edu

ADMINISTRATIVE APPLICATIONS COORD.
David Cothrun
President
Taft College
29 Emmons Park Drive
P.O. Box 1437
Taft, CA 93268
(805) 763-4282

DIGITAL COUNTERPART
C. Michael Greene, Jr.
Networking Consultant
Education Industry Marketing Group
Digital Equipment Corporation
Three Results Way, MR03-2/E7, Box 1003
Marlboro, MA 01752-9103
(508) 467-2149
GREENE%SANTEE.DEC@DECWRL
GREENE%SANTEE.DEC@DECWRL.DEC.COM
GREENE@SANTEE.DEC.COM

GRAPHICS APPLICATIONS SIG
CHAIRPERSON
Bijoy Misra
Smithsonian Institution
60 Garden Street, MS39
Cambridge, MA 02138
(617) 495-7392
BIJOY@CFA2.BITNET

SESSION NOTE EDITOR
IMAGE PROCESSING WORKING GROUP CHAIR
Dr. Robert Goldstein
Eye Research Institute
20 Staniford Street
Boston, MA 02114
(617) 742-3140
GOLDSTEIN%CDV.DECNET@MGHCCC.HARVARD.EDU

SYMPOSIUM REPRESENTATIVE
HARDCOPY WORKING GROUP CHAIR
Henry Schneiker
P.O. Box 42767
Tucson, Arizona 85733
(602) 325-5884

STORE REPRESENTATIVE
Stephen Schultz
Rochester Institute of Technology
Center for Imaging Science
One Lomb Memorial Drive
Rochester, NY 14623
BITNET: sls4255@ritvax
USENET: harvard!rochester!ritcv!ulta!sls4255

LIBRARY REPRESENTATIVE
Robert Krieg
Uphjohn Company
MS: 7266-267-1
301 Henrietta Street
Kalamazoo, MI 49001
(616) 385-7563

COMMUNICATIONS REPRESENTATIVE
NEWSLETTER EDITOR
Robert Hays
P.O. Box 1567
3621 South State Road
Ann Arbor, MI 48106
(313) 769-8500

ASSOCIATE NEWSLETTER EDITOR
(Open)

STANDARDS COORDINATOR
Paul Waterstraat
Multiware, Inc.
2121 Second Street, Bldg B, Suite 107
Davis, CA 95616
(916) 756-3291

INFORMATION OFFICER
Bill Kramer
NASA Ames Research Center
NAS Systems Division MS 258-6
Moffett Field, CA 94035
(415) 694-4418

ENGINEERING GRAPHICS WORKING GROUP CHAIR
Daniel Land
John Fluke Mfg. Co., Inc.
Mail Stop 221B
P.O. Box C9090
Everett, WA 98206
(206) 356-5257

HUMAN FACTORS/WINDOWS WORKING GROUP CHAIR
Dottie Jo Elliot
Northrup Services, Inc.
P.O. Box 12313
Research Triangle Park, NC 27709
(919) 541-1300

WORKSTATION WORKING GROUP CHAIR
Jim Sims
Space Telescope Science Institute
3700 San Martin Drive
Baltimore, MD 21218
(301) 338-4949

MEMBERS-AT-LARGE
Mike York
Boeing Computer Services
P.O. Box 24346 M/S 03-73
Seattle, WA 98124
(206) 342-1442

Mike McPherson
A.H. Case Center for CAEM
Michigan State University
236 Engineering Blvd.
East Lansing, MI 48824
(517) 353-9769

Pam Vavra
Lysander Solutions
P.O. Box 3368
Manhattan Beach, CA 90266

DEC COUNTERPARTS
Jim Flatten
Spit Brook, NH

Rick Landau
Marlboro, MA

HARD NEWS

HARDWARE MICRO SIG

CHAIRMAN
Willian K. Walker
Monsanto Research Corp.
Miamisburg, OH

PRODUCT PLANNING COORDINATOR
George Hamma
Synergistic Technology
Cupertino, CA

PRE-SYMPOSIUM SEMINAR COORDINATOR
James R. Lindesmith
Monsanto Research Corp.
Miamisburg, OH

COMMUNICATIONS COORDINATOR
John G. Hayes
Information Systems
South Central Bell
Birmingham, AL

NEWSLETTER EDITOR
Carmen D. Wiseman
Digital Review
Boston, MA

SESSION NOTE EDITOR
DAARC SIG LIAISON
Bill Tippie
Kinetic Systems Corp.
Lockport, IL

STANDARDS COORDINATOR
CAMAC WORKING GROUP COORDINATOR
Peter Clout
Los Alamos National Lab
los Alamos, NM

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

TOEM (CHIPS & BOARDS)
Jack J. Peterson
Horizon Data Systems
Richmond, VA

HHK (HARDWARE HINTS & KINKS)
Wayne Kesling
Monsanto Research Cor.
Miamisburg. OH

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

PERFORMANCE MEASUREMENT COORD.
William Wallace
600 W. Washington Street
Peoria, IL

CSS COORDINATOR
Pratap Gohel
E.I. duPont
Ingleside, TX

NETWORKS SIG LIAISON
Sandra Traylor
Target Systems
Yorba Linda, CA

VAX SIG LIAISON
Dave Schmidt
5100 Centre Avenue
Pittsburgh, PA

UNISIG LIAISON
Jim Livingston
1 Results Way
Cupertino, CA

SITE SIG LIAISON
Emily Kitchen
A.H. Robins Co.
Richmond, VA

RT-11 SIG LIAISON
Gary Sallee
Sallee Software Consulting
yorba Linda, CA

RSX SIG LIAISON
Hans Jung
Associated Press
New York, NY

MEMBERS-AT-LARGE
Mike Rembis
American Dade
Costa Mesa, CA

Hans Dahlke
Richland, WA

Jim Cutler
EDS Tower
16533 Evergreen
Southfield, MI

DEC COUNTERPARTS TERMINALS
Nina Abramson
Maynard, MA

TOEM (Chips & Boards)
Art Bigler
Marlboro, MA

DIAGNOSTIC
George D. Cooke
Maynard, MA

STORAGE
Marilyn Fedele
Maynard, MA

MSD (Micro Systems Develp.)
Roy Rodgers
Maynard, MA

PRINTER PRODUCTS
Frank Orlando
Maynard, MA

DECUS EUROPE LIAISON
Hans Zoller

IAS SIG

Alan Frisbie
Flying Disk Systems, Inc.
4759 Round Top Drive
Los Angeles, CA 90065
(213) 256-2575

NEWSLETTER EDITOR
Frank R. Borger
Radiation Therapy
Michael Reese Medical Center
Lake Shore Drive @ 31st St.
Chicago, IL 60616
(312) 791-2515

WHIMS COORDINATOR
Kathleen Anderson
Eaton Information Management
System Division
Hampton, VA
(804) 326-1941

RSX LIAISON
Ray French
Boeing Computer Services
Seattle, WA
(206) 655-6228

MEMBER-AT-LARGE
Doug Reno
Abbot Laboratories
North Chicago, IL
(312) 937-7504

PAST CHAIRMAN
Mike Robitaille
Grumman - CTEC, Inc.
6862 Elm Street
McLean VA 22101
(703) 556-7400 x431

CHAIRMAN EMERITUS
Bob Curley
Division of Medical Physics
University of Pennsylvania
Philadelphia, PA
(215) 662-3083

SYMPOSIA COORDINATOR
Lynda L. Roenicke
Special Systems Branch
Naval Ocean Systems Center
271 Cataline Blvd Code 742
San Diego, CA
(619) 225-7569

LIBRARY COORDINATOR
Ted Smith
The University of PA
Philadelphia, PA 19101
(215) 662-3500

MEMBER-AT-LARGE
Kerry Wyckoff
Salt Lake City, UT

DEC COUNTERPART
Nancyfaye Autenzio
Stow, MA
(617) 496-9606

LANGUAGES AND TOOLS SIG

CHAIRMAN
Sam Whidden
American Mathematical Society
201 Charles St.
P.O. Box 6248
Providence, RI 02940
(401) 272-9500

VICE CHAIR
SYMPOSIA COORDINATOR
Earl Cory
Eaton Corporation
31717 La Tienda Dr.
Westlake Village, CA 91359
(818) 706-5385

STORE REPRESENTATIVE
Chair, TECH. PROD OF DOC. W/G
Howard Holcombe
RCA
Front & Cooper Sts.
Camden, NJ 08055
(609) 338-4946

NEWSLETTER EDITOR
ALT. CommComm REP.
Al Folsom, Jr.
Fischer & Porter Co.
E. County Line Rd.
Warminster, PA 18974
(215) 674-7154

SESSION NOTES EDITOR
Mark Katz
GTE Government Systems
100 First Avenue
Waltham, MA 02154
(617) 466-3437

AUSTRALIAN L&T INTERFACE
Gordon Brimble
Bldg. 180 Labs Area
Defence Research Centre
Box 2151 GPO
Adelaide, S.A. Australia 5001
(61)(8)259-6119

INTERSIG COORDINATOR
Dorothy Geiger
Wollongong Logistics Group
49 Showers Drive #451
Mountain View, CA 94040
(415) 948-1003
(415) 962-7160

EUROPEAN METHODS
L&T INTERFACE
Bernd Gliss
Max-Planck-Institute
Heisenbergstra Be 1
7000 Stuttgart 80, W. Germany
(711) 686-0251

PAST CHAIR
PRODUCTIVITY TOOLS COORDINATOR
Kathy Hornbach
Digital Equipment Corporation
110 Spit Brook Rd., ZK03-3/Y25
Nashua, NH 03062

CHAIR TECO WORKING GROUP
Mark J. Hyde
Advanced Computing Services
209 Ardsley Drive
DeWitt, NY 13214
(315) 446-7223

MEMBER, ANSI BASIC X3J2 STDS. COMM.
STANDARDS COORD.
PDP-11 REP.
CHAIR, PDP-11 LAYERED PRODUCT W/G
ACTING SYMPOSIUM COORDINATOR
Stephen C. Jackson
SCJ Consulting, Inc.
7260 University Avenue N.E.
Suite 105
Minneapolis, MN 55432
(612) 571-8430

CHAIR, CONFIG. MGMT. WORKING GROUP
SIG SECRETARY
Mark Alan Kidwell
Texas Instruments Inc.
P.O. Box 801 M/S 8006
McKinney, TX 75069
(214) 952-2058

DEVEL. COUNTERPART, PDP-11 SOFTWARE
Joe Mulvey
Digital Equipment Corp. ,ZK01-3/J10
110 Spit Brook Road
Nashua, NH 03062-2642
(603) 881-1218

LISP/AI COORDINATOR
Don Rosenthal
Space Telescope Science Institute
Homewood Campus
Baltimore, MD 21218
(301) 338-4844

LIBRARY REPRESENTATIVE
SIG TAPE LIBRARIAN
CHAIR, PUBLIC DOMAIN SOFTWARE W/G
Tony Scandora
Argonne National Laboratory
CMT 205
Argonne, IL 60439
(312) 972-7541

CHAIR, C WORKING GROUP
James Maves
Eaton Corp, IMSD
31717 La Tienda Drive
Box 5009
Westlake Village, CA 91359
(818) 706-5395

STEERING COMMITTEE MEMBER-AT-LARGE
Jay Wiley
Bechtel Power Corp
12400 East Imperial Highway
Norwalk, CA 90650
(213) 807-4016

SESSION CHAIRS COORDINATOR
BOF CHAIRS COORDINATOR
SESSIONS QUALITY COORD.
ALT. SYMPOSIUM COORD.
Gary C. Lelvis
IMSL
2500 Park West Tower One
2500 City West Blvd.
Houston, TX 77042-3020
(713) 782-6060

CHAIR, FORTRAN WORKING GROUP
Scott Krusemark
Systemation, Inc.
8473 Daisywood Ave N.W.
North Canton, OH 44720
(216) 499-6251

CHAIR, LOW LEVEL LANGUAGES W/G
Gerald Lester
Computerized Processes Unlim.
4200 South 1-10 Service Road
Suite #205
Metairie, LA 70001
(504) 889-2784

DEVEL. COUNTERPART, COMM. LANG.
Jim Totton
Digital Equipment Corp.
ZK02-3/K06
110 Spit Brook Road
Nashua, NH 03062

ALT. ANSI X3J9 PASCAL STDS. COMM.
Phil Wirth
E-Systems, Garland Division
Box 660023, MS 53730
Dallas, TX 75266-0023
(214) 272-0515 x4319

ALT. ANSI X3J4 COBOL STDS. COMM.
Dale Marriott
El Paso County Office Bldg.
27 E. Vermijo Street
Colorado Springs, CO 80903
(303) 520-6435

ACTING CHAIR, DIBOL WORKING GROUP
ASSOC. W/G COORD. UNSCHEDULED TOPICS
CHAIR, COBOL WORKING GROUP
Bruce Mebust
Burlington Northern Railroad
176 East Fifth Street
P.O. Box 64962
St. Paul, MN 55164
(612) 298-2382

CHAIR, SECURITY WORKING GROUP
Rich Harris
General Research Corp.
5383 Hollister Avenue
P.O. Box 6770
Santa Barbara, CA 93160-6770
(805) 964-7724

ASSISTANT CAMPGROUND COORD.
Tom J. Jeffrey
Rockwell International Corp.
1225 N. Alma Road
Richardson, Texas 75081
(214) 996-7818

CHAIR, ADA WORKING GROUP
Lisa Kerby-Rodgers
GTE Government Systems
100 Ferguson Drive
P.O. Box 7188
Mountain View, CA 94039
(415) 966-2720

CHAIR, PROJECT MANAGEMENT W/G
Lynn C. Lewis
Lawrence Livermore National Lab.
University of California
P.O. Box 808
Livermore, CA 94550
(415) 422-8949

CHAIR, OBJ. ORIENTED DES. W/G
Frank B. Modruson
Arthur Andersen & Co.
33 West Monroe Street
Chicago, IL 60603
(312) 580-0033

CHAIR, TeX/LaTEX WEB W/G
John Osudar
Argonne National Lab.
9700 South Cass Avenue
Argonne, IL 60439
(312) 972-7505

CHAIR, VAXset W/G
David J. Powell
The Upjohn Company
7294-25-7
301 Henrietta St.
Kalamazoo, MI 49007
(616) 385-7214

MEM., ANSI DIBOL X3J12 Stds. Comm.
Kenneth Schilling
2314 Mira Vista Avenue
Montrose, CA 91020
(818) 249-0795

SUITE & RECEPTION COORD.
Matt Variot
Eaton Corporation
Box 5009
31717 La Tienda Drive
Westlake Village, CA 91359
(818) 706-5388

CHAIR, TPU/EVE/LSE W/G
John Wilson
Kellogg Company
724 Oak Brook Blvd
Battle Creek, MI 49015
(616) 961-3515

VICE CHAIR
SYMPOSIUM LOGISTICS COORD.
Terry Medlin
Survey Sampling, Inc.
1 Post Road
Fairfield CT 06432
(203) 255-4200

ASST. MASTERS COORDINATOR
Dena Shelton
Cullinet Software Inc.
2860 Zanker Rd, Suite 206
San Jose, CA 95134
(408) 434-6636

CHAIR, BASIC W/G
WISHLIST COORDINATOR
Bob Van Keuren
UserWare International, Inc.
4087 Cahmoune Ave.
San Diego, CA 92105
(619) 283-5285

SIC-5

INCOMING SIG CHAIR
Joseph Pollizzi, III
Space Telescope Science Institute
3700 San Martin Drive
Homewood Campus
Baltimore, MD 21218
(301) 338-4901

CHAIR, SCAN WORKING GROUP
WORKING GROUPS COORD.
VOLUNTEERS COORD.
David Ream
Lexi-Comp
26173 Tallwood Drive
N. Olmsted, OH 44070
(216) 777-0095
(216) 468-0744

CHAIR, PASCAL WORKING GROUP
MEMBER, ANSI PASCAL X3J9 STDS. COMM.
CHAIR, MODULA-2 W/G
STANDARDS COORD.
E. Wayne Sewell
E-Systems, Garland Div.
Box 660023, MS 53700
Dallas, TX 75266-0023
(214) 272-0515 x3553

CHAIR, SOFTWARE METRICS W/G
CAMPGROUND COORD.
MEMBER, ANSI C X3J11 STDS. COMM.
Michael S. Terrazas
LDS Church
50 E. North Temple, 27th Floor
Salt Lake City, UT 84150
(801) 531-3246

MEMBER, ANSI COBOL X3J4, STDS. COMM.
Bruce Gaarder
Donahue Enterprises, Inc.
2441 26th Ave., S.
Minneapolis, MN 55406
(612) 721-2418

ALT. SEMINAR COMM REP.
Janet E. Bressan
Boeing Aerospace Co.
P.O. Box 3999, MS3C-24
Seattle, WA 98124
(206) 773-9404

CHAIR, RPG WORKING GROUP
Charles Williamson
Hargray Telephone Co.
P.O. Box 5519
Hilton Head Is., SC 29938
(803) 686-1204

SEMINAR COMMITTEE REP.
Barry C. Breen
Sundstrand Data Control, Inc.
15001 N.E. 36th Street
P.O. Box 97001
Redmond, WA 98073-9701
(206) 885-8436

MASTERS COORD.
CLINIC DIRECTOR
CHAIR, CASE & TOOLS INTEGRATION W/G
CHAIR, CROSS-DEV. SYSTEMS W/G
George Scott
Computer Sciences Corp.
304 West Route #38, P.O. Box N
Moorestown, NJ 08057
(609) 234-1100

DEVEL. COUNTERPART TECH. LANG.
Leslie J. Klein
Digital Equipment Corp.
ZK02-3/N30
110 Spit Brook Road
Nashua, NH 03062

DIGITAL COUNTERPART
Celeste La Rock
Digital Equipment Corp.
ZK02-3/Q08
110 Spit Brook Road
Nashua, NH 03062

FOLDER EDITOR
Donald E. Amby
Delco Electronics Corporation
P.O. Box 471, MS1 A21
Milwaukee, WI 53201
(414) 768-2682

MEMBER ANSI PL/I X3J1 STDS COMM.
Arthur Coston
Applied Information Systems, Inc.
500 Eastowne Dr.
Chapel Hill, NC 27514
(800) 334-5510

ASST. SESSION CHAIRS COORD.
Antonino N. Mione
Rutgers University
Center for Computer & Info. Serv.
Hill Center
P.O. Box 879
Piscataway, NJ 08855-0879
(201) 932-4784

UPDATE.DAILY REPORTER
Terry Shannon
Computer Information Systems
165 Bay State Drive
Braintree, MA 02184
(508) 848-7515

CHAIR, APL WORKING GROUP
Chet Small
MIT Lincoln Laboratory
244 Wood Street
Lexington, MA 02173
(508) 981-4172
(508) 863-5500 x-4172

CHAIR, PL/I WORKING GROUP
Jack Straub
13102 Borgman
Huntington Woods, MI 48070
(313) 358-6338
(313) 541-1941

ASST. NEWSLETTER EDITOR
Jim Wilson
Pfizer Inc.
QC Division
P.O. Box 88
Terre Haute, IN 47808
(812) 299-2121 x271

COMMCOMM REPRESENTATIVE
Kerry Wycoff
1117 E. 1000 Street
Spanish Fork, UT 84660
(801) 240-5948
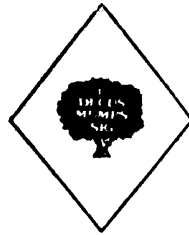
LARGE SYSTEMS

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

COMMUNICATIONS REPRESENTATIVE
NEWSLETTER EDITOR
Clyde T. Poole
The University of Texas at Austin
Department of Computer Sciences
Taylor Hall 2.124
Austin, TX 78712-1188
(512) 471-9551
ARPANET/CSNET:ctp@sally.utexax.edu

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

SYMPOSIUM REPRESENTATIVE
Vacant

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

SEMINARS REPRESENTATIVE
Robert C. McQueen
(201) 428-4242
ARPANET:SIT.MCQUEEN@cu20B.COLUMBIA.EDU

SUPERCOMPUTING
Vacant

SIG VICE-CHAIRMAN
Ralph M. Bradshaw
Johnson & Johnson
Research &Scientific Services
Management Information Center
Raritan, NJ 08869-1489
(201) 685-3434

LIBRARY REPRESENTATIVE
SIR/MENU BALLOT
Jack Stevens
The Gillette Company
Technical Services, 4U-3
1 Gillette Park
Boston, MA 02106-2131
(617) 463-2089

SIG MARKETING
Steve Attaya
Weiner Enterprises
P.O. Box 23607
Harahan, LA 70183
(504) 733-7055
ARPANET:G.ATTAYA@R20.UTEXAS.EDU

CORPORATE ISSUES
Ralph Bradshaw
Johnson & Johnson
Research and Scientific Services
Management Information Center
Raritan, NJ 08869-1489
(201) 685-3434

DEC COUNTERPARTS
Dave Braithwaite
Digital Equipment Corporation
Marlboro, MA

Rich Whitman
Digital Equipment Corporation
Marlboro, MA

Reed Powell
Digital Equipment Corporation
Marlboro, MA

MUMPS SIG

CHAIRMAN
Chris Richardson
Richardson Computer Research
P.O. Box 8744
La Jolla, CA 92038
(619) 488-6193

NEWSLETTER EDITOR
VICE-CHAIR
COMMCOMM REP.
Mark J. Hyde
Advanced Computing Services
209 Ardsley Drive
DeWitt, NY 13214
BITNET: MJHYDE@SUNRISE
INTERNET: MJHYDE@SUNRISE.ACS.SYR.EDU
(315) 446-7223

SYMPOSIUM SCHEDULER
Brad Hanson
Group Health, Inc.
2829 University Ave., S.E.
Minneapolis, MN 55414
(612) 623-8427

LIBRARY REPRESENTATIVE
PDP-11 WORKING GROUP REP.
Michael McIntyre
PRx, Inc.
43 Bradford Street
Concord, MA 01742
(617) 369-3566

SEMINARS REPRESENTATIVE
Edward Woodward
Science Applications Intl. Corp.
10260 Campus Point Drive MS42
San Diego, CA 92121
(619) 535-7210

CAMPGROUND COORDINATOR
ASSIST. SYMPOSIUM SCHEDULER
Jerry Hsu
Rubicon Corp.
1200 E. Campbell
Richardson, TX 75083
(214) 231-6591

SESSION NOTES EDITOR
Paul A. Price
SciCor, Inc.
2643 Rand Road
Indianapolis, IN 46241
(317)244-8811

PAST CHAIR
MUMPS DEV. COMMITTEE REP.
Mark Berryman
Digital Equipment Corp.
3 Results Way (MR03-2/H7)
Marlborough, MA 01752
(617) 467-4875
BITNET: BERRYMAN@DSM.DEC.COM

DEC COUNTERPART
Dave Smith
Digital Equipment Corp.
2 Iron Way (MR03-2/H7)
Marlborough. MA 01752
(617) 467-2397

ALTERNATE DEC COUNTERPART
Denise Simon
Digital Equipment Corp.
129 Parker Street (PK02-1/M23)
Maynard, MA 01754
(617) 493-9077

NETWORKS SIG

CHAIRMAN
Stuart Lewis
Douglas Furn. of California, Inc.
5020 W. 73rd St., Box 97
South Suburban, IL 60499-0097
(312) 458-1505

SYMPOSIUM COMMITTEE REP
L. Stuart Vance
University of Texas System
Office to Telecomm. Services
Balcones Research Center
10100 Burnet Road
Austin, TX 78758-4497
(512) 471-2416

SEMINARS COMMITTEE REP
Jeffrey Snover
47 Walden Pond Dr.
Nashua, NH 03060
(508) 256-6600

STANDARDS REP
Jim Ebright
Software Results Corp.
2887 Silver Drive
Columbus, OH 43211
(614) 267-2203

COMMCOMM REP
Allen Jay Bennett
Logisticon, Inc.
1551 Alexander SE
Grand Rapids, MI
(616) 452-1823

NEWSLETTER EDITOR
Judi Mandl
University of Conn. Health Center
263 Farmington Ave.
Farmington, CT 06032
(203) 674-3912

SESSION NOTES EDITOR
Mary Marvel-Nelson
General Motors Research Lab.
Warren, MI 48090
(313) 986-1382

WISH LIST COORDINATOR
Stuart L. Labovitz
USAF AFWAL/AADM-2
Bldg 620 Area B
WPAFB, OH 45433-6543
(513) 255-7680

PAST CHAIRMAN
Bill Brindley
HDQ Naval Security Group Cmd.
(202) 282-0527

TECHNOLOGY COORDINATOR
Bill Hancock
ERI Training
P.O. Box 13557
Arlington, TX 76094-0557

MEMBER-AT-LARGE
Sandy Traylor
Target Systems
21063 Carlos Rd.
Yorba Linda, CA 92686

DEC COUNTERPART
Monica Bradlee
Digital Equipment Corporation
550 King St. LKG2-1/U2
Littleton, MA 01460
(508) 486-7341

OFFICE AUTOMATION SIG

(*) CHAIR
Joseph W. Whatley
Nielsen Media Research
375 Patricia Ave.
Dunedin, FL 34698
(813) 734-5473 x 2438

(*) VICE CHAIR
Ralph Bradshaw
Johnson and Johnson
Route 202
Raritan, NJ 08869-1489
(201) 685-3434

SIR PROGRAM ADMINISTRATOR
Edward L. Bowen
VAX Systems Manager
BellSouth Services
1876 Data Drive, Room N204
Birmingham, AL 35244
(205) 988-6800

LUG COORDINATOR & OASIG FOUNDER
Tom Orlowski
American Council on Education
One Dupont Circle (Suite 110)
Washington, DC 20036
(202) 939-9371

OASIS ADMINISTRATOR
Dale E. Coy
E-8, MS/J957
Los Alamos National Laboratory
P.O. Box 1663
Los Alamos, NM 87545
(505) 667-3270

OASIS ASSISTANT ADMINISTRATOR LIBRARY REPRESENTATIVE
Bob Hassinger
Liberty Mutual Research Ctr.
71 Frankland Road
Hopkinton, MA 01748
(508) 435-9061

SUITE COORDINATOR
Jeanne Ward
State University of New York at Stony Brook
Stony Brook, NY 11794-3499
(516) 632-7795

ASSISTANT SUITE COORDINATOR
Mary L. Keenan
State University of New York at Stony Brook
Stony Brook, NY 11794-2400
(516) 632-8053

(*) SYMPOSIA COORDINATOR
Mitch Brown
Bank of Boston
Central Systems, 99-02-07
Box 2016
Boston, MA 02106-2016
(508) 434-1587 / 6318

ASSISTANT SYMPOSIA COORDINATOR
Lynda L. Peach
Mustang Energy Corp.
1100 First National Center East
Oklahoma City, OK 73102
(405) 272-9471

ROADMAP/PUBLICATIONS COORDINATOR
Scott McClure
Office Systems Senior Analyst
Industrial Tape Division/3M
220-8E-01 3M Center
St. Paul, MN 55144-1000
(612) 736-4297

SESSION CHAIR COORDINATOR
Dee Wade
Office Automation Specialist
Silicon Systems
14351 Myford Road
Tustin, CA 92680
(714) 731-7110

SPEAKER COORDINATOR
Monica Falk
MIF Enterprises
3260 S. Irving Street, Suite C202
Englewood, CO 80110
(303) 781-5726

CAMPGROUND COORDINATOR
LT. Scott Cline
AFWAL/AATC
Wright-Patterson AFB, OH 45433-6543
(513) 255-5623

(*) COMMUNICATIONS COMMITTEE REP
Mary-Jane Bolling
Foreign Mission Board
3806 Monument Avenue
Richmond, VA 23230
(804) 353-0151

NEWSLETTER EDITOR
Therese LeBlanc
T.M. LeBlanc & Assoc.
275 London
Wheeling, IL 60090
(312) 459-1784

TAPE COORDINATOR
Roger Bruner
Foreign Mission Board
3806 Monument Avenue
Richmond, VA 23230
(804) 353-0151

STORE COORDINATOR
Ellen Shields
E-Systems
P.O. Box 12248
St. Petersburg, FL 33733
(813) 381-2000 x2010

ASSISTANT STORE COORDINATOR
Linda A. Peters
E-Systems
P.O. Box 12248
St. Petersburg, FL 33733
(813) 381-2000 x3517

SIG 'EFFECTS' COORDINATOR
Robert O. Wilkinson
Naval Underwater Systems Center
Code 301
New London, CT 06320
(203) 440-4654

SESSION NOTES EDITOR
George Bone
Code 2301.2 mailstop T-11
Mare Island Naval Shipyard
Vallejo, CA 94592-5100
(707) 646-2531

SIG PERQS COORDINATOR
Enid Woods
Word Processing Coordinator
Crowe, Chisek and Company
330 E. Jefferson Blvd.
P.O. Box 7
South Bend, IN 46624
(219) 232-3992

(*) SPECIAL PROJECTS COORDINATOR
Katherine Trimm
PIVOTAL, Inc.
6892 Dorado Ct.
Tucson, AZ 85715
(602) 886-5563

SECURITY WORKING GROUP CHAIR
Ray Kaplan
PIVOTAL, Inc.
6892 E. Dorado, CT
Tucson, AZ 85715
(602) 886-5563

(*) Executive Committee Members

OASIS (OA SIG NOTES CONFERENCES)

(603) 884-1738   1200 baud   8 bit, no parity
(603) 88401742   2400 baud   8 bit, no parity

PERSONAL COMPUTER SIG

CHAIR
Lynn Jarrett
San Diego Union-Tribune Pub. Co.
350 Camino de la Reina
San Diego, CA 92108
(619) 293-1130

SYMPOSIA COORDINATOR
Jimbo Wilson
Nat'l Tech. Inst. for Deaf
Rochester Inst. of Tech.
P.O. Box 9887
Rochester, NY 14623
(716) 475-6241

CAMPGROUND COORDINATOR
Jim Hobbs
Golden, CO 80401-1295
(303) 277-2855

WORKSTATIONS/PRO WORKING GROUP CHAIR
Thomas R. Hintz
University of Florida
IFAS Computer Network
Bldg. 120
Gainesville, FL 32611
(904) 392-5180

MACINTOSH WORKING GROUP CHAIR
Michael J. Prezbindowski
Boeing Computer Services
P.O. Box 24346, MS 6H-23
Seattle, WA 98124-0346
(206) 234-1328

PCSA WORKING GROUP CHAIR
Fran Garrett
San Diego Union-Tribune Pub. Co.
350 Camino de la Reina
San Diego, CA 92108
(619) 293-1676

RAINBOW/DECMATE WORKING GROUP CHAIR
Vince Perriello
Crosfield Composition Systems
One Crosfield Avenue
West Nyack, NY 10994
(914) 353-4000

COMM COMM REP/SESSION NOTES EDITOR
Dr. Thomas Warren
Oklahoma State University
Department of English
Dir. Tech. Writing Program
Stillwater, OK 74078
(405) 744-6218

NEWSLETTER EDITOR
Gary Rice
McDonnell Douglas
5555 Garden Grove Blvd.
MS: K200 77/200
Westminster, CA 92683
(714) 952-6582

SEMINARS COORDINATOR
Tim Bundrick
3480 TCHTW/TTVC
Goodfellow AFB, TX 76908-5000
(915) 657-5424

BUTTON COORDINATOR
Ken Stricker
Martin Marietta Aerospace
P.O. Box 5837 MP 320
Orlando, FL 32855
(305) 356-6589

STORE REP
George Dover
Tektronix Inc.
MS 73-646
P.O. Box 500
Beaverton, OR 97077
(503) 627-4592

MEMBERS-AT-LARGE
Barbara Maaskant
UT Health Science Center
7703 Floyd Curl Drive
San Antonio, TX 78284
(512) 567-2200

RSTS SIG

CHAIRMAN
Charles Mustain
Stark County School system
Data Services Division
7800 Columbus Rd. N.E.
Louisville, OH 44641
(216) 875-1431

COMMUNICATIONS REPRESENTATIVE
STORE REPRESENTATIVE
Ed Beadel
Instructional Computer Center
S.U.N.Y. College at Oswego
Oswego, N.Y. 13126
(315) 341-3055

SYMPOSIA COORDINATOR
Glenn Dollar
Digital Computer Consultants Inc.
21363 Lassen St., Suite 205
Chatsworth, CA 91311
(818) 341-9171

ASS'T SYMPOSIA COORDINATOR
Dan Stoller
Natural Country Farms
P.O. Box 758
58 West Road
Rockville, CT 06066
(203) 872-8346

NEWSLETTER EDITOR
Terence M. Kennedy
St. Peter's College
Department of Computer Science
2641 Kennedy Blvd.
Jersey City, NJ 07306
(201) 435-1890

LIBRARY REPRESENTATIVE
R.R. Patel (PAT)
Krupp/TaylorUSA
12800 Culver Blvd
Los Angeles, CA 90066
(213) 306-3646

PRE-SYMPOSIA SEMINAR COORDINATOR
Scott Castleberry
1750 North Collins
Suite 108
Richardson, TX 75080
(214) 437-3477

VICE CHAIRMAN
WISH LISTS COORDINATOR
Lynnell Koehler
Campus America
POISE Prod. Ctr.
201 North Nevada Avenue
Roswell, NM 88201
(505)625-5500

EDUSIG LIAISON
George Wyncott
Purdue University Computer Center
W. Lafayette, IN

RSTS PRODUCT PLANNING COORDINATOR
Errol E. Ethier
Information Design and Management, Inc.
23 Hunting Avenue
(617) 842-4220
Shrewsbury, MA 01545

DEC COUNTERPART
Kathy Waldron
Digital Equipment Corporation
Continental Blvd.
Merrimack, NH 03054

MEMBERS-AT-LARGE
Edward F. Beadel, Jr.
Manager
Instructional Computing Center
S.U.N.Y. College at Oswego
Oswego, NY 13126
(315) 341-3055

Mark Hartman
Jadtec Computer Group
546 W. Katella Avenue
Orange, CA 92667
(714) 997-8928

Jeff J. Killeen
Information Design & Management, Inc.
31 Hopedale Street
Hopedale, MA 01747

TAPE COPY COORDINATOR
W. Franklin Mitchell, Jr.
Erskine College
1 Washington Street
Due West, SC 29639
(803) 379-2131
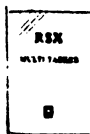
RSX SIG

CHAIRMAN
Dan Eisner
Perkin-Elmer Corp.
Garden Grove, CA

SYMPOSIA COORDINATOR
Rick Sharpe
Toledo Edison
Toledo, OH

PRE-SYMPOSIUM SEMINAR COORDINATOR
Hans Jung
Associated Press
New York, NY

COMMUNICATIONS REPRESENTATIVE
Jay Allen Bennett
Logisticon, Inc.
5035 Whitneyville Road
Ada, MI 49301
(616) 452-1823

NEWSLETTER EDITOR
MULTI-PROCESSORS WORKING GROUP COORDINATOR
Bruce Mitchell
Machine Intelligence & Industry Magin
Byron, MIN

STORE COORDINATOR
Jim Hopp
Carlton Financial Computation
South Bend, IN

SESSION NOTE EDITOR
Burt Janz
BHJ Associates
Nashua, NH

LIBRARIAN
Glenn Everhart
Mt. Holly, NJ

CAMPGROUND COORDINATOR
Jerry Ethington
Prolifix Inc.
Frankfort, KY

DEC COUNTERPARTS
Lin Olsen
Nashua, NH

Dick Day
Nashua, NH

WORKING GROUP COORDINATOR
Sharon Johnson
Epidemiology
Minneapolis, MN

WORKING GROUP CHAIR
Evan Kudlajev
Philadelphia Electric Co.
Philadelphia, PA

RSX GROUP CHAIR SOFTWARE CLINIC COORD.
Roy S. Maull
U.S. Air Force
Offutt AFB, NE

SOFTWARE CLINIC COORDINATOR
Bruce Zielinski
RCS
Moorestown, NJ

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

SRD WORKING GROUP COORDINATOR
Bob Turkelson
Goddard Space Flight Center
Greenbelt, MD

ACCOUNTING & PERFORMANCE WORKING GROUP COORD.
Denny Walthers
American McGaw
Irvine, CA

MENU COORDINATOR
Ed Cetron
Center for Biomedical Design
Salt Lake City, UT

MEMBERS-AT-LARGE
Jim McGlinchey
Warrenton, PA

Jim Neeland
Hughes Research Labs.
Malibu, CA

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

Ralph Stamerjohn
Creve Coeur, MO

RT-11 SIG

CHAIRMAN

John T. Rasted
JTR Associates
58 Rasted Lane
Meriden, CT 06450
(203) 634-1632

COM. COM VOTING REP.
COBOL CONTACT

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

STANDARDS COORDINATOR

Robert Roddy
Naval Ship Research Ctr.
Bethesda, MD 20084
(301) 227-1724

MACRO CONTACT

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

NEWSLETTER EDITOR
TECO CONTACT
PRODUCT PLANNING CONTACT

John M. Crowell
Multiware, Inc.
2121-B Second St. Suite 107
Davis, CA 95616
(916) 756-3291

NETWORKING CONTACT

Jim Crapuchettes
Omnex Corp.
2483 Old Middlefield Way
Mountain View, CA 94043
(415) 966-8400

WISH LIST CONTACT
UNIX/ULTRIX CONTACT

Bradford Lubell
L.A. Heart Lab, UCLA
10833 Le Conte Avenue
Los Angeles, CA 90024-1760
(213) 206-6119

TSX & C CONTACT

Jack Peterson
Horizon Data Systems
P.O. Box 29028
Richmond, VA 23229
(804) 740-9244

RUNOFF CONTACT

John Davis
Naval Ship Research Center
Code 2950
Bethesda, MD 20084
(301) 227-1592

LUG CONTACT

Ned Rhodes
Software Systems Group
2001 North Kenilworth St.
Arlington, VA 22205
(703) 534-2297

PERSONAL COMPUTERS

Dennis V. Jensen
AMES Labs. ISU/USDOE
310 Metallurgy
Ames, Iowa 50011
(515) 294-4823

SYMPOSIA COORDINATOR

Milton Campbell
Talisman Systems
Drawer CP-255
Manhattan Beach, CA 90266
(213) 318-2206

TAPE COPY GENERATION
TAPE COPY DISTRIBUTION
RT DECUS LIBRARY CONTACT

Tom Shinal
Syntropic Technology
P.O. Box 198
Waterford, VA 22190
(703) 882-3000

PRE-SYMPOSIUM SEMINAR
RT-11 SUITE MANAGER

Bruce Sidlinger
Sidlinger Computer Corp.
4335 N.W. Loop 410, #209
San Antonio, TX 78229
(512) DIG-ITAL

BASIC CONTACT

Ralston Barnard
Div 7523
Sandia Labs
Alburquerque, NM 87185
(505) 844-5115

PRO RT-11 & HARDWARE

Bill Walker
Monsanto Research Corp.
P.O. Box 32, A-152
Miamisburg, OH 45342
(513) 865-3557

FORTRAN CONTACT

Robert Walraven
Multiware, Inc.
2121-B 2nd St. Suite 107
Davis, CA 95616
(916) 756-3291

OTHER LANGUAGES

Gary Sallee
19912 Fernglen Drive
Yorba Linda, CA 92686
(714) 970-2864

SITE SIG

CHAIRMAN

Timothy Fraser
Specialized Bicycle Components
15130 Concord Circle #77
Morgan Hill, CA 95037
(408) 779-6229

SYMPOSIA COORDINATOR

Sue Abercrombie
48 Malilly Rd.
Portland, ME 04103
(207) 772-2837

SESSION NOTE EDITOR
LARGE SYSTEMS SIG LIAISON

Gary Bremer
Emerson Electric Co.
8100 W. Florisant
St. Louis, MO. 63136
(314) 553-4448

NEWSLETTER EDITOR
NETWORKS SIG LIAISON
OA SIG LIAISON

Gregory N. Brooks
Washington University
Behavior Research Labs
1420 Grattan St.
St. Louis, MO. 63104
(314) 241-7600 ext. 257

HARDWARE COORDINATOR
HMS SIG Liason

Emily Kitchen
A.H. Robins Co.
1211 Sherwood Ave. RT-2
Richmond, VA. 23220
(804) 257-2925

COMMUNICATIONS COMMITTEE REPRESENTATIVE
AI SIG Liason

Terry C. Shannon
Digital Review
160 State St.
6th Floor
Boston, MA. 02109
(617) 367-7190

PRE-SYMPOSIA SEMINAR COORDINATOR

Phillip Ventura

STAFF MANAGEMENT

Adam Zavitski
Simmonds Precision ICD
3100 Highland Blvd.
Raleigh, NC. 27625
(919) 872-9500

MEMBERS-AT-LARGE

Ann Goergen
Texas Instruments
13510 N. Central
M/S 437
Dallas, TX. 75266
(214) 995-4629

HMS SIG Liason
RT SIG Liason

David Hunt
Lawrence Livermore National Lab
MS L-230
P.O. Box 808
Livermore CA. 94550
(802) 656-3190

Gary Siftar
Digital Equipment Corporation
Tulsa, OK.

DEC COUNTERPARTS

Joe Allen
Stow MA.

Lil Holloway
Bedford MA.

Susan Porada
Marlboro, MA.

UNISIG

CHAIRMAN

Kurt L. Reisler
Hadron Incorporated
9990 Lee Highway
Fairfax, VA 22030
(703) 359-6100
decvax!seismo!hadron!klr

SYMPOSIA COORDINATOR

Sam Kimery
Purdue University Math Dept.
Math Science Bldg.
West LaFayette, IN 47907
(317) 494-6055
ucbvax!helicon.math.purdue.edu!kimery

SESSION NOTES EDITOR

William Cheswick
New Jersey Institute of Tech.
Computing Services
323 Martin Luther King Blvd.
Newark, NJ 07102
(201) 596-2900
bellcore!njitccc!bc

NEWSLETTER EDITOR

Sharon Gates-Fishman
NDC Corporation
730 E Cypress Avenue
Monrovia, CA 91016
(818) 358-1871
!amdahl!cit-vax!ndc!sgf

SIC-9

COMMCOMM REPRESENTATIVE
James W. Livingston, Jr.
Measurex Automation Systems
10411 Bubb Rd
Cupertino, CA 95014-4150
(408) 973-1800 x 766
ihnp4!masl!jwl

ADMINISTRATIVE DAEMON
Dorothy A. Geiger
The Wollongong Group
49 Showers Drive, #451
Mountain View, CA 94040
(415) 948-1003
ihnp4!decwrl!dgeiger

TAPE LIBRARIAN
Carl Lowenstein
Marine Physical Laboratory
Scripps Institute of Oc'graphy, P-004
LaJolla, CA 92093
(619) 294-3678
[ihnp4,decvax,akgua,dcdwest,ucbvax]
!sdcsvax!mplvax!cdl

USENET LIAISON
STANDARDS COORDINATOR
Ed Gould
Mt. Xinu
2910 7th St.
Suite 120
Berkley, CA 94710
(415) 644-0146
ucbvax!mtxinu!ed

MINISTER WITHOUT PORTFOLIO
Norman Wilson
Bell Laboratories, 2C-529
600 Mountain Avenue
Murray Hill, NJ 07974
(201) 582-2842
[decvax,ihnp4]!research!norman

SEMINARS COORDINATOR
Steven Stepanek
Computer Science Dept.
School of Eng. & Computer Science
18111 Nordhoff St.
Northridge, CA 91330
(818) 885-2799 or 3398
ihnp4!csun!sgs

DEC COUNTERPART
Gary Oden
Digital Equipment Corporation
Continental Blvd, MK02
Merrimack, NH 03054
(603) 884-5111
decvax!oden

VAX SYSTEMS SIG
CHAIRMAN
Susan T. Rehse
Lockheed Missiles & Space Co.
0/19-50, B/531,
P.O. Box 3504
Sunnyvale, CA 94088-3504
ctp@cs.utexas.edu.

EXECUTIVE COMMITTEE
Jeffrey Jalbert
J C C
P.O. Box 381
Granville, OH 43023

Betsy Ramsey
American Mathematical Society
P.O. Box 6248
Providence, RI 02940

David Schmidt
Management Science Associates
6565 Penn Avenue
Pittsburgh PA 15206-4490

Al Siegel
Battelle Memorial Institute
505 King Avenue
Columbus, OH 43201-2693

David Wyse
Projects Unlimited
3680 Wyse Road
Dayton, OH 45414-2539

SYMPOSIA COORDINATOR
Betsy Ramsey
American Mathematical Society
P.O. Box 6248
Providence, RI 02940

SYMPOSIA COORDINATOR, ASST.
Michael Carullo
Westinghouse Electric Corp.
10120 Kings Bench Court
Ellicot City, MD 21043

SESSION CHAIRMAN COORDINATOR
Elaine Hall
Westinghouse
P.O. Box 746
M/S 432
Baltimore, MD 21203

SEMINAR COORDINATOR
Robert McQueen
Knoll Pharmaceuticals
MIS Department
30 North Jefferson Road
Whippany, NJ 07981

LIBRARY COORDINATOR
Glen Everhart
General Electric
25 Sleigh Ride Road
Glen Mills, PA 19342

COMMUNICATIONS COORDINATOR
David Wyse
Projects Unlimited
3680 Wyse Road
Dayton, OH 45414-2539

NEWSLETTER EDITOR
Clyde T. Poole
The University of Texas at Austin
Department of Computer Sciences
Taylor Hall 2.124
Austin, TX 78712-1188
ARPANET/CSNET:ctp@sally.utexas.edu

PAGESWAPPER VAX MANAGER
Lawrence J. Kilgallen
Box 81, MIT Station
Cambridge, MA 02139-0901

WORKING GROUP COLUMNIST
Eric Wentz
G.E. Lighting
Nela Park Noble Road #1711
Cleveland, Ohio 44112

SESSION NOTES EDITOR
Ken Johnson
Meridien Technology
P.O. Box 2006
St. Louis, MO 63011

SYMPOSIUM HANDOUT
BOOTBLOCK EDITOR
John L. Prather
Technicon Instruments Corp.
511 Benedict Avenue
Tarrytown, NY 10591

STORE COORDINATOR
G. Beau Williamson
Rockwell International
1200 N. Alma Road
MIS 406-280
Richardson, TX 75081

MASTER'S LIST COORDINATOR
Carl Friedberg
Seaport Systems
165 William Street, 9th Floor
New York, NY 10038-2605

SYSTEM IMPROVEMENT REQUEST (SIR)
David Schmidt
Management Science Associates
6565 Penn Avenue
Pittsburgh, PA 15206-4490

VOLUNTEER COORDINATOR
Elizabeth Bailey
222 CEB
Tennessee Valley Authority
Muscle Shoals, AL 35661

WEST COAST VOLUNTEER COORDINATOR
Ron Tencati
Jet Propulsion Lab
4800 Oak Grove Drive
MS:602-145
Pasadena, CA 91109

CAMPGROUND COORDINATOR
David Paul
Shell Oil Company
MS: D2145
P.O. Box 1380
Houston, TX 77251-1380

SUITE COORDINATOR
Frank Cannan
Hamilton/Avnet
103 Twin Oaks Drive
Syracuse, New York 13206

WORKING GROUP COORDINATOR
Jeffrey Jalbert
J C C
P.O. Box 381
Granville, OH 43023

COMMERCIAL WORKING GROUP CHAIR
E.F. Berkley Shands
Washington University
Department of Computer Science
P.O. Box 1045
St. Louis, MO 63136

DECNET SECURITY WORKING GROUP CHAIR
Ron Tencati
Jet Propulsion Lab
4800 Oak Grove Drive
MS: 602-145
Pasadena, CA 91109

HIGH-END VAX SYSTEMS WORKING GROUP CHAIR
Al Siegel
Battelle Memorial Institute
505 King Avenue
Columbus, OH 43201-2693

INTERNALS WORKING GROUP CHAIR
Carl Friedberg
Seaport Systems
165 William Street, 9th Floor
New York, NY 10038-2605

MICROVAX WORKING GROUP CHAIR
David Mehren
Pivotal, Inc.
6892 East Dorado Court
Tucson, AZ 85715-3264

MIGRATION AND HOST DEVELOPMENT
VAXINTOSH WORKING GROUP CHAIR
Jim Downward
KMS Fusion Inc.
P.O. Box 156D
Ann Arbor, MI 48106

MULTIPROCESSOR WORKING GROUP CHAIR
Eugene Pal
U.S. Army
CAORA (ATORCATC)
Fort Leavenworth, KA

PERFORMANCE WORKING GROUP CO-CHAIR
John T. Peterson
Data Metrics Systems Inc.
5270 Lyngate Court
Burke, VA 22015

PERFORMANCE WORKING GROUP CO-CHAIR
Louise Wholey
Measurex Corporation
One Results Way
Cupertino, CA 95014

REAL TIME/FOREIGN DEVICES WORKING GROUP CHAIR
Larry Robertson
Bear Computer Systems
56512 Case Avenue
North Hollywood, CA

SECURITY WORKING GROUP CHAIR
C. Douglas Brown
Sandia National Labs
Division 2644
P.O. Box 5800
Alburquerque, NM 87185-5800

SYSTEM MANAGEMENT WORKING GROUP CHAIR
Steve Tihor
251 Mercer Street
New York, NY 10012

VAXCLUSTER WORKING GROUP CHAIR
Thomas Linscomb
Computation Center
University of Texas
Austin, TX 78712

VMS USER'S NETWORK WORKING GROUP CHA
Jamie Hanrahan
Simpact Associates
9210 Sky Park Court
San Diego, CA 92123

ADVISORS
Joseph Angelico
U.S. Coast Guard Detachment
National Data Buoy Center
NSTL Station, MS 39529-6000

Jack Cundiff
Horry-Georgetown Tech. College
P.O. Box 1966
Conway, SC 29526

Don Golden
Shell Oil Company
MS:D2145
P.O. Box 1380
Houston, TX 77251-1380

Marg Knox
Computation Center
University of Texas
Austin, TX 78712

Art McClinton
Mitre
1820 Dolley Madison Blvd.
McLean, VA 22102

Ross Miller
Online Data Processing
N 637 Hamilton
Spokane, WA 99202

SUBMITTING ARTICLES TO HARD NEWS

The purpose of HARD NEWS, 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 would like to publish it in the newsletter. We hope that HARD NEWS will be published at least six times a year.

You can submit material to the editor, Carmen Wiseman, or to the HMS SIG chair, Bill Walker. We can accept submissions in a wide variety of formats:

o   Items can be sent to the editor on VMS-format RX50s, TK50 cartridges, or IBM PC format 5 1/4" floppies. The SIG chair prefers RT-11 floppies but can handle any reasonable media.

o   Hard copy, like cash, is always acceptable. Camera-ready copy will save us a lot of typing, but we don't insist on it. You can also use the Hardware Submission Form in the "Questionnaires" section of the combined SIGs Newsletters.

o   Those of you with access to DCS can send things to WALKER or WISEMAN. DCS is usually checked on a daily basis.

o   You can reach the SIG chair on CompuServe as "Bill Walker 71066,24" or via EasyLink mailbox 62752448 or MCI Mail account 333-1675. You can reach the editor via EasyLink mailbox 62960090 (be sure to say ATTN: or TO: Carmen Wiseman somewhere in the body of the message).

If you have anything to submit, send it! If it is a mess, but we can read it, we will get it into the newsletter somehow. Finally, if you have any questions about submitting material, call one of us. The telephone numbers are listed below.

Contributions can be sent to:

William K. Walker   Carmen D. Wiseman
Monsanto Research Corp.   OR   Digital Review
P.O. Box 32   A-152   ==   Prudential Tower, Suite 1390
Miamisburg, OH   45342   800 Boylston Street
(513) 865-3557   (work)   Boston, MA   02199
(513) 426-7094/0344 (home)   (617) 375-4361 (work)

SUB

Ask the WOMBAT WIZARD
Submission Form

To submit a problem to the WIZARD, please fill out the form below and send it to:

> WW Editor, Philip A. Naecker
> Consulting Software Engineer
> 3011 North Mount Curve Avenue
> Altadena, CA 91001
> USA

Name:_____ DECUS Membership No. _____

Affiliation:_____

Address:_____

_____

_____

Telephone Number:_____

Statement of Problem:_____

_____

_____

_____

_____

_____

Please following the following guidelines when submitting support material:

1. If you are trying to demonstrate a method or a concept, please simplify the procedures, records, and other information to the shortest form possible.

2. Annotate your attachments. Simple comments or hand-written notes ("Everything worked until I added this statement.") go a long way toward identifying the problem.

3. Keep an exact copy of what you send. And number the pages on both copies. But send everything that is related to your question, even remotely.

4. If you would like a direct response or would like your materials returned, please don't forget to include a stamped, self-addressed envelope large enough to hold the materials you send.
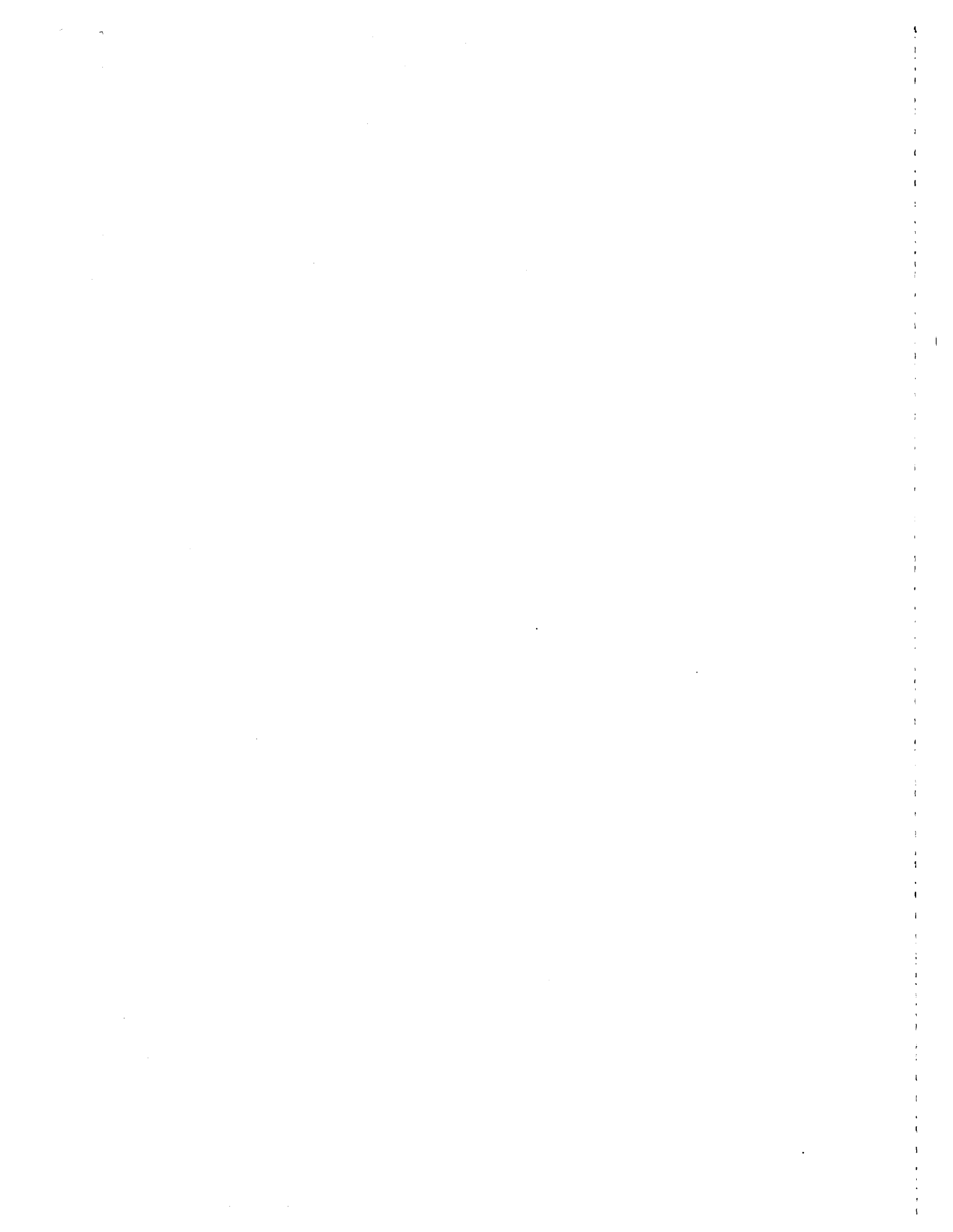
DATATRIEVE/4GL SIG
1989 Product Improvement Request Submission Form
Submission Deadline – January 13, 1989

Submittor: DECUS Membership Number:
Address: Firm:

Phone: Product or Products:

How to write a PIR.

A PIR should be directed at a specific product or group of products. Be sure to give the full name of the product(s) and version numbers if applicable. Describe the functionality you would like to see in as complete terms as possible. Don't assume that the PIR editors or software developers know how it is done in some other software product - state specifically how you want the software to function. Provide justification of your request and give an example of its use. If you can, suggest a possible implementation of your request.

Abstract: (Please limit to one or two short sentences.)

Description and Examples: (Use additional pages as necessary.)

Mail submission to:

T. Chris Wool, PIR Editor
E. I. duPont DeNemours & Co.
Engineering Department
P. O. Box 6090
Newark, DE 19714-6090

HARDWARE SUBMISSION FORM -- A SIG INFORMATION INTERCHANGE

Message

Contact
Name

Address

Telephone

Type of equipment

SUBMIT ANY TYPE OF HARDWARE PROBLEMS AND/OR FIXES.

SEND TO:

William K. Walker   Carmen D. Wiseman
Monsanto Research Corp.   OR Digital Review
P.O. Box 32 A-152   == Prudential Tower, Suite 1390
Miamisburg, OH 45342   800 Boylston Street
                              Boston, MA 02199

MASTERS APPLICATION

Name: _____ **Title** _____

Company: _____

Address: _____

_____

_____ **Phone:** (   ) _____

Network Address: _____ **Date:** _____

The Languages & Tools SIG has established the designation "LANGUAGES AND TOOLS MASTER", to be applied to selected, qualified people willing to share their expertise in various subjects with others. Masters are people who are knowledgeable enough in one or more languages or tools to be comfortable answering questions about them. The qualifications of an L&T Master are: expertise in a specific area, a willingness to have his/her name published as a Master, and a willingness to volunteer services in different ways. Each product may have several Masters, and there is an overall Masters Coordinator who is a member of the L&T Steering Committee.

Masters are asked to serve other users (and, under some circumstances, DEC), as a resource on products within their competence. In addition to being listed in the L&T Masters Directory (published in the newsletter) as available for occasional telephone consultation, Masters may act as 'Doctors' at Symposium Clinics, present Symposium sessions on the products of interest to them, field test products, interact with DEC product managers when appropriate, or act as a reference for a product for Digital salespeople. Especially on mature products, the SIG is anxious for knowledgeable users to offer product tutorial sessions at Symposia, and Masters can be of great help here. At Symposia, Masters will wear an identifying button bearing the legend "Ask Me About....." and the name of the language or tool in which he/she specializes.

If you'd like to serve as an L&T Master, please mark the products on which you are willing to answer questions with an "**M**" (for Master). Please mark any other products running at your site with an "**A**" (for "also running") to provide users with a broader picture of your facilities. (Although not an L&T product, Mumps is included here at the request of the Mumps SIG as a service to Mumps users). You may request removal of your name from the Masters Directory at any time, although you may continue to be listed for a month or two, because of publication lead times.

I am qualified to act as an L&T Master for the following products: ☐ Mumps

☐ Debug	☐ Bliss	☐ CMS	☐ TPU	☐ C	☐ Test Manager
☐ Pascal	☐ Basic	☐ MMS	☐ EVE	☐ Ada[1]	☐ Runoff & DSR
☐ Fortran	☐ Cobol	☐ LSE	☐ EDT	☐ APL	☐ TeX & LaTeX
☐ Document	☐ Dibol	☐ SCA	☐ TECO	☐ RPG	☐ Cobol Generator
☐ VAX Notes	☐ Emacs	☐ PCA	☐ PL/I	☐ Scan	☐ Software Project Mgr

Briefly describe your experience with those you checked. _____

_____

_____

_____

How long have you held your present position? _____

Are you able to attend at least one symposium each year? _____

Users are encouraged to seek assistance with products by calling appropriate Masters listed in the Directory. As a Master, your name and telephone number will be published in the Masters Directory, and users will call on you for limited help from time to time. Please check, below, any additional activities you might do:

☐ Field-test new versions of your product at your work site.

☐ Provide feedback on the product when needed by its DEC product manager.

☐ Act as a reference for the product at the request of Digital Sales or Marketing people.

Mail to: **George Scott, L&T SIG Masters Coordinator, Computer Sciences Corporation, 304 West Route #38, P.O. Box N, Moorestown, NJ 08057.**

[1]Ada is a trademark of the DoD

Languages & Tools SIG

WISHLIST QUESTIONNAIRE

Name: _____ **Title** _____.

Company: _____

Address: _____

_____

_____ **Phone:** ( ) _____

Network Address: _____ **Date:** _____

The Languages & Tools SIG is principally concerned with the DEC and public domain software products listed below. If your request directly involves one of these products, please check which one (if you have more than one request, please use a separate form for each):

☐ Debug	☐ Bliss	☐ CMS	☐ TPU	☐ C	☐ Test Manager
☐ Pascal	☐ Basic	☐ MMS	☐ EVE	☐ Ada[1]	☐ Runoff & DSR
☐ Fortran	☐ Cobol	☐ LSE	☐ EDT	☐ APL	☐ TEX & LATEX
☐ Document	☐ Dibol	☐ SCA	☐ TECO	☐ RPG	☐ Cobol Generator
☐ VAX Notes	☐ Emacs	☐ PCA	☐ PL/I	☐ Scan	☐ Software Project Mgr

If your request or suggestion doesn't relate to one of the products listed above, check which one of the following Language & Tools SIG topics it concerns:

☐ Newsletter	☐ Symposium Sessions	☐ Pre-Symposium Seminars
☐ Masters Program	☐ Working Group Activities	☐ Session Notes
☐ Information Folder	☐ SIG Tape	☐ DECUS Store Item
☐ Other L&T SIG topic:	_____	

Wish List Request—brief description: _____

_____

_____

_____

Complete description—please explain your request thoroughly; don't assume we know details of other products or services; give examples. _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Mail to: **Shava Nerad, L&T Wishlist Coordinator, MIT, 77 Mass Ave. W91-219A, Cambridge, MA 02139; (617)253-7438**

---

[1]Ada is a trademark of the DoD

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:

JUDI MANDL
UCONN HEALTH CENTER
263 FARMINGTON AVENUE, BLDG. #19
FARMINGTON, CT 06032

Title: _____

Message: _____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
____ _____

Your Name: _____
Address: _____
_____
Telephone: _____

If this is a reply to a previous DATAGRAM, what #? ___

Signature: _____ **Date:** ____

JUDI MANDL
UCONN HEALTH CENTER
263 FARMINGTON AVENUE, BLDG. #19
FARMINGTON, CT 06032

Fold Here

VAX Systems SIG Fall 1988 SIR Ballot

DECUS membership number_____(six digits)

Our site uses the following VAX cpus (check all that apply)

11/730,11/725 ☐	11/750	☐	11/780,785	☐	6200 series	☐	
8500 series ☐	8600 series	☐	8700/8800	☐	8200/8300 series	☐	
MicroVAX I,II ☐	MicroVAX 2000, VAXstation 2000	☐	MicroVAX 3000 series	☐	Other:		

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:	

I support the following as the most important System Improvement Requests. (List from zero to fifteen SIR's):

SIR Number:

I oppose the following SIR's as detrimental. (List from zero to five SIR's):

SIR Number:

Mail to:

Dave Schmidt
Management Science Associates
6565 Penn Avenue
Pittsburgh, PA 15206-4490

Reminder:
To be counted, your ballot must
be received by August 31.

VAX Systems SIG
System Imporvement Request Submission Form

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 (uese additional pages if required):

Dave Schmidt
Management Science Associates
6565 Penn Avenue
Pittsburgh, PA 15206-4490
USA

DECUS U.S. CHAPTER
SUBSCRIPTION SERVICE SIGS NEWSLETTERS ORDER FORM
(U.S. 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.
- Minimum of **$25.00** for orders placed via a credit card.

- 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 # _____

Company _____

Address _____

_____

City _____ State _____ Zip _____

Telephone # ___(_____)_____

Subscription Service Offering	Unit Price	Quantity	Total
SIGs Newsletters	$35.00	_____	_____
Spring '87 Proceedings (SP7)	15.00	_____	_____
Fall '87 Proceedings (FA7)	15.00	_____	_____
Spring '88 Proceedings (SP8)	15.00	_____	_____
Fall '88 Proceedings (FA8)	15.00	_____	_____
		Total Amount	$ _____

☐ MASTERCARD ☐ VISA ☐ DINERS CLUB/CARTE BLANCHE® ☐ AMERICAN EXPRESS

Credit Card # __ __ __ __   __ __ __ __   __ __ __ __   __ __ __ __ Expiration Date _____

I understand that there will be no refunds even if I decide to cancel my subscription.

Signature _____

FOR DIGITAL EMPLOYEES ONLY

Badge # _____ Cost Center _____

Cost Center Mgr. Name _____ Cost Center Mgr. Signature _____

MAIL TO: Subscription Service, DECUS (BP02), 219 Boston Post Road, Marlboro, MA 01752-1850, (508) 480-3659. As of July 1, 1988 the phone number will be (508) 480-3659

FOR DECUS OFFICE ONLY

Check Number _____ Bank Number _____

Amount $ _____

S&M

DECUS U. S. Chapter
Application For Membership

☐ New Membership   ☐ Update to Current Membership Profile
Current DECUS Membership Number _____

Please provide a complete mailing address, include zip code in accordance with postal regulations for your locality.

Are you an employee of Digital Equipment Corporation?
☐ Yes ☐ No

NOTE: Please print clearly or type!

Name: _____

Company: _____

Address: _____

_____

_____

City/Town/State/Zip: _____

_____

Home # (   ) _____ Business # (   ) _____

How did you learn about DECUS?
(Please check applicable item):

1 ☐ Another DECUS Member   4 ☐ Digital Sales
13 ☐ Local Users Group   2 ☐ Symposia
5 ☐ Hardware Pkg.   14 ☐ Special Interest Group
8 ☐ DECUS Chapter Office   6 ☐ Software Pkg.
7 ☐ Software Dispatch (Digital Newsletter) 10 ☐ Digital Store
12 ☐ Advertising

Do you wish to be included in mailings conducted by Digital (for marketing purposes etc.?) ☐ Permission   ☐ Refusal

Primary Business Activity at your location (check one):

Non-Computer Related
31 ☐ Manufacturing (other)
32 ☐ Agriculture, Construction
33 ☐ Energy, Mining, Oil
34 ☐ Engineering, Architecture
47 ☐ Transportation
35 ☐ Utilities
36 ☐ Government-Local, State
37 ☐ Government-Non-Military
38 ☐ Government-Military
41 ☐ Education
40 ☐ Medical or Legal Services
39 ☐ Finance, Banking, Insurance

42 ☐ Trade (wholesale, retail)
43 ☐ Research & Development
44 ☐ Leisure
45 ☐ Media
46 ☐ Other_____

Computer or DP related
25 ☐ Manufacturing (DP Equipment)
26 ☐ Software Development
27 ☐ Communications & Networking
28 ☐ Systems House, VAR/OEM
29 ☐ Consultant
30 ☐ Other DP Services

Special Interest Groups (SIGs) Enrollment:

I wish to participate in the following DECUS U.S. Chapter Special Interest Groups:

3 ☐ Artificial Intelligence   15 ☐ Networks
7 ☐ Business Applications   34 ☐ Office Automation
6 ☐ Data Management Systems   36 ☐ Personal Computer
5 ☐ DATATRIEVE/4GL   18 ☐ RSTS
8 ☐ Education   17 ☐ RSX/IAS
10 ☐ Graphics Applications   19 ☐ RT-11
11 ☐ Hardware and Micro   21 ☐ UNIX
16 ☐ Languages and Tools   26 ☐ VAX
14 ☐ MUMPS   32 ☐ Site, Management & Training
31 ☐ DAARC (Data Acquisition, Analysis, Research, and Control)

Using the classification numbers above, please indicate which SIG would be the primary focus for your interests? #

Using the classification numbers above, please indicate which SIG would be of secondary focus for your interests? #

Total Employees in entire Company/Institution/Government Department (check one):

2 ☐ 10,000 or More   6 ☐ 250 to 499
3 ☐ 5,000 to 9,999   7 ☐ 100 to 249
4 ☐ 1,000 to 4,999   8 ☐   6 to 99
5 ☐   500 to 999   9 ☐ Fewer than 6

Primary Job Function (check one):

Organization Management
11 ☐ General & Corporate
12 ☐ Financial
13 ☐ Administrative Services
14 ☐ Marketing

Computer/Systems Operations
20 ☐ Management
21 ☐ Supervisory
22 ☐ Staff

Engineering/Manufacturing
30 ☐ Management
31 ☐ Staff

Science/Research & Development
40 ☐ Management
41 ☐ Staff

Other
50 ☐ Consultant
51 ☐ Educator
52 ☐ Other _____

Citizen of the United States? ___ Yes ___ No

Country _____

Signature _____

Forward To:
Digital Equipment Computer Users Society
Membership Group
219 Boston Post Road, BP02
Marlboro, MA, 01752-1850
Phone: (508) 480-3446/3418/3659

STATUS CHANGE

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

(   ) Change of Address
(   ) Please Delete My Membership Record
        (I Do Not Wish To Remain A Member)

DECUS Membership No: _____

Name: _____

Company: _____

Address: _____

_____

State/Country: _____

Zip/Postal Code: _____

_____

Mail to: DECUS - Attn: Subscription Service
219 Boston Post Road, BP02
Marlboro, Massachusetts 01752-1850

USA

ing label
bel is not
print old
re. Include
nstallation,
university.