DECUS U.S. CHAPTER

SIGs

Your best connection for:

peer-to-peer interaction

technical exchange

NEWSLETTER

JANUARY
FEBRUARY
MARCH
APRIL
MAY
JUNE
JULY
AUGUST
SEPTEMBER
OCTOBER
NOVEMBER
DECEMBER

VOLUME 5

NUMBER 2

DECUS

GENERAL TABLE OF CONTENTS

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:

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

SIGs Newsletter,   Volume   5, Number 2   October 1989

Wombat Examiner,   Volume 11, Number 2

Election Notice

The policies and procedures of the DATATRIEVE/Fourth Generation Language Special Interest Group requires that the SIG Chair must be (re)elected every two years. The current chair, was elected at the Fall 1987 Symposium and took office at the Spring 1988 Symposium. An election, therefore, will be held at the Fall 1989 Symposium in Anaheim; the next two-year term will begin at the Spring 1990 Symposium in New Orleans.

Don Stern has declared his candidacy for re-election as SIG Chair. Other nominations may be make via DCS or at the open meeting of the SIG's Steering Committee on Sunday, November 5, 1989, at 6:00 PM in the DTR/4GL SIG Suite at the Marriott Hotel in Anaheim. Nominations will also be accepted at that time for the position of Vice-Chair of the SIG.

The elections will be conducted at the meeting of the SIG Steering Committee on Friday, November 10, 1989, at 4:00 PM, also in the Suite.

From the Chairman

Donald E. Stern, Jr., SIG Chair, Warner Lambert Co., Milford, CT

The symposium at Anaheim is rapidly approaching. The SIG Steering Committee is working very hard to make the symposium an enriching experience for all DECUS members and particularly those members with an interest in DATATRIEVE and other 4th Generation Languages.

As usual the SIG will be hosting a suite at the Marriott Hotel in which we will have Digital hardware and software. Come to the suite to find an answer to a problem, volunteer to get involved in DECUS leadership, or just to escape the rigors of symposia.

Additionally, the SIG will be sponsoring a variety of activities in a campground area. Some of these activities include meetings of the 4GL Working Groups. Members interested in Accent-R, Corvision (Application Factory), Focus, Ingres, Oracle, Powerhouse, RALLY, or Smartstar are encouraged to come to the working group meetings and get involved. Users of other 4GLs are invited to come seek out users with a common interest.

I'm looking forward to this symposium in order renew acquaintances and to make new friends. I'm always looking for new volunteers as well. If you would like to get more involved, stop by and see me.

Volunteers Needed at Symposia

Harry J. Miller, Volunteer Coordinator, Ontario, CA

Enhance your enjoyment of the Anaheim Symposium by participating in volunteer SIG Activities. Session chairs and suite hosts/hostesses are need to assist with SIG activities. Volunteers receive an appreciation gift of a much sought after Wombat Polo shirt. To participate, attend a drop-in meeting of volunteers between 5:00 and 6:00PM on Sunday, November 5, 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 10:00PM). 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 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 hardware doesn't spout legs.

In addition to the Wombat Polo shirt, the SIG will also send a "thank you" letter to the volunteer's boss if the volunteer requests it.

RALLY Working Group, Fall '89 Symposium Primer

Steven C. Fredrickson, Chairman, RALLY Working Group, Seattle, WA

Well, that time is upon us again. The DECUS Fall Symposium, this year being held in Anaheim, is shaping up to be another robust one for the DTR/4GL SIG. As chairman of the RALLY Working Group, I'd like to invite all of those users and abusers of RALLY, as well as those who are interested in learning about the product, to attend the various events scheduled for the week.

At the time of publication, the preliminary schedule was fairly well- rounded. The schedule is outlined below. I'd like to offer some highlights.

TUESDAY – November 7

2:00 – 3:00 pm	DT074	RALLY Technical Overview(DEC Presentation)
3:00 – 4:00 pm	DT073	RALLY Tutorial Forms/Reports(DEC Presentation)
4:00 – 5:00 pm	DT075	Optim. Perf. w/RALLY & Rdb/VMS(DEC Presentation)

THURSDAY – November 9

12:00 – 1:00 pm	DT061	RALLY – A Case Study (User Presentation)
1:00 – 2:00 pm	DT078	RALLY with 3GLs(DEC Presentation)
2:00 – 3:00 pm	DT048	RALLY User Panel(Users Presentation)
3:00 – 4:00 pm	DT047	RALLY Working Group(Redondo Room)
4:00 – 6:00 pm	DT049	RALLY/TEAMDATA Clinic(Problem Solving Time!)
8:00 – 10:00 pm	DT005	Wombat Magic

Tuesday, November 7th, the sessions are being presented by Digital. The first session is geared for those just looking at the product. The following two sessions provide some helpful hints and insight into application development.

Thursday, November 9th, there is a full afternoon of sessions. The first RALLY session, at noon, is given by a speaker from the Department of Transportation. Their application is a rather interesting one, and the presentation should offer some useful insights. Following at 1:00 pm is a session given by Digital on RALLY and 3GLs. If you're looking into integrating RALLY with 3GLs or vice-versa, this session is a must. The Panel Discussion at 2:00 pm brings together various RALLY users who will share with the world their actual (and often candid) viewpoints on the product. Come and hear the perspective from the 'man on the streets'.

From 3:00 – 4:00 pm is the Working Group meeting in the Redondo Room. This is the opportunity to meet other users of RALLY along with Digital personnel and discuss in an open forum your ideas and concerns. I highly recommend this meeting for current RALLY users. An added bonus this year is that the RALLY/Teamdata Clinic is scheduled right after the Working Group Meeting, from 4:00 – 6:00 pm in the DTR/4GL SIG Suite. The Clinic is your chance to go one-on-one with either a Digital developer or experienced RALLY user and get some answers to those tough-to-solve problems. Last, but not least, is Wombat Magic. Got a neat RALLY trick?. This Session is just for you. Other users like yourself get together to share some 'magic' and good times. There may even be a prize for the best RALLY Magic!

All in all, Anaheim looks like another fun and exhaustive time!. We hope to see you there, so until then ...
RALLY Ho!!

Notice to all VAX/FOCUS users, and potential users

Les Hulse, FOCUS Working Group Chair, Boston, MA

A session sponsored by all of the 4GL working groups entitled "User Panel Comparison of 4GLs" will be presented on Monday, November 6, from 11:30AM to 1:00PM. This session will present user experiences in implementing different 4GLs in different environments and will serve as a springboard for other 4GL-oriented sessions throughout the symposium.

For all users interested in FOCUS on the VAX, there will be eight sessions and a working group meeting at the fall 1989 DECUS Symposium (November 6–10). Six of the sessions will be sponsored by the DATATRIEVE/4GL SIG and one each by Business Applications and Office Automation SIGs.

A working group meeting is scheduled for 9–10PM on Monday evening in the Redondo Room. A key discussion topic will be how the working group will approach the 4GL problem presented in the September 1989 DECUS SIG Newsletter. This meeting is open to all symposium attendees with an interest in FOCUS (who are still awake at the end of the first day of sessions).

A "FOCUS Wish List" will again be available in the DATATRIEVE/4GL campground for user submissions. IBI will respond to the requests in the "Wish List" session on Thursday evening from 8–9PM. We hope to have enough time to also take questions from the floor.

Pre–Symposium Report of the Cortex Working Group

Eric S. Dubiner, Cortex Working Group Chair, Wilmington, DE

The Cortex Working Group will once again be meeting as a Birds–of–a–Feather session in Anaheim. Check the Update.Daily BOF schedule for time and place.

Since Atlanta, I have been working with Cortex, and their user group CORUS, to create a direction for the working group. I have been told that we will have at least one person from Cortex Product Engineering present at the working group meeting to answer questions and respond to comments.

We have an opportunity to present to Cortex our concern as to the the direction of the product ... so come prepared for a productive working meeting.

If you have any questions or would like to be included on the future mailings of the Working Group, please do not hesitate to contact me at the address given in the list of SIG officers in the back of the newsletter.

Accent R Tips and Techniques

Donna E. Lehman, ACCENT R Instructor, NIS

@SCN_READ_KEYSTROKE

When using @SCN_READ_KEYSTROKE in a PM or CM, it will be executed the number of times which you indicate a value to be compared to its result:

```
LEAVE IF @SCN_READ_KEYSTROKE EQ 278, 13
```

will require two RETURNs to leave this line. It will read the first key and if it isn't PF3, will wait again for the next read to see if it's 13 (a RETURN). If this is causing delays and the necessity for repeated keystrokes, rewrite the code as follows:

```
@SCN_READ_KEYSTROKE TO @INTEGER
LEAVE IF @INTEGER EQ 258, 13
```

Now the keystroke will be read once and the value checked immediately.

SCREEN PUT_CHARS

If you indicate @SCN_ERASE in a SCREEN PUT_CHARS command, all characters on that line will be erased before the new text is displayed. Specify @SCN_NOERASE if you want to keep the original text in addition to the new characters.

@FILL_ACTION SETS @SCN_TERM_CODE

If you set "END" to @FILL_ACTION, be aware that SMF will put the value of @SCN_END_CODE into @SCN_TERM_CODE before exiting the FILL FORM. If you use PF3 to terminate (258 TO @SCN_CANCEL_CODE), and your PERFORM ALWAYS AFTER routine sets "END" to @FILL_ACTION, @SCN_TERM_CODE is not going to be 258 (PF3) after the FILL has ended, but will be 26 (CTL–Z), or whatever the value you have set to @SCN_END_CODE.

DATA INDEX ON MULTIPLE FIELDS

If you usually sort and search on more than one field, define them as a combined domain in the data index:

    DOMAIN ST_CITY ON STATE,CITY

To select, a semicolon is an "AND" condition...

    FIND DOMAIN ST_CITY WHEN STATE;CITY EQ 'OHIO';'PARIS'

and is faster than

    FIND IF STATE EQ 'OHIO' AND CITY EQ 'PARIS'

CODE SEGMENTS

RELATE statements can now be specified in Code Segments (after version 10.09) to be included in PMs. Any code may be written in Code Segments and included where it would be typed into a CM, DI, PM, SD, SF or ID, not just PMs.

@AUX AND @EOF

@AUX is usually used to determine if a record was retrieved from a data set (e.g., LEAVE IF @AUX NE 'YES'). It's not the entire or only solution to the last or bad record situation, however. In addition, consider @EOF EQ 'YES' to indicate end of file and check the value of @AUX to determine what really caused the @AUX NE 'YES'. Be sure to label @EOF – there's one for each open data set.

    START:10
     GET FROM X1 NEXT RECORD HUSH
     LEAVE:10 IF @AUX NE 'YES'
     PERFORM DISPLAY_FIELDS
    REPEAT:10
    EXIT ROUTINE IF @EOF:X1 EQ 'YES'
    PERFORM FAILED_OPEN IF @AUX EQ 'OPEN'

or check for @AUX's other values (CLOSE, DUP, EMPTY, LOCK, MISSI, etc.). The "last record" read might not be the last record.

Quiz

What is the result of the following?

    *USE DS DEC
    *SELECT WITH NISINC ON VAX SHOW ACCENT(R)

    —
    —
    —
    —

    The DBL is as follows...
    SD NISINC: SD DEC:
    0001. VAX,C,2   0001   VAX,C,2
    0002. R,I,2   0002   ACCENT,C,6,OCCURS 4

    DS NISINC: DS DEC:
    X1 2   X1 A   B   C   D
    X2 1   X2 E   F   G   H
    X3 4   X3 P   Q   R   S
    X4 1   X4 T   U   V   W

Dear Wombat Wizard

Dear Wombat Wizard:

We deal with data which can have a large dynamic range so we use USAGE IS REAL data type. Sometimes when we store data into a record we can not retrieve that record with an EQUALS RSE (record selection expression). The problem most often occurs with the values 0.3 and 0.6, but I suspect that other values will exhibit the anomaly as well. Why can't we retrieve the data this way and how can be work around this problem?

Signed,

Have a REAL Problem

Dear REAL Problem:

This is a REALly interesting problem; it is one related to the representation of the data; and similar problems occur in almost every computing language. However, this particular one is rather unique to VAX-DATATRIEVE.

So that everyone can follow the details, I'll construct a test domain upon which we can show this particular problem. Consider:

```
DEFINE DOMAIN TEST USING TEST-RECORD ON TEST.DAT;
DEFINE RECORD TEST-RECORD USING
01 TEST-REC.
    03 REAL-FIELD USAGE IS REAL.
;
```

Then use the following procedure to put some test data into the domain.

```
DEFINE PROCEDURE LOADIT
DEFINE FILE FOR TEST;
READY TEST WRITE
DECLARE X PIC 9V99.
X = 0.0
REPEAT 100 BEGIN

    X = X + 0.01
    STORE TEST USING REAL-FIELD = X
    END
END-PROCEDURE
```

Then, see which records we can retrieve with the procedure

```
DEFINE PROCEDURE TESTIT
DECLARE COUNTER USAGE IS INTEGER.
DECLARE X PIC 9V99 EDIT-STRING IS 9.99 .
READY TEST SHARED READ
X = 0
REPEAT 100 BEGIN
    X = X + 0.01
    COUNTER = 0
    FOR TEST WITH REAL-FIELD = X
        COUNTER = COUNTER + 1
    IF (COUNTER EQ 0) THEN PRINT X
    END
END-PROCEDURE
```

When we execute TESTIT, we get

```
DTR> :TESTIT
```

```
X
0.15
0.23
0.27
0.30
0.43
0.46
0.49
0.53
0.54
0.59
0.60
0.67
0.73
0.86
0.92
0.98
0.99
```

Thus, we find **seventeen numbers** which we **stored which we can not retrieve with the same value we used in the STORE!** If we change the data type on the field REAL–FIELD to USAGE IS DOUBLE, USAGE IS G–FLOATING, USAGE IS H–FLOATING, or PIC 9V99 we can retrieve all of the records. However, if we try the procedure TESTIT in one of the version of DATATRIEVE–11 we can retrieve **none** of the records!!

The root of the problem is how data is stored and compared in one or more of the many possible pairs of data types. Floating point data types such as USAGE IS REAL are represented and stored as a signed fraction (a number between –1 and +1) which is called the mantissa times a power of two which is called the exponent. [More precisely, a REAL datum is four contiguous bytes starting on an arbitrary byte boundary. Bits are labeled from the right, 0 through 31. The form of a REAL datum is sign magnitude, with bit 15 the sign bit, bits 14:7 an excess 128 binary exponent, and bits 6:0 and 31:16 a normalized 24–bit fraction with the redundant most significant fraction bit not represented.]

Thus, the representation of any number which is not the sum of powers of twos is an approximate one when stored in a finite number of bits. We have exactly the same problem in representing one third in finite number of decimal digits as in

$$1/3 = 0.33333...$$

When DATATRIEVE encounters a record selection expression like

```
WITH REAL-FIELD = STRING-VALUE
```

DATATRIEVE tries to convert the value on the left side of the equals sign and the value of the right side of the equals sign to a common data type with equivalent precision. This works if the right side is a STRING data type and the left side is DOUBLE, G–FLOATING, H–FLOATING, or STRING, but sometimes fails if the left side is REAL. However, if the Boolean comparison involves ANY KIND OF A CALCULATION such as

```
WITH REAL-FIELD = CALCULATE-VALUE
```

both sides are converted to H–FLOATING data type (the type with the most precision and greatest dynamic range). [It will be left for the reader to show that if the procedure TESTIT is modified by

```
FOR TEST WITH REAL-FIELD = 1.0 * X
```

the Boolean fails on a different and larger set of cases!]

If you this this behavior is strange, consider the affect of trailing zeros (implied precision) in the right side string value with

```
DTR> find test with real-field = 0.3
[0 records found]

DTR> find test with real-field = 0.30
[0 records found]

    .  ! keep trying with more and more trailing zeros
```

```
                .
                .
DTR> find test with real-field = 0.3000000000000000
[0 records found]

DTR> find test with real-field = 0.30000000000000000
[1 record found]

DTR> find test with real-field = 0.300000000000000000
[1 record found]

DTR> find test with real-field = 0.3000000000000000000
[0 records found]

DTR> find test with real-field = 0.30000000000000000000
Data conversion overflow.
[0 records found]
```

Clearly, at least clearly to the Wizard, this is a problem in the precision of the data in the EQUALS RSE. This problem also should be looked at a bit more carefully by the VAX-DATATRIEVE developers see if something more can be done to make this behave a bit more predictably.

There is one fairly obvious way (at least, it is obvious to the Wizard) to force DATATRIEVE to use the same precision on both sides of the record selection expression. This can be accomplished with the FORMAT USING construct. An example of this would be

```
WITH FORMAT REAL-FIELD USING 9.99 = STRING-VALUE
```

This would cause DATATRIEVE to do the comparison as strings. However, this would not work for the full dynamic range of data that could be stored in the real field. If the value on the right side of the RSE is prompted for, one could format both sides like

```
WITH FORMAT REAL-FIELD USING 99.99 = FORMAT *."value" USING 99.99
```

Another way would be to use a global variable which has exactly the same data type as the field used in the RSE. Consider

```
DECLARE TEMP-REAL USAGE IS REAL.
TEMP-REAL = *."value"
READY TEST
FIND TEST WITH REAL-FIELD = TEMP-REAL
```

In this case, the FIND will always retrieve the desired records since the data conversions AT THE STORE and AT THE RETRIEVE are exactly the same.

One of the best features of the DATATRIEVE is the automatic data conversion that it does behind the scenes – out of sight of the user. This is especially true for conversion of strings to and from numbers and the conversion of date data type. However, there are situations when the user, weather a novice wombat breeder or a wizard, must keep in mind what is really going on at the level of bits and bytes to produce the desired results. However, if you want to avoid this problem and can afford the space (4 extra bytes), you might want to use USAGE IS DOUBLE rather that USAGE IS REAL.

Signed,

The Wombat Wizard (WDP&SH&JG)

Part 3 of Spring '89 Wombat Magic will appear next month.

Special RALLY Product Improvement Request

John L. Henning, Digital Counterpart, Nashua, NH
T. Chris Wool, PIR Coordinator, du Pont, Newark, DE

Digital seeks feedback from users of RALLY. What improvements would you like to see in future versions?. In particular, we encourage you to use the DTR/4GL SIG Product Improvement Request (PIR) process. RALLY Engineering has some specific PIRs for which we request feedback. If you are a RALLY user, please read the attached lists of suggestions, and give us your feedback by voting in the PIR process. In addition, you can discuss the fine points of the suggestions by:

- Entering a REPLY to one of the Notes on DECUServe dealing with RALLY
- Attending Fall DECUS and discussing them with the Digital representatives in the DTR/4GL Suite.

DECUServe provides an electronic conferencing system for DECUS members; subscription information is available by calling 508-480-3418 or writing

> DECUS – DECUServe Processing
> 219 Boston Post Road, BP02
> Marlboro, MA. 01752-4605

Discussion of RALLY primarily occurs within the DATATRIEVE conference, since RALLY is one of the products covered by the DTR/4GL SIG. Please see Notes 57-62 within the DATATRIEVE conference.

* * * INSTRUCTIONS * * *

A "point allocation" scheme will be used to determine your likes and dislikes for a particular product improvement request. You have a total of 50 points with which to vote. You may allocate either positively (in favor of a PIR) or negatively (against a PIR). The number of points you allocate to a particular PIR indicates how strongly you feel about it. In order to assure a wide range of choices, however, you may not allocate more than 10 points (positive or negative) to any one PIR and the absolute value of the total may not exceed 50 points.

Use the ballot which is in the Questionnaire Section in the back of the newsletter. You may copy the ballot if you prefer rather than removing a page from your newsletter. You may submit your ballot electronically on DECUServe (deadline December 4); you may deliver your ballot in person to the DTR/4GL SIG Suite or to one of the RALLY Working Group offices at the Anaheim Symposium (deadline November 10); or you may mail your ballot to Chris Wool at the address indicated on the PIR ballot (**deadline for receipt of mailed ballots is December 15**).

NOTE:
The only commitment RALLY Engineering has is to respond in writing to the top ten PIRs in the overall ranking, for publication in the Wombat Examiner and 4GL Dispatch. The submission of PIRs by RALLY Engineering in no way implies or states a commitment by RALLY Engineering to implement the proposed PIR.

End-user features:

RALLY developers create applications which are run by end-users. The PIRs below would provide additional features for application end users:

F89-1

Abstract: Provide 'Undelete record' command

Discussion: RALLY applications allow users to delete a record with a single keystroke. If autocommit is in effect, the user might delete the wrong record without the ability to recover. This option would allow users to restore the deleted record. [Nominations for what keystroke should be bound to "undelete record" will be accepted on DECUServe, at DATATRIEVE note #58].

F89-2

Abstract: Require confirmation before delete

Discussion: The RALLY Guide to Application Development explains how to attach a "confirm before delete" option to a form. This PIR suggests that the application definer be able to select this feature simply by checking a box on the form definition. [For discussion at note 58: should the confirmation message be customizable by the application definer?]

F89-3

Abstract: Remember previous query criteria

Discussion: Sometimes the user wants to follow up a query with a slightly different query. This option would cause the application to restore the previous query criteria when re-entering query mode, so the user could then edit to make changes. [Again, DECUServe can be used to discuss fine points, such as whether this behavior should become the default.]

F89-4

Abstract: Allow 'next/previous record' to cross group boundaries

Discussion: RALLY form/reports consist of "groups". Navigation commands (such as arrow keys) generally move only within the current group, which means that the the user should understand the form/report group structure. An alternative navigation model would hide the group structure from the end user by allowing next/previous record commands to cross group boundaries.

F89-5

Abstract: Display current mode

Discussion: RALLY Form/Reports can be used in various modes (browse, insert, update, query ...). This PIR suggests that RALLY prominently display the current mode, to help the users understand their current state. [Note: Screen space would be required; suggestions about where to place the mode display are welcome on DECUServe at DATATRIEVE note 58.]

F89-6

Abstract: Step-by-step queries

Discussion: This feature would provide the application user with very specific step-by-step instructions when performing a query, automatically giving hints at each step of entering the mode, moving to a field, typing in a valid query expression, and executing the query.

Report features:

F89-7

Abstract: Adjust size of form/report to current environment

Discussion: Sometimes people wish to define a single form/report for use on different sized devices (for example, a 24-line VT330 screen, a 48-line DECterm window, and a 60-line piece of paper). RALLY lets you do this, but each form/report has a single "page length" which is used in all situations. This can result in paper reports with lots of white space, or reports that you can only see part of on the screen (until you scroll them).

This proposal would add an option to adjust form/reports to the size of the calling RALLY task.

F89-8

Abstract: Character string formatting

Discussion: RALLY currently provides the ability to format numbers and date strings on output, but not character strings. This option would provide output formatting of character strings.

F89-9

Abstract:	Set terminal width from RALLY
Discussion:	RALLY tasks can be of arbitrary width, and are formatted to fit the environment in which they are run. For example, a 132 column report can be run on an 80-column terminal, and the user can scroll through it. If that same report (and task) were run on a 132-column terminal (or DECterm window), the user need not scroll at all.
	At this time, RALLY makes no attempt to re-size the device it is using. The proposed feature would provide some method for changing the device size from RALLY. [DECUServe discussion of details is welcome, such as whether resize should be automatic on invocation of a new task, or done explicitly by an ADL function. Please enter your opinions at DATATRIEVE note 59]

F89-10

Abstract:	Don't start parent record on a separate page from children
Discussion:	This option would provide better widow/orphan control by not starting a parent record unless at least some of its children will also fit on the same page.

Menu features:

F89-11

Abstract:	Extend Menu Builder to include All-in-1 style menus
Discussion:	RALLY menus can look like All-in-1 style menus today, but only after the developer edits them. The default style is similar to VAX TEAMDATA. This option would provide another default style, similar to ALL-in-1.

F89-12

Abstract:	Display menu choices according to authorized access
Discussion:	RALLY allows the application definer to associate passwords with menu choices. This PIR suggests that both selection and display of menu choices should be controlled through a protection scheme based upon authorized user access.

Form features:

F89-13

Abstract:	Enhance autocommit features in parent-child situations
Discussion:	"Autocommit" causes changes to be applied to the database as soon as the user leaves the current record. This definition implies that when the user moves the cursor from a parent record to a child record, the parent is committed. The proposed enhancement would defer committing until the user moves to a higher group, or moves to another record in the parent group. No commits would occur when the user moves back and forth between a parent record and its children.

F89-14

Abstract:	Highlight current field
Discussion:	This feature would allow highlighting of the current field. [Should a highlighted current field remain highlighted when the user moves the cursor over to the List Of Values?. Opinions welcome on DECUServe, at DATATRIEVE note 61.]

F89-15

Abstract:	Provide more action sites
Discussion:	RALLY provides a wide variety of "action sites," where the application developer can add additional processing or affect flow of control. This feature would further expand the set of available sites, for example, before/after visiting record, before/after performing query, after user presses INSERT key. [To discuss what specific action sites are needed, please reply to DATATRIEVE note 61 on DECUServe.]

F89-16

Abstract: Ability for ADL to move the cursor to a given field

Discussion: ADL procedures are able to move the cursor from one field to the next (or previous field). This feature would allow a procedure to move the cursor directly to a specific named field.

F89-17

Abstract: Ability to set keypad to numeric or "application" mode

Discussion: Data entry applications sometimes require the keypad to be in numeric mode. This PIR would add an option in ADL and/or a command available to the end user that changes the mode of the keypad.

F89-18

Abstract: Multiple legends for a field

Discussion: RALLY currently allows the definer to specify text that pops up whenever the user visits a given field. This is a popular feature. A proposed enhancement would provide the ability to specify more than a single legend for a given field, depending on the circumstances. [For discussion: how should the legend be selected?. According to the mode currently in effect?. By an ADL procedure? Other?]

F89-19

Abstract: Option to make fields required in query mode

Discussion: Some applications want to require that, if the user does a query, the user must query on a certain field, for example, to reduce the number of records retrieved. Note: a new action site "before executing query" would allow definers to get the effect of this feature, although by writing ADL procedures.

F89-20

Abstract: Option to suppress LOV validation in query mode

Discussion: Some applications have stricter restrictions on data entry than on query. This option would skip List of Values Validation in query mode.

F89-21

Abstract: Trap broadcast messages

Discussion: This feature would trap incoming broadcast messages and present them in an orderly fashion.

F89-22

Abstract: Ability to vary the text of the working message

Discussion: Whenever RALLY applications perform a long operation, RALLY automatically displays a "Working" message. This option would allow application definers to define what text should be displayed, rather than just "working", and thereby give a more informative message.

Definition time enhancements:

F89-23

Abstract: DCL object

Discussion: RALLY applications can execute DCL commands by an External Link to call LIB$SPAWN. This option would allow the definer to directly specify a DCL command that is to be performed (for example at an action site) thus providing the convenience of not needing to define the external link.

F89-24

Abstract: Ability to turn off Integrity Checking

Discussion: RALLY applications are checked for correctness on a fairly frequent basis, for example as edits are completed to a form/report. Frequent integrity checking has the advantage of giving the application definer feedback about problems in a timely fashion, but for large applications integrity checking consumes significant time.

This proposal would allow the definer to effectively say "I'm an expert – don't bother checking what I do until I tell you to.". Of course, the user of this feature would have to accept the fact that when he finally does verify his application, there may be a great many errors to report.

F89-25

Abstract: LSE for editing ADL

Discussion: This feature would make it possible to use LSE and LSE templates when editing ADL procedures.

F89-26

Abstract: Run–time variables from DML

Discussion: A powerful feature of RALLY V2 is the ability to specify expressions that are evaluated at run–time and which restrict selection of records for a data source definition (DSD) when it is opened by a form/report. This proposal would extend the feature to also be available for DSDs opened through ADL's Data Manipulation Language.

F89-27

Abstract: Option for the definer to turn on/off legends

Discussion: RALLY "legends" provide incremental help messages to the user, and are a well–received feature for providing a friendly application environment. However, expert users may prefer to save the time required for screen painting legends. This option would provide a way for the definer to turn off display of all legends.

F89-28

Abstract: Extend RALLY UPDATE to handle renaming of database fields

Discussion: The RALLY UPDATE utility will automatically make certain changes to all Data Source Definitions based upon changes in the underlying database – for example, if the field is changed from text to numeric or the size is changed. But if a field is renamed in the database (e.g. EMP_NAME is deleted and EMPLOYEE_NAME is created), the definer must edit each DSD to reflect the change. This proposal would extend RALLY UPDATE to handle the case of field renaming.

F89-29

Abstract: Extend use of CDD/Plus

Discussion: RALLY makes significant use of CDD/Plus today; but there is room for extension, for example by extending impact analysis to additional RALLY objects or using additional attributes that are defined in the dictionary. [Discussion topic for DECUServe: what specific extensions are most important?. See DATATRIEVE note 62.]

F89-30

Abstract: Provide LOV "Starts With" for definition time

Discussion: A new feature of RALLY V2.1 when using Rdb databases allows the end–user to type in the first few characters of a field, press Gold–Select, and see a list of choices that start with the entered characters. This PIR suggests providing similar capability within the RALLY Definition Environment.
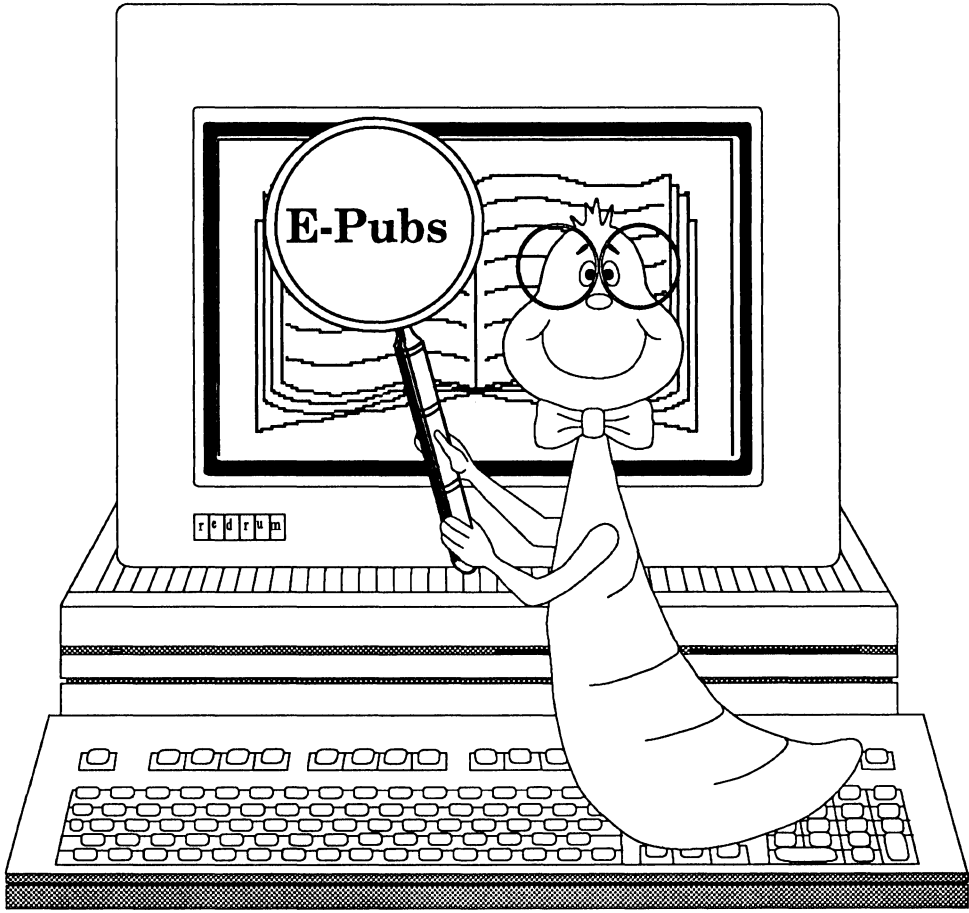
FOCUS on Electronic Publishing
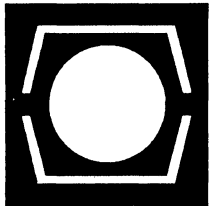
DECUS

in _FOCUS_

Submission Rules

Contributions of articles, letters to the editor, etc. are solicited and gladly accepted. Submissions can be directed to the editor as follows:

Richard Wolff
Bonneville Power Admin.
Routing SWHP
PO Box 3621
Portland, OR 97208

(503) 230-5894   (voice)
(503) 230-5316   (fax)

Editorial Policy

Editorials, letters to the editor and articles in this newsletter are solely the opinions of the authors and do not necessarily reflect the official views of the Digital Equipment Computer Users Society, Digital Equipment Corporation, or the authors' employers.

The Editor's Screen

Richard Wolff

Welcome to the third issue of _Focus_. I'm sure the time will come when the issue number will be of little importance but, since I breathe a sigh of relief as each publishing milestone is reached, the issue number still means something to me. In any case, welcome back for another installment.

This issue begins with Kevin Kindschuh's comments "From the Chair" which introduce our latest recruits from within Digital. Their participation in the E-Pubs UIG underscores DEC's committment to the area of electronic publishing. Please join me along with Kevin and the other E-Pubs members in welcoming these new participants into the E-Pubs family.

Besides serving as the E-Pubs vice-chair, Bill Koppelman also coordinates our working groups. He presents the current and evolving breakdown of our sub-interests with his "Working Group News."

My involvement in electronic writing goes back quite a ways. Before my first encounter with Runoff on a VAX, I used a product called UNADS on a Sperry Univac mainframe. Yet, I'm just a student regarding electronic publishing. I have listed some of my sources from my studies. I hope you find these books and magazines as useful as I have.

Finally, I would like to focus your attention to the upcoming Fall Symposium in Anaheim. The November event will include over 40 sessions covering products like TeX, DECwrite and Interleaf as well as topics detailing user efforts in electronic publishing and the strategic directions for DEC and third parties in this arena. We hope to see you there.

From the Chair

Kevin Kindschuh

E-Pubs is fortunate to be expanding our group! Several changes have occurred and are reflected in our steering committee membership list.

Don Hedman is now our second Digital Counterpart. He is CDA Program Product Manager and works for the Software Development Technologies group's Core Applications subgroup (all part of Central Engineering/ Spitbrook). Core Applications include DECwrite and other products based on CDA/DDIF/etc.

Don is anxious to make sure we have the engineering contacts and dialogue we need within Digital. Having an engineering counterpart will complement nicely the marketing relationship with Business and Office Information Systems we enjoy with Cathy St. Martin. We're pleased to have him, and honored that Engineering would initiate this relationship!

Marian Weisenfeld is a Digital Contact for us in Software Business Technologies. She is the Bookreader Product Manager. SBT also manages the License Management Facility and CDROM technologies for software distribution and publishing. She is also a VAX SIG counterpart and has done work with the L&T SIG. She has expressed her willingness to assist with the E-PUBS group in the future. I had the pleasure of meeting her at the SIG Council/Countparts meeting recently.

Rick Landau is a Digital Contact for us in the areas of printer technologies and PostScript. He is also a Counterpart for GAPSIG. I met him also at the SIG Council meeting. He is very excited about E-Pubs and anxious to help in his areas.

These four: Cathy, Don, Marian and Rick will provide E-Pubs with a broad range of contacts and a great deal of support from Digital. Lynn Jarrett who was our Mentor for a while and was PC SIG Chair has taken a position in Leadership Development. We wish Lynn success in her new efforts.

Note that we still need a Seminar Rep, a Store Rep/Special Effect Coordinator, and chairs for the DECwrite WG and other Working Groups. Please contact me if you are interested in any of these positions or in participating in any other capacity.

Working Group News

William Koppelman

Working Groups have grown from virtually every SIG to address special interest topics in a more focused forum. They appear as individuals network at DECUS Symposia, and other technical seminars, and find a common interest and bond in sharing information. They form within a SIG that has an interest in or responsibility for their issues and look to that SIG's Steering Committee for the nurturing and guidance to grow.

Working Groups may be product or topic specific, ie. languages, systems or applications, or they may of focused interest, ie. Newspaper or Magazine publishing. They SIG Council has recognized the valuable contributions Working Groups make in the development and growth of DECUS and the individual SIGs. As a result, they prepared a "White Paper" dealing with the "Care and Feeding of Working Groups." This is an excellent guide for all Working Groups, special interest committees or individuals interested in further developing special interests.

Currently the Electronic Publishing UIG is coordinating the activities of the following Working Groups:

     Interleaf W/G
     TEX/LATEX/WEB W/G
     DECWrite W/G

We also have interest in or a working relationships with:

     Videotex W/G
     Tech Prod of Doc W/G

Additional interest has been expressed in developing Working Groups for:

     Datalogics Users
     Newspapers
     DECPage Users

In short, our Working Groups will continue to be the focus of supporting activities to the E-PUB's mission. They have a formal structure of their own which enhances the value of the sessions and feedback they provide to the E-PUBs Steering Committee.

We will profile each Working Group in future articles to acquaint everyone with the special interests that are served and the individuals who guide their activities.

Post Script

Richard Wolff

This newsletter is produced on an Apple Macintosh II using Aldus Pagemaker, Deneba Software's Canvas and Claris Corporation's MacDraw II and MacWrite. I look forward to producing future issues on a VAXstation 3100 with a comparable set of software featuring DECwrite. In the mean time, I hope you'll overlook any appearance of indiscetion or sacrilege on my part.

Sources

For information about electronic/ desktop publishing, refer to the following books and publications:

Daniel Makuta and William Lawerance, *Complete Desktop Publisher*, Greensboro, NC; Compute! Publications, Inc., 1988

Tom Lichty, *Design Principles for Desktop Publishers*. Glenview, IL; Scott, Foresman and Company, 1989

Publish!, San Francisco, CA 94107; PCW Communications, Inc.

Personal Publishing, Carol Stream, IL 60188; Hitchcock Publishing Co.

ITC Desktop, New York, NY 10017 International Typeface Corporation.

Desktop Publishing Buyer's Guide and Handbook, New York, NY 10011; Bedford Communications, Incorporated.

Steve Lambert and Suzanne Ropiequet (editors), *CD ROM: The New Papyrus*, Redmond, WA 98073; Microsoft Press, 1986.

Suzanne Ropiequet (editor), *CD ROM 2: Optical Publishing*, Redmond, WA 98073; Microsoft Press, 1987.

CD-ROM Review, Peterborough, NH 03458; CW Communications/ Peterborough Inc.

Computer Pictures, Torrance, CA 90505; Montage Publishing, Inc.

Training on the more popular desktop publishing tools as well as on various aspects of desktop design is becoming available at colleges and universities. Private classes and seminars are another important source for this type of training.
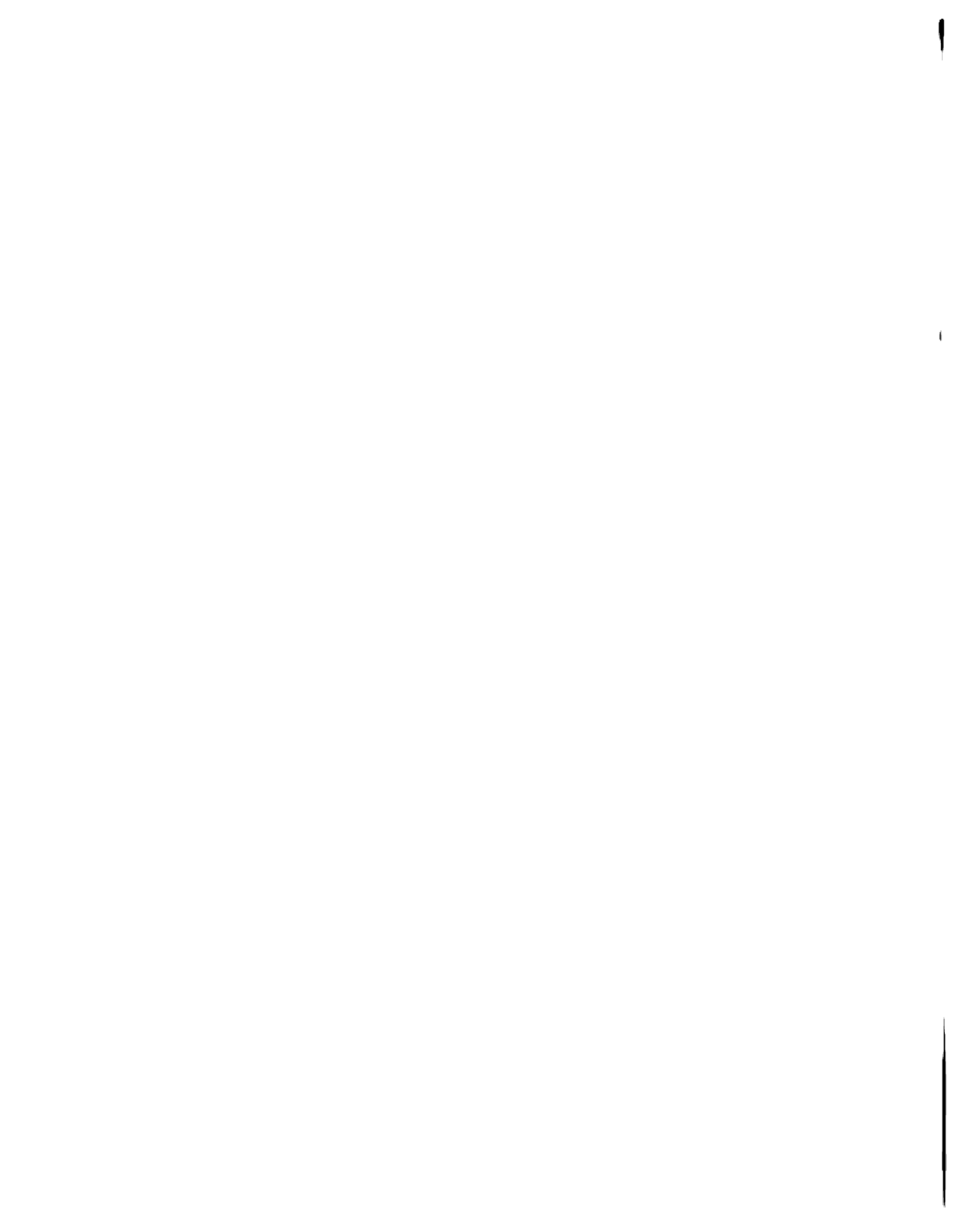
The Graph Paper

October, 1989

VAXimage overview

Geoffrey Ward, Digital Equipment Corporation

(The following article was transcribed by Bob Hays from a combination of the viewgraphs and audio cassette of the talk given by Geoffrey Ward at the Spring 1989 Symposium)

The VAXimage program recently announced by Digital provides a standard, device independent method for image acquisition and processing. Today, VAXimage is divided into three parts:

      a.   VAXimage Applications Services (VAS)
      b.   VAXimage Scanning Subsystem (MD300)
      c.   VAXimage Scan Application (VSA)

The VAS supplies software tools to build image applications. The MD300 supplies an image scanner with an interface to a QBUS-based VAX system. VSA is an application to control the MD300.

The VAXimage Applications Services are library resident routines for image acquisition, manipulation and display that use the VMS native mode call interface to provide language independence. In addition, the library conforms to industry and national standards such as the CCITT G3 and G4 image data compression standard. VAS is divided into three parts as shown in Figure 1. The application services are designed for use by software developers ranging from those creating internal applications to sophisticated end users. Because VAS provides a robust, consistent model for image processing, applications that desire consistent image data representation, treatment and interchange within the Digital domain are candidates for using the VAXimage Applications Services.

mailing address

Robert L. Hays
3621 South State Road
Ann Arbor, MI 48106
(313) 769-8500 x. 458

publication info

This newsletter is prepared using Mass-11 and Mass-11 Draw from MEC on VAXes and VAXstations, Illustrator from Adobe, MacPaint II from Claris and a VersaScan scanning subsystem on various Macintoshes (file transfer courtesy of PacerLink software and Kinetics FastPath hardware), and is printed on an LN03R from our VAXcluster using the PostScript page description language from Adobe.

submissions

Articles, copies of viewgraphs, tips and tricks, and graphics output can be submitted to the GAPSIG newsletter; here's how YOU can make submissions:

1) Send 1600 or 6250 BPI tape in either ASCII or Mass-11 (TM) format. Include a letter with your name and address, and please send any charts or graphics in hard copy form.

2) Send hard copy.

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

editorial policy

This editor has a simple editorial policy: we print our own views (from the editor and from the chair's desk), letters to the editor, and articles submitted by graphics users. If you don't agree with something printed here, mail your letter to the editor at the address at the top of this column; don't use expletives and don't list pricing or delivery information. We are here to serve the DEC graphics community, so please contact us with any comments, praise, or, well, yes, criticism. We welcome your inputs!

Subscriptions

Subscription information is available at the end of the magazine.
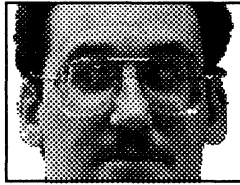
from the editor

Robert Hays
P. O. Box 1567
3621 South State Road
Ann Arbor, MI 48106
(313) 769-8500 x458

Your editor is busy doing work (oh no, Mister Bill!). Yep, it's DECwindows time at the ranch; we're building a prototype data acquisition system with a DECwindows front end. Hopefully, I'll find some time to set down some of my, er, "experiences" here in these pages. For now, though, you'll have to be happy with some other information on DECwindows and on VAXimage.

The Fall Symposium is coming up fast. I should know, I'm almost done preparing my three sessions (two are brand new ones that, given enough time, will result in Proceedings papers). I expect the Fall Symposium to be lots of fun and excitement, especially since the GAPSIG plans to hold a number of important, fun events.

Everyone on the steering committee is working hard to bring together our plans for celebrating our tenth anniversary as a working SIG. Plans include a special folder of materials, an exciting exhibit pointing out the special marks of the last ten years in Digital graphics, and perhaps a very special button for all you button collectors.... There will be a special celebration on Thursday night to watch for. And, of course, we will have all the new Digital equipment in the campground ready for hands-on demonstrations of the power of the new graphics desktop in action.

Now that I have a VAXstation 3100 on my desk, I can attest to the value of a multi-processing window system; my software development throughput has really increased since DECwindows came to my desk! The tools provided really help with my daily balancing act. Now, if I could only drag files from FileView to the LSE editor....

That's all for now; come back at the same bat-time next month for more hot stuff from the GAPSIG!

VAXimage overview

(Cont'd from p.1)
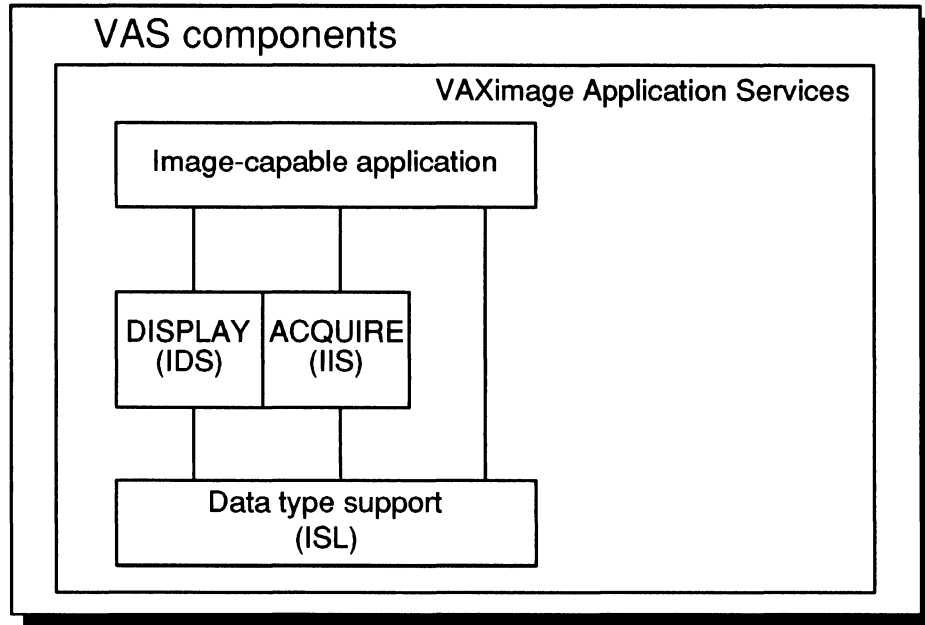
Figure 1. VAS is made up of three components which work together to provide a rich set of image services.

The Image Display Services (IDS) are routines for rendition and presentation of images to display (output) devices. The library uses a device independent interface which allows for new output devices to be added at a later date. Also provided are static and panned DECwindows image widgets to speed up development of DECwindows display programs.

IDS supports the DECwindows toolkit, Xlib programming and various hardcopy environments. The operational model is shown in Figure 2. IDS works by providing a device independent bi-tonal rendition model. This model matches image attributes to device characteristics based on attributes defined by the Image Support Library (see below). Predefined templates for Digital hardcopy devices are provided, and tools are available for user defined devices; currently, SiXEL, PostScript and Ximage templates are provided. Renditions can be decompressed, scaled and rotated. There are two widgets provided for DECwindows support: static and panned.

The Image Services Library (ISL) supplies routines for manipulation of images. The key concept used thoughout is the idea of an image data type. This data type combines image attributes and data into an abstract application type, much as the FORTRAN RECORD extension provides for abstracted data aggregates.
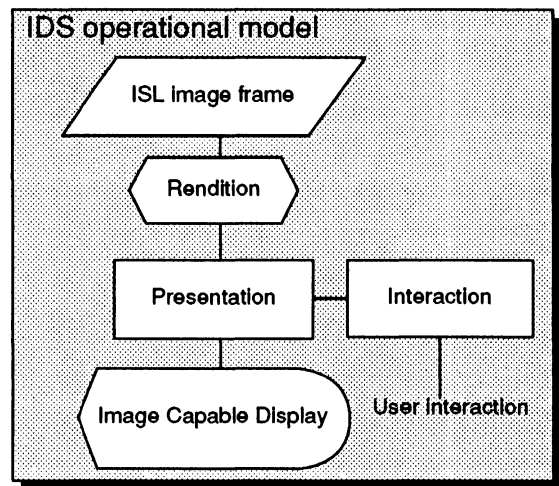
Figure 2. The IDS operational model provides flexibility and modularity.

VAXimage overview

(Con't from p. 3, c. 1)

The image data type is based on the DDIF industry standard. Various attributes can be assigned to images; see Figure 3 for more information.

Image attributes	
DDIF FRAME ATTRIBUTES	Size and position of DDIF frame
USER ATTRIBUTES	Application specific information
PRESENTATION ATTRIBUTES	Describe how image is rendered
CODING ATTRIBUTES	Describe how image is encoded

Figure 3. A number of attributes are available to control imaging with VAXimage.

There are five groups of routines in the ISL:

- a. Frame services
- b. Region of interest
- c. Formatting services
- d. Domain adjustment
- e. Processing

The frame services provide control of image frames in memory. Deletion, creation, import and export of frames are provided for. Also, image attributes can be set and changed on individual frames.

The ROI services allow for the creation and management of regions of interest, or segments of an image that can be operated on. Currently all ROIs are treated as rectangles, but eventually chain codes and polylines will be supported; currently, the farthest boundaries of both chain codes and polylines are evaluated and a rectangular ROI is used.

Formatting services provide support for import and export of DDIF images though file access support, stream data support and multiframe import/export. SiXEL encoding and Encapsulated PostScript (EPS) encoding are supported. Data compression using CCITT standards are provided.

The domain adjustment functions allow for copying, scaling or rotating images using single precision floating point factors.

The processing portion of ISL provides a combine function that has BITBLT capability (AND/NOT/OR bitmap comparisons) that can optionally use ROI information and also allows for an 8 x 8 mask bit pattern.

The Image Input Services (IIS) provide network-compatible device support for the MD300 scanner. Contrast, brightness, resolution, scan mode, frame and unit of measure controls are provided though the library. Image data can be read either synchronously or asynchronously.

The IIS is designed to insulate applications from future VAS improvements and device additions by providing a device/operating system independent interface to image acquisition devices.

```
#define BUF_SIZE 1500000

int scan_dev_id;
int fid, new_fid;
unsigned char buffer[BUF_SIZE];
int buffer_size = BUF_SIZE;

$DESCRIPTOR(scan_dev,"LDA0:");
/* Assign the scanner, reset attributes to default */
scan_dev_id = IIS$ASSIGN ( &scan_dev );
IIS$RESET_ATTRIBUTES ( scan_dev_id );

/* Scan the image into the ISL frame */
fid = IIS$READ_FRAME ( scan_dev_id, buffer, buffer_size, iosb );
```

Figure 4. This code fragment shows how easy it is to use the IIS to grab images from the MD300 scanning subsystem.

Figure 4 provides a C source code fragment that uses Image Input Services (IIS) to capture an image. Note the call to IIS$RESET_ATTRIBUTES, which ensures that the scanner is set correctly before taking an image. This is very important in a network environment where another user may have left the machine in an unknown state. IIS$READ_FRAME actually grabs the image from the scanner.

Figure 5 is an example of using the ISL for processing an image; in this example an image is rotated forty-five degrees. IMG$ROTATE performs the rotation and creates a new image frame in memory. IMG$DELETE_FRAME erases the original image from memory which reclaims the virtual memory used by the original image. Figure 6 is an example of a user action routine. Figure 7 is a code fragment that exports a DDIF file of the current image identified by the context variable ctx.

Figure 8 uses the IDS to display an image to a DECwindows device; IdsPannedImage provides support for panning the image inside the window.

VAXimage overview

(Cont'd from p. 4, c. 2)

```
int cmprs_type;
int num_lines;
int num_pixels;
/* item list for GET_FRAME_ATTRIBUTES */
struct GET_ITMLST itmlst[]=
{
{IMG$_COMPRESSION_TYPE,&sizeof(int),&cmprs_type,0,0},
{IMG$_NUMBER_OF_LINES,&sizeof(int),&num_lines,0,0},
{IMG$_PIXELS_PER_LINE,&sizeof(int),&num_pixels,0,0},
};
/* read back attributes of image frame */
IMG$GET_FRAME_ATTRIBUTES ( fid, itmlst );
/* decompress frame (if compressed) */
if (cmprs_type != IMG$K_PCM_COMPRESSION )
    IMG$DECOMPRESS ( fid );
/* rotate the frame 45 degrees */
angle = 45.0;
new_fid = IMG$ROTATE ( fid, &angle, 0, 0, 0 );
/* clean up old frame if not needed */
IMG$DELETE_FRAME ( fid );
```

Figure 5. ISL provides a number of display control procedures which save users from writing their own and provides a standard format for image display control.

There are five major areas where performance of the library routines can be improved. First, in terms of processing time, use the NEAREST_NEIGHBOR flag for scaling and rotating continuous tone images. User-written action routines for I/O can provide speed improvements. Use simple image compression techniques like Group 3-1D instead of 2-D. Use of the CCITT compression techniques for bi-tonal images can improve disk utilization, and in the future DCT compression should be available to provide continuous tone compression. Use the SERIAL_BINARY_ENCODING in PostScript to make smaller PostScript files; this also speeds up printing time. To improve memory usage, delete unneeded frames when finished with them. Use user-action routines to import and export data, and, when using the scanner, use the lowest resolution required for a job.

Some VMS tuning will also improve performance. Since the image frames use memory, paging and pagefile size problems are most likely. The parameters to consider first are listed in Figure 9.

```
char buffer(512); /* export buffer */
char filename[]="test.ddif";
int ctx; /* stream context */
/* create user RAB using user-written routine */
user_RAB = user_rms_open ( filename );
/* open file for export */
ctx = IMG$CREATE_DDIF_STREAM ( IMG$K_MODE_EXPORT,
buffer, 512, 0, &user_action, user_RAB );
IMG$EXPORT ( fid, 0, ctx, 0, 0, 0, 0, 0 );
}
/* user-written routine to empty buffer */
int user_action ( bufptr, buflen, usrprm )
char *bufptr;
int buflen, usrprm;
{
printf ( "addr buffer %d length %d param %d\n", bufptr,
buflen, usrprm );
return (1);
}
```

Figure 6. User action routines can decrease memory requirements, as in this example.

```
int ctx;
int fid;
/* open file for export */
ctx = IMG$OPEN_DDIF_FILE ( IMG$K_MODE_EXPORT, filename,
0, 0 );
/* export frame using DDIF stream context */
fid = IMG$EXPORT_DDIF_FRAME ( fid, 0, ctx, 0, 0, 0, 0, 0 );
/* close DDIF stream */
IMG$CLOSE_DDIF_FILE ( ctx, NULL );
```

Figure 7. Images can be output in DDIF format using ISL.

```
Widget image, toplevel;
/* read back attributes from frame */
IMG$GET_FRAME_ATTRIBUTES ( fid, itmlst );
/* init the DECwindows toolkit */
toplevel = XtInitialize ( "test", "test", NULL, 0, &argc, argv );
image = IdsPannedImage ( toplevel, 0, 0, 512, 512, fid, NULL,
view_callback, NULL, NULL );
/* tell toolkit to manage the widget and realize it */
XtManageChild ( image );
XtRealizeWidget ( toplevel );
XtMainLoop ();
```

Figure 8. IDS supplies tools for displaying images including pan and zoom.

VMS tuning parameters

WSDEF - default memory allocation at startup
WSQUO - guaranteed allocation if requested
WSEXTENT - maximum allocation for this process
PAGEFILEQUOTA - limit on virtual address space for process
WSMAX - maximum working set for any process
VIRTUALPAGECNT - system wide virtual address space limit
Set PAGEFILE.SYS to size of virtual address space of all concurrent processes

Figure 9. Systems using VAXimage have some special tuning requirements.

3D graphics: a tutorial

Susan Stearman, Digital Equipment Corporation

(This is a serialization; there are seven chapters overall and so seven parts to the serial. This month, part 7, the end of the line - ed.)

references

The following is a fun and easy to read introduction to graphics in general which also discusses 3D graphics:

> Computer Images (from the Time/Life series "Understanding Computers")

The next two books are the bibles of computer graphics. The first reference has two appendices with details on matrices and matrix arithmetic. These are both recommended for anyone serious about computer graphics. Please note, these are text books:

> Newman, W.M and Sproull, R.F., Principles of Interactive Computer Graphics, McGraw-Hill, New York, 1973.

> Foley, J.D. and VanDam, A., Fundamentals of Interactive Computer Graphics, Addison-Wesley, Reading, MA, 1984.

This reference is a very detailed description of various graphics algorithms. It is a good guide for someone implementing a graphics subsystem:

> Rogers, David F., Procedural Elements for Computer Graphics, McGraw-Hill, New York, 1985.

The following is a sampling of technical publications related to computer graphics:

> Computer Graphics World, PennWell Publications, Littleton, MA. (applications and marketplace related stuff).

> IEEE Computer Graphics and Applications, an IEEE publication (Technical journal with mostly research papers; usually each issue has a theme).

> ACM Topics on Graphics, an ACM publication (Technical journal intended as a research publication vehicle).

seminars in Anahiem

Daniel Land, Seminars Representative

The GAPSIG is proud to sponser five pre-symposia seminars in Anaheim. Two seminars were presented in Atlanta. Both were well attended and enthusiastically recieved.

INTRODUCTION TO DIGITAL IMAGE PROCESSING
Stephen Schultz, Rochester Inst. of Technology

INTRODUCTION TO THE X WINDOW SYSTEM Peter Hack, Digital Equipment Corp.

PORTING UIS APPLICATIONS TO DECWINDOWS Fred Kleinsorge, Digital Equipment Corp.

UNDERSTANDING PHIGS, THE PROGRAMMER'S HIERARCHICAL INTERACTIVE GRAPHICS SYSTEM Jim Flatten, Digital Equipment Corp.

ADVANCED POSTSCRIPT PROGRAMMING TECHNIQUES
Ken Anderson, Adobe Systems

Seminars are driven by your needs. Attendees at the workstation working group meeting in Atlanta mentioned that they needed hard information and examples about porting applications from UIS to DECwindows, this was the type of information which the GAPSIG needs, and the result is a full day seminar devoted to just how to do it with real examples.

Engineering Graphics Users:

The Engineering Graphics Working Group will meet at the GAPSIG Campground (room Pacific A) on Tuesday, November 7th, from 11:00 AM to 12 noon. Please share your ideas and concerns with DECUS members and Digital.

gapsig

call for participation, spring 1990

DECUS meets twice a year to exchange information at national symposia. The sessions for the Fall Symposium in Anaheim are being determined now. Therefore, it's time to start thinking about presenting sessions for the GAPSIG in New Orleans next spring at the DECUS Spring Symposium.

The Graphics SIG deals with both hardware and software graphics topics. These include image processing, window systems, graphics workstations, graphics hardcopy, graphics standards, CAD/CAM, network graphics, visualization, animation, and various graphics languages and systems. The application sessions range from small business and scientific graphics to complex graphics and image processing on VAXstations and large processors.

Tutorial sessions on any of the above topics are encouraged. We solicit presentations which cover present and proposed graphics standards. The Graphics theme for the upcoming symposium is "Network Graphics" which encompasses all aspects of DECWindows as well as other network graphics protocols. Presentations on novel uses of graphics tools and/or uses of graphics equipment in the laboratory, on campus, for data analysis, data presentation or training are welcome. Those sessions may not necessarily have graphics as its main thrust, but use graphics as a tool for getting the job done.

We will conduct a graphics hardcopy contest and encourage attendees to bring renderings of interesting graphics for submission to the contest. We will also produce a tape of graphics related software and encourage attendees to produce submissions for this tape. At the symposium, provisions will be made for any special requirements needed for demonstrations and special display equipment.

DECwindows corner

Commands   Edit   Options

Wecome!
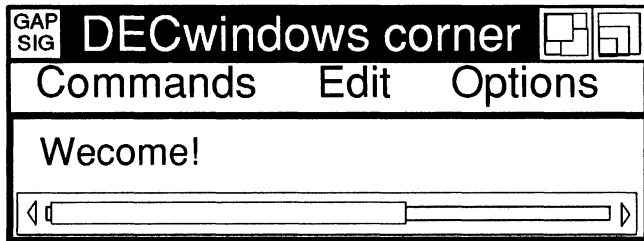
Laura Vanags, DECwindows Working Group Chair

It's almost Fall Symposium time. The excitement is building fast. What new and wonderful things will DEC have for us this time? The DECwindows team will be at Anaheim in full force to answer questions and address user concerns. (Last DECUS, Digital did a great job supporting us. Thanks you!)

This Symposium, the GAPSIG has several excellent Pre-Symposium Seminars which answer user needs - including UIS to DECwindows migration. In addition, we will have a DECwindows stream covering many interesting topics. And last, but not least, look for the **Infamous DECwindows Clinic.** Both VMS and ULTRIX participation is encouraged. Come tell us what you're doing with DECwindows, get DECwindows hints and kinks from others, and listen to the DECwindows team answer your questions. Novices are welcome.

I'd like to provide DEC with a user DECwindows wishlist again. If you have something you want to contribute, send me mail on BITNET (VANAGS@FNALAD), or DECNET (ALMOND::VANAGS), or IPNET (VANAGS@DEVL.FNAL.GOV).

For a preview of the hints and kinks, here's something I found at Fermi. The 8 plane graphics board off-screen memory currently has a total of 1024x2048 bytes. 1024x864 of this is used for the visible screen. 1024x1124 is left. Part of this is used for the text cache, 1024x32 for small pixmaps, part for the visible screen save when the operator window is displayed, and part for scratch use for mutiple large pixmaps (eg. when you copy from one to another). This only leaves 1024x864 bytes for user pixmaps!!! Has anyone come up with a solution to this problem? Has anyone tried using the font area on the graphics board?

Applications

Graphics

SIG
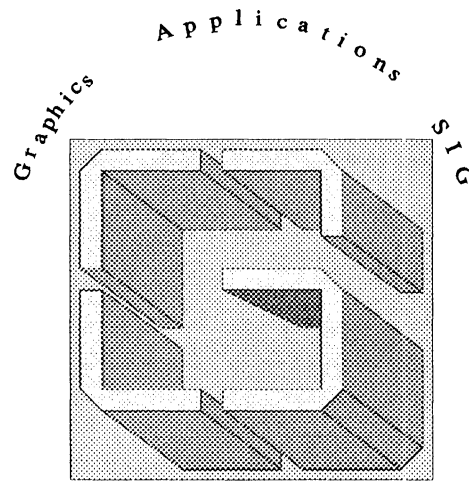
Fall, 1989 in Anahiem

Presents

The Graphics Hardcopy Contest!

The Graphics Applications Special Interest Group (GAPSIG) is once again sponsoring a **Graphics Hardcopy Contest** during the Fall '89 DECUS Symposium in Anahiem. This is your chance to have that stunning graphic recognized by your peers!

The rules are:

1) The Contest is open to all DECUS members.
2) There are two entry categories:
    (a) color, and
    (b) black & white.
3) Prizes of for each category will be awarded.
4) All entries will be displayed in the Graphics Applications SIG Campground at the symposium and are the property of DECUS with the appropriate copyrights. In addition, some entries may be published in the SIGs Newsletter and other DECUS publications.
5) The judging will occur at a scheduled Symposium session by a panel composed of the members of the GAPSIG Steering Committee.
6) The winners will be announced in Friday's Update.Daily at the Symposium, the GAPSIG Wrapup session on Friday and later through this newsletter. You do not need to be present to win.
7) Entries must be an original of size 7" x 10" or larger. A Digital Equipment computer or peripheral must have been an integral part of the production process. Color and halftone prints, plotter/printer outputs, and inkjet/laser prints are all acceptable. Each entry must be accompanied by the full name and address, company affiliation, DECUS membership number and a ten line description of the picture including the hardware and software used for the production.
8) Entries must be deposited at the GAPSIG Campground by Wednesday evening, November 8, 1989, or mailed to:

Bijoy Misra
Harvard-Smithsonian
Center for Astrophysics
60 Garden Street, MS39
Cambridge, MA 02138

Mailed entries must arrive by November 1, 1989 to be entered in the contest.

HARD NEWS

The Newsletter of the DECUS Hardware/Micro SIG
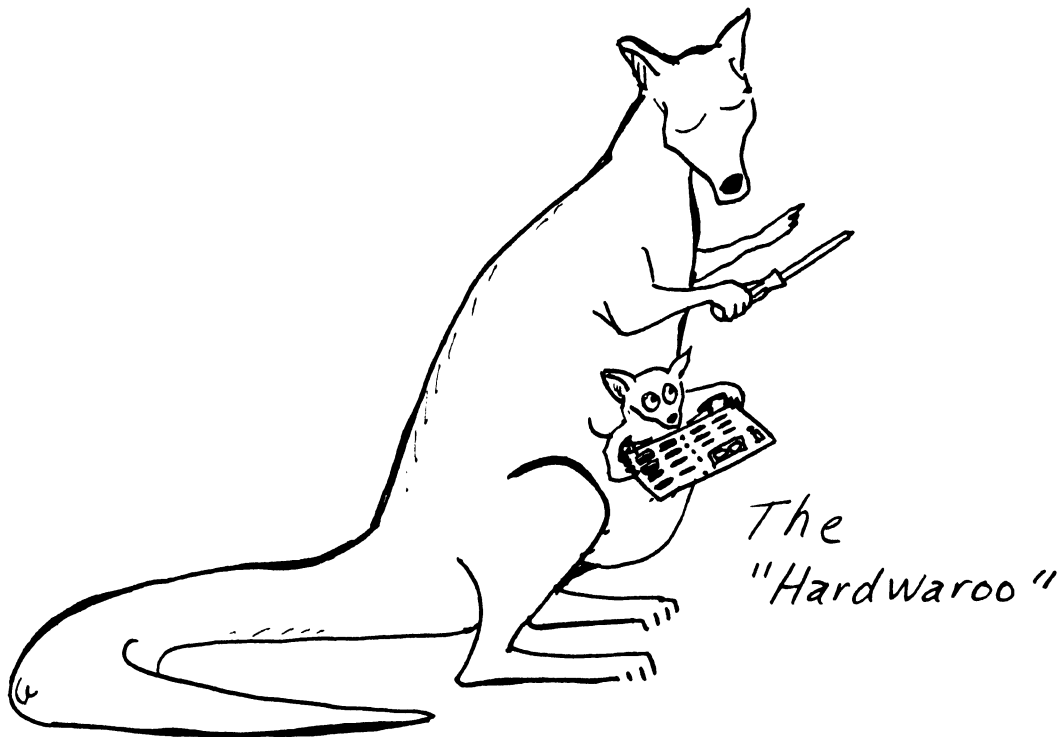
The
"Hardwaroo"

IN THIS ISSUE...

From The Editor...

Here's another reminder that DECUS is a volunteer organization and, therefore, I always need your help with material for the newsletter.
Between the Chair of the HMS SIG, Bill Walker, and myself, we can take submissions in several media including RX01, RX02, and RX50 floppies as well as TK50 tapes. We can also make special arrangements for other media when necessary.

Please remember that your problems and fixes that you've found for your problems are needed and appreciated by other DECUS members. Send your cards, letters, and submissions to:

Neil Krandall
RDB Cincinnati
1440 Elkton Place
Cincinnati, OH 45224

To call (with ideas, brief gripes, suggestions, or corrections), my number is (513) 681-1677. I am also available on Compuserve (via E-MAIL and the PDP-11 SIG) as 71046,1316.

This month we have the reprint of a presentation given by Kyle Hall at the Spring DECUS symposium in Atlanta. This article contains the printed version of the overhead transparencies from which Kyle spoke. Thanks, Kyle, for sharing this valuable overview with the readers of the HMS SIG newsletter.

Digital's CPU Technologies

A Technological Overview of

CISC and RISC

Both Grew out of Performance Issues

CISC:

Size and Speed of Ferrite-Core Memory

High Cost of Microcode RAM

RISC:

Compilers generally did not use complex instructions

Architectural support for complex instructions generally lowered the clock rate

If CISC was a Gladiator

Hardware supports complex HLL tasks

Hardware may provide specialized support for the operating system

Reduces "instructions per task"

Long design cycle

Machine instructions may duplicate HLL functions

A single complex instruction may not execute faster than a series of simple instructions

If RISC was a Gladiator

👍

Easier and faster to design hardware

Simple instruction sets lend themselves to streamlined instruction handling

Easier for compiler to optimize code

👎

More "instructions per task"

More memory used

Optimizing compilers may be slow

"Taste Great"
CISC
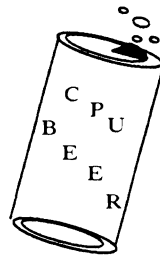
Hardware supports many complex, variable length instructions

Hardware supports many addressing modes

Hardware provides "special" instructions

Eases the design of compilers

"Less Filling"
RISC
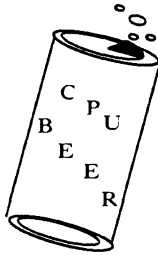
Memory is accessed in a "Load/Store" fashion

CPU contains a relatively large register file

CPU can issue most instructions in one cycle

Few "specialized" instructions

"Still Less Filling"
RISC
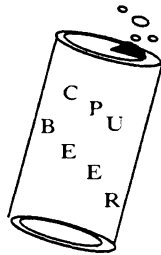
Hardware supports relatively few instructions

Hardware only supports simple, non–variable length instructions

Hardware supports only a few addressing modes

Compilers are forced to be more complex and do more work

RISC tries to increase performance

Performance is measured in "time per task"

"time per task" = C x T x I

C = cycles per instructions (cpi)

T = time per cycle (the inverse of clock speed)

I = instructions per task

RISC tries to reduce C and T, sometimes at
the expense of I

Reducing C (cycles per instruction)

Load/Store architectures

Instruction pipelines

Delayed Load instructions

Delayed Branch instructions

Pipelining

Fetch Instruction	ALU Operation	Write Results
F	A	W

Clock Cycles

	1	2	3	4	...
Instruction #1	F	A	W		
Instruction #2		F	A	W	
Instruction #3			F	A	W
Instruction #4				F	A	W

Pipelining

CPU

Write Results Unit	
	Instruction 1 Instruction 2
Arithmetic Logic Unit	
	Instruction 1 Instruction 2 Instruction 3
Instruction Fetch Unit	
	Instruction 1 Instruction 2 Instruction 3 Instruction 4

Clock Cycle 1   Clock Cycle 2   Clock Cycle 3   Clock Cycle 4
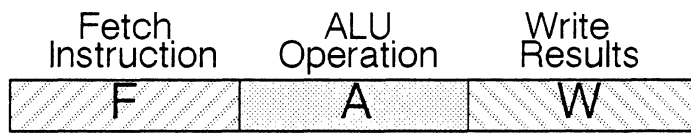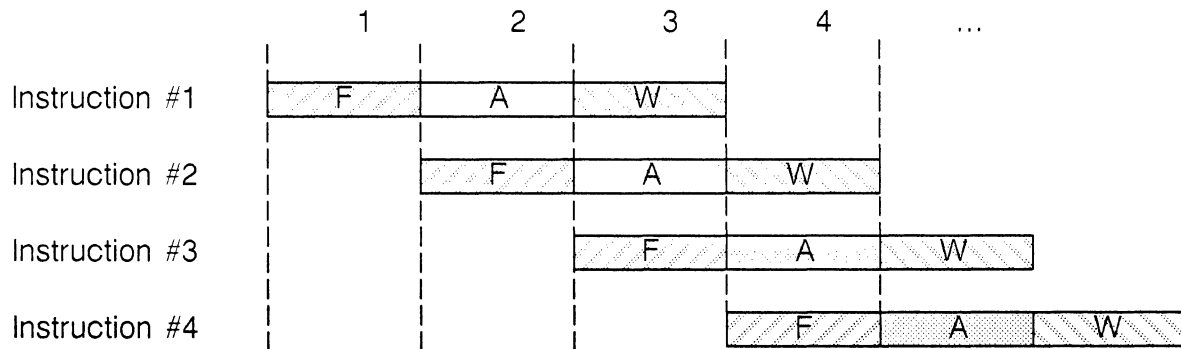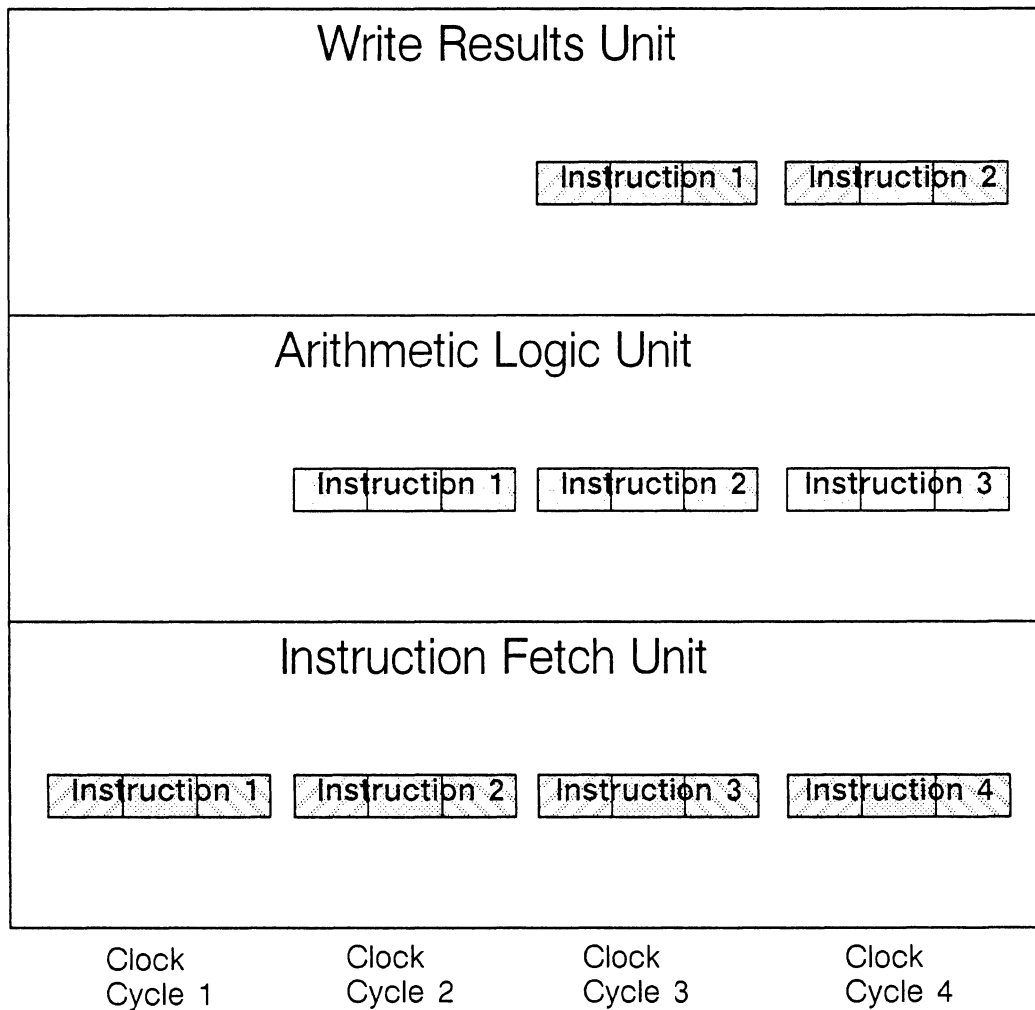
Delayed Load Instructions

Traditional Technique

Latency

Do Something	Load Data	Stall or NOP	Stall or NOP	Use Data	Do Some-thing else

Delayed Load Technique

Other Instruction	Load Data	Do Something	Do Some-thing else	Use Data	Other Instruction

Cycle 1   Cycle 2   Cycle 3   Cycle4   Cycle 5   Cycle 6
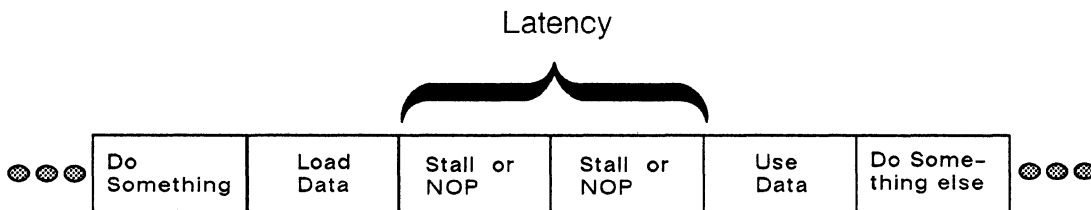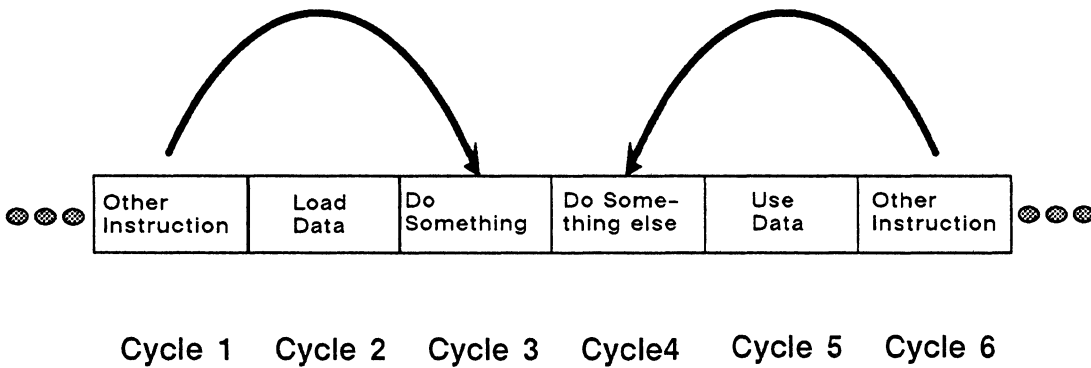
Delayed Branch Instructions

Stall Technique

Latency

S
P
O
T

Do Something	Do Some-thing else	Branch to SPOT	Stall or NOP	Stall or NOP	Complete Branch	Next Instruction

Assumption Technique

S
P
O
T

Do Something	Do Some-thing else	Branch to SPOT	Next Instruction	Next Instruction	Complete Branch	Next Instruction
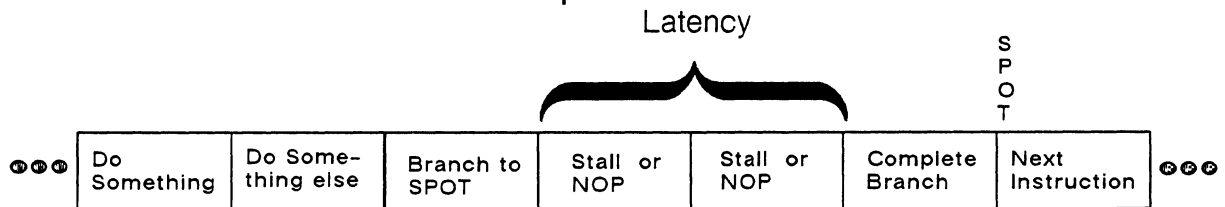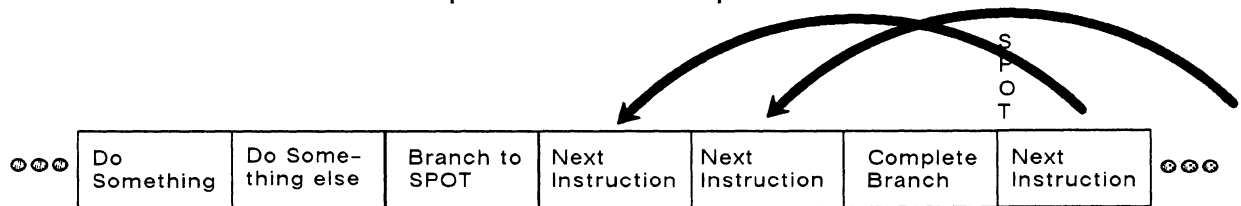
Delayed Branch Technique

S
P
O
T

Previous Instruction	Previous Instruction	Branch to SPOT	Do Something	Do Some-thing else	Complete Branch	Next Instruction

Cycle 1   Cycle 2   Cycle 3   Cycle4   Cycle 5   Cycle 6   Cycle 7
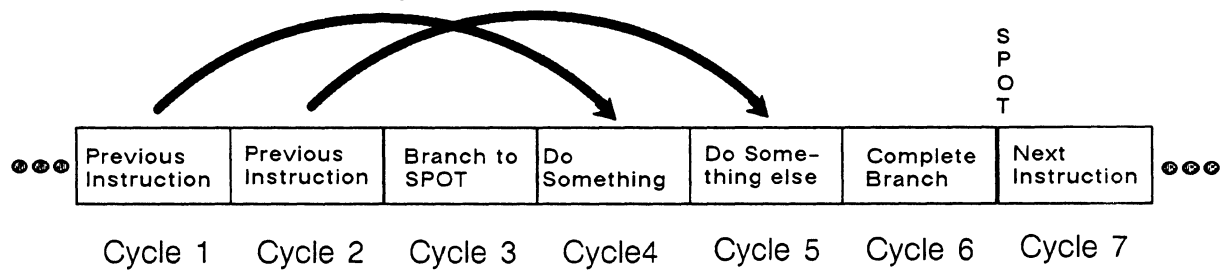
Reducing T (time per cycle)

Architectural simplicity

Instruction decode time

Instruction operation time

Instruction access time

Dependent upon memory bandwidth

Reducing I (instructions per task)

Optimizing compilers
Register allocation

Optimizing operating system
Larger on–chip TLB to support Virtual Memory

Limited support for interrupt or exception vectors

No support for operating system specific instructions

Limited operating modes and protections schemes

Genealogy of RISC

Berkeley Technique

Register Windows

| | Procedure 1 | Procedure 2 | Procedure 3 |

R137

| In |
| Local |
| Out |

Procedure 2

| In |
| Local |
| Out |

Procedure 3

| In |
| Local |
| Out |

R10
R9

Global

R1
R0
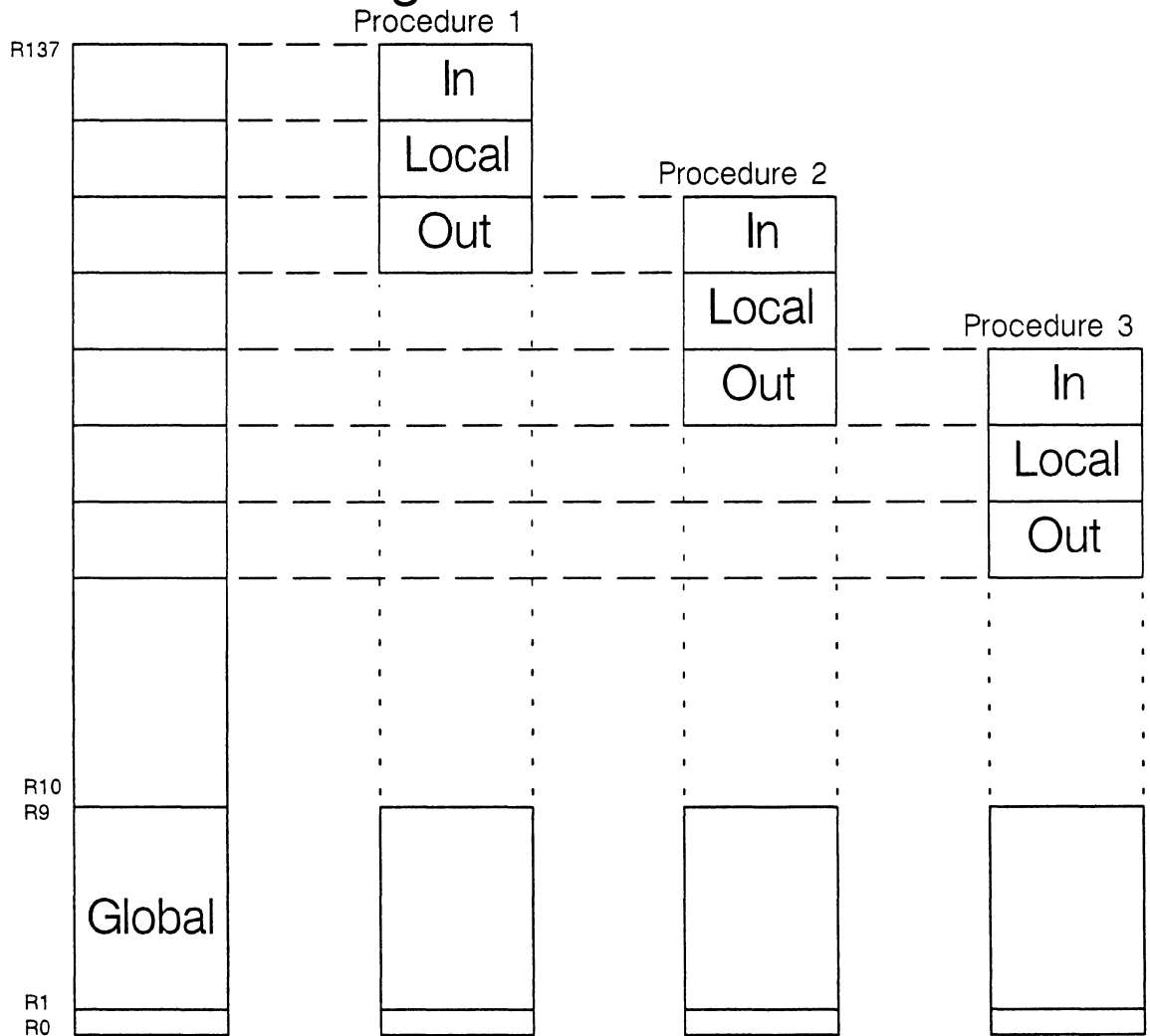
Based on Berkeley RISC I Architecture

Spring DECUS, 1989   Kyle Hall
Atlanta, Georgia

Stanford Technique

MIPS = Microprocess without Interlocked
     Pipeline Stage

MIPS uses a "reorganizer" to avoid interlocking

Register File

R15	
R14	
R13	
R12	
R11	
R10	
R9	
R8	
R7	
R6	
R5	
R4	
R3	
R2	
R1	
R0	

Compiler optimizes register usage

Comparison of four commercial processors

	CVAX VAXstation 3100	R2000 DECstation 3100	SPARC SUN 4	HP PRECISION
Instructions	304 (181)	102 (64)	117	140
Instruction Formats	2 or lots	3	3	1
Data Formats	19	9	12	13
Addressing Modes	23	1	2	3
Pipeline Depth	3	5	4 or 5	3
Registers	16	32	128 – 140	32
Register Windows	No	No	Yes	No
Claimed Performance	3–4 vups or 3.3–5 mips	10 vups or 11–12 mips	7.6 mips	10–15 mips
On–chip VA Support	28 E. TB	64 E. TLB	No	8 Space Reg.
Cache	1Kb on–chip 32Kb I/D	64 Kb I & 64 Kb D	?	64 Kb I & 64 Kb D
Floating Point CP	Yes	Yes	Yes	Yes
Single Double Quad	Yes Yes Yes	Yes Yes No	Yes Yes Yes	Yes Yes Yes
Operating Systems	VMS or ULTRIX	ULTRIX	SUN OS	MPE XL or HP–UX

Spring DECUS, 1989
Atlanta, Georgia

Kyle Hall

DECstation 3100 Block Diagram

Kyle Hall

Leverage

CMS LSE TPU ADA PCA VAX 8600

Languages & Tools

TABLE OF CONTENTS

EDITOR'S NOTES

It's always a pleasure to put out an issue like this one, with several great articles. For one thing, I don't have to write much to fill out to the bottom of this page!

For those of you who attend symposia, and wonder why you are constantly being asked to fill out evaluation cards at L&T sessions, George Scott has provided an analysis of the most highly rated sessions from the last symposium. With the growth of L&T, and the number of sessions we present, the evaluation cards provide a valuable tool in deciding which sessions to repeat. The next time a session chair repeats the litany about the evaluation cards, remember, we DO use them!

Also in this issue we have the next in the series by John Roman, on software tools in VAX Macro, and a very readable discussion of the Fortran 8X standard by the DECUS representative on the standard committee, Rochelle Lauer.

The issue rounds out with two reports from our working groups. Pete Siemsen describes a new TECO tape which has been submitted to the DECUS library, for you TECO fans; and Shirley Bockstahle-Brandt has submitted an excellent article on sources for ADA components.

That's it for this month. If you would like to participate in the *Leverage* Merry-Go-Round, or if you have any questions, suggestions, or animadversions, please contact me. See you next month.

Al Folsom, *Leverage* Ed.

L&T SESSIONS AT DECUS ATLANTA

by George L. Scott

The L&T session evaluation cards from Atlanta have spoken, and here is what they said. One thousand nine hundred ten cards were received for all of the L&T sessions. There were 421 votes for Excellent, 714 votes for Very Good, 534 votes for Good, 194 votes for Fair, and 47 votes for Poor. The overall rating for all L&T sessions, based on votes received for all sessions combined, is 2.66, or one third down from Very Good. Table 1 shows the same information for the L&T sessions with at least ten votes and an average score of at least Very Good (3.00 on a scale from 4.00 for Excellent to 0.00 for Poor). Table 2 shows the vote distribution and summaries for the four L&T sessions that received at least fifty votes. Table 3 is a key for the other two tables. Some of the comments for the top four vote getters and top ten averages are in the following paragraphs.

The best received session at Atlanta was LT049, Introduction to C Pointers, presented by Chet Small, with an average rating of 3.77 and a total of 39 votes. Typical comments are: "Best presentation. Need more like these for learners." "Speaker very good presenter and entertaining! Very knowledgeable and clear style." "Very good presentation of slides. If more people would have slides like these then the session notes would be more useful for sessions we are unable to attend." "Super presentation!" "Life would have been much better if Mr. Small was my university prof when I was learning C." "Very good real world examples."

The second best received session was LT099, Standards Futures, by James Ebright, with an average score of 3.70 and a total of 10 votes. Comments noted include: "Clear presentation, Well organized. DECUS should have more information on POSIX and POSIX standards." "Very comprehensive, factual non-biased & objective." "Very informative."

In third place is LT093, More Pointers on C Pointers, by Chet Small, scoring 3.57 with 61 cards received. Commentators said: "One of very few talks that was directed at the perfect level between a 'sell of product' and a show of speaker intelligence." "Continue using 'flowchart' for making and analyzing bizarre pointers." "Best tutorial I had in C pointers." "Held my attention better than any other session. Also did a better job of conveying information." "Wonderful speaker." "Just right material." "Thanks for the good overheads." "Good presentation style. Some examples went by pretty fast but notes help!" "Working code examples are very helpful."

LT157, Advanced CMS Tutorial, by Edgar Whipple, came in fourth. It received a 3.46 with 24 cards. Comments include: "Very entertaining, humorous, and interesting." "Edgar Whipple is a FABULOUS speaker. Keeps it light & entertaining but does a great job of getting technical points across in an understandable way. Product news was very well done." "Lots of good stuff. Need more sessions on CMS tricks, like this." "Very technical. It's good to see serious experts!" "Thanks for getting slides into session notes. Makes it much easier to listen."

In fifth place is LT158, MMS Q&A, by Edgar Whipple, with a 3.44 on 10 votes. Among the comments are: "Should be repeated - can't get this anywhere else." "Put the slides and comments in the newsletter." "The presentation helped answer many questions, but was a little long."

Beth Benoit submitted the sixth place LT141, Parallelism in VAX C, which received 3.38 on 13

votes. Attenders commented: "Good presentation. Please publish answers to questions that you 'have to check on' in L&T newsletter." "Good speaker. Excellent slides." "Best speaker so far, and today is Thursday."

Seventh is LT140, What's New in VAX C, by Beth Benoit, receiving 3.36 on 25 votes. People said: "New stuff is great. Lots of new things to play with. Worried about debugging decomposed code." "Excellent presentation on new features." "Lots of relevant information presented well." "This Fortran user found it very interesting."

Linda Ziman gave eighth place LT161, Software Development Metrics, receiving 3.33 with 15 votes. Measured comments are: "Good content." "Picked up some good ideas, methods for collecting metrics." "You gave me some very good ideas." "Explanation of types of charts & metrics - conclusions drawn or action taken very helpful.

There was a tie for ninth place. One is LT025, Languages and Tools Question and Answer, with a rating of 3.29 and 17 votes. People thought about it this way: "Thanks for the opportunity." "Good dialog." "Wonderful to have this access." "Is good idea...."

The other ninth highest scorer is LT164, On-Line Documentation, by Mary Utt and scoring 3.29 with 17 votes. Comments include: "Would like to see in-house tools become products." "Outstanding - please repeat @ future symposia!" "Well presented. Well organized. Interesting. Useful." "Get this product to market soon. It's too good for DEC to keep in-house." "Good session - thanks for the handouts."

The session receiving by far the most votes was LT143, Debug Tutorial, by Kevin Routley. Ninety-three people dropped comment cards in the boxes at this session, with an average vote of 2.40. Of course, there were many written comments, including: Immediately useful." "Great presentation, full of information, well organized." "Repeat!!" "Great how-to session. Immediately useful." "Good visual aids. It's also nice to have a handout." "Could be expanded. Like to see more examples." "Great!" "Necessary presentation at each symposium. Handouts appreciated, though poorly reproduced. Well ordered presentation; made it good." "Just what I needed." "I got a lot out of it. Handouts were good!" "Good overview. Interesting DECwindows preview." "The handout is especially helpful." "Learned some new items even though I have been using Debug for several years."

The second biggest vote getter, LT093, is also one of the highest rated. It is discussed above.

The third biggest vote getter is LT022, the Languages and Tools Roadmap, with 60 votes and a score of 2.52. The session comment cards were introduced in this session. Comment on the Roadmap include: "Had a nice, informal flair." "Useful info." "Nice intro to the week." "Very informative for a first time DECUS attendant." "Nice to have the chance to meet working group chairs." "Informative."

In fourth place on votes is LT101, The Art of Debugging, by Richard Gilbert, with 53 votes and a score of 1.38. Significant comments include: "Real world examples of tricky problems." "Enjoyed the presentation." "Richard Gilbert is a good speaker. Clear, simplified presentation." "Too specific into Fortran problems - should generalize." "Could have been more specific with the examples."

Table 1. Sessions Averaging Very Good or Better

NR	SESSION NAME	EXCL	VYGD	GOOD	FAIR	POOR	VOTE	RAW	NORM
49	An Introduction to Pointers in C	31	7	1			39	147	3.77
99	Standards Futures	7	3				10	37	3.70
93	More Pointers on C Pointers	38	20	3			61	218	3.57
157	Advanced CMS Tutorial	13	9	2			24	83	3.46
158	MMS Q&A	6	2	2			10	34	3.40
141	Parallel Decomposition in VAX C	7	5		1		13	44	3.38
140	What's New in VAX C	13	8	4			25	84	3.36
161	Software Development Metrics	8	4	3			15	50	3.33
25	Languages and Tools Q&A	8	6	3			17	56	3.29
164	Producing Online Doc for Bookrdr	6	10	1			17	56	3.29
63	MMS Tutorial - Beginners & Advan	10	11	4			25	81	3.24
162	Managing Tech Change in SW Org	21	16	10			47	152	3.23
29	VAX Basic Toolkit	6	5	3			14	45	3.21
131	Managing Software Projects	11	18	4			33	106	3.21
128	EVE for EDT & WPS Users	6	13	3			22	69	3.14
24	L&T Magic and Wizardry	5	10	3			18	56	3.11
134	What's New with VAX Ada	3	5	2			10	31	3.10
149	VAX Scan Tutorial	8	8	3		1	20	62	3.10
68	C RSTS Basic Migration to VMS	3	7		1		11	34	3.09
167	TPU Programming	4	6	3			13	40	3.08
132	SW Development Environ Crystal B	10	17	6	1		34	104	3.06
70	Using VMS Routines from Cobol	6	15	5			26	79	3.04
32	DEC/Test Manager Tutorial	6	11	6			23	69	3.00
62	A Remote CMS Tool	4	5	2	1		12	36	3.00
151	Recent Features in VAX Cobol	2	6	2			10	30	3.00

Table 2. Sessions with over 50 Cards Received

NR	SESSION NAME	EXCL	VYGD	GOOD	FAIR	POOR	VOTE	RAW	NORM
143	VAX Debug Tutorial	14	30	33	11	5	93	223	2.40
93	More Pointers on C Pointers	38	20	3			61	218	3.57
22	Languages and Tools Roadmap	7	19	32	2		60	151	2.52
101	The Art of Debugging	2	7	13	18	13	53	73	1.38

Table 3. Key for Tables 1 and 2

NR	: session number.
SESSION NAME	: session title.
EXCL	: count of session survey card votes for Excellent.
VYGD	: count of session survey card votes for Very Good.
GOOD	: count of session survey card votes for Good.
FAIR	: count of session survey card votes for Fair.
POOR	: count of session survey card votes for Poor.
VOTE	: count of session survey card votes of any kind.
RAW	: (4 * EXCL + 3 * VYGD + 2 * GOOD + 1 * FAIR + 0 * POOR), giving the total rating votes received for a session.
NORM	: RAW / VOTE, giving the average rating for a session.

SOFTWARE TOOLS IN VAX MACRO

John Roman
M/S GG3I
Monsanto Company
Chesterfield, MO 63198

VI. FIND TOOL: STRUCTURED MACROS

Introduction

Ever since my Macro-11 days with IAS, I have been intrigued by Structured Macros. These macros implement the common constructs of structured programming, such as if-then-else, do-while, and the like on top of the assembler code. Now that I am programming VAX Macro (when I get the chance), I also like to use Structured Macros when appropriate. There are several advantages to using them:

1.   Quality -- the use of the macros minimizes errors by properly handling branch structures.

2.   Productivity -- it is possible to code faster using Structured Macros as their syntax is closer to pseudo-code and higher level languages.}

3.   Training -- because their syntax is more like higher level languages, novice Macro programmers can use them as they learn and become proficient with VAX Macro.

4.   Style -- the use of macros can help enforce the common style and make programs better by being more readable.

Of course, there are disadvantages to using structured macros. For one thing, they place a layer between the coder and the actual instructions. If one is concerned about speed or size of code, the macros may get in the way and not produce optimum code. Second, in the past, debugging code with macros has been difficult. However, now with VMS Version 5, the debugger default is to display the source code and step by line. If desired, it is also possible to step by instruction. As a result debugging code with macros is now much easier. Third, Structured Macros are not standardized. Each site may use a different set of macros with a different syntax. Therefore exchanging code requires exchanging the macro library at the same time.

For these reasons, I don't recommend using Structured Macros to every programmer or every shop. Some may decide not to use them. However, in many cases they help in learning and programming in VAX Macro.

In this article of the series on Software Tools in VAX Macro, let us consider the use of Structured Macros. They are particularly useful with code in the Find tool that we are in the

progress of building, because of the emphasis on control structures in that tool.

The Structured Macro Package

I am using the Structured Macro package called SMAC from Batelle Columbus Labs. It was written by Gary Grebus while he was there and it is available on one of the DECUS tapes. There are other packages available; one is the MLR package from Roderick A. Eldridge of Iowa State University.

The SMAC library has macros for several of the common constructs of structured programming. The following are implemented:

```
IF condition THEN
    . . .
ELSE
    . . .
ENDIF

REPEAT
    . . .
UNTIL       | condition     |
            | FOREVER       |
            | ONCE          |

WHILE condition DO
    . . .
ENDWHILE
```

The REPEAT and WHILE structures can include the BREAK statement, which conditionally exits from the structure, and the NEXT statement, which conditionally branches to the bottom of the structure. The form of these two statements is:

```
BREAK  [label]  IF condition

NEXT  [label] IF condition
```

The optional label must be defined at the appropriate place in the structure. If the label is omitted, these statements apply to the innermost structure which contains them.

A *condition* has the form of:

```
| test
| test {AND} test {AND} test ....    |
|      {OR }      {OR }               |
```

Up to six tests can be included in one condition. The tests are performed in left to right order, with no precedence.

A *test* has the form:

```
<relation,[arg1],[arg2],[TYPE=x]>
```

Relation is the test to be performed and is specified as the appropriate suffix to the branch opcode; i.e. BBC is BC, BEQL is EQL, BLBS is LBS, etc. ARG1 and ARG2 are the operands for the test. If ARG2 is omitted, a TST instruction is generated. If both arguments are omitted, the condition codes are assumed to be set and only the appropriate branch instruction is generated. If both arguments are specified, a CMP instruction is generated. TYPE controls the data type of the comparison, with B for byte, W for word, etc. The default is L for longword.

Here the if-then-else construct is used to get the maximum of two numbers:

```
IF <GTR, R1, R2> THEN
        MOVL       R1, MAX
ELSE
        MOVL       R2, MAX
ENDIF
```

This is an example of stringing tests together:

```
IF <NEQ, R1, R2> AND <NEQ, R6> THEN
        SUBL3      R1, R2, R0
        DIVL2      R0, R6
ELSE
        CLRL       R6
ENDIF
```

As an example of the do-while construct, this code will search for the end of a null terminated string or stop at the end of the buffer (at the address BUFF_END).

```
MOVAB       SOURCE_BUFF, R0
MOVAB       DEST_BUFF, R4
WHILE <NEQ, (R0), TYPE=B> AND <LSS, R0, #BUFF_END> DO
        MOVB       (R0)+, (R4)+
ENDWHILE
```

This loop clears a buffer given an address and a count of longwords in the buffer:

```
MOVL     COUNT, R0
MOVAL    BUFFER, R1
REPEAT
    CLRL       (R1)+
    SUBL2    #1, R0
    BREAK IF <LSS, R0>
UNTIL    FOREVER
```

Although not necessarily the most efficient implementation (the MOVC5 is better for most values of COUNT), it does indicate how the structure can be used.

There are several other miscellaneous macros in the SMAC package. One of the most useful is the CALL macro. Its format is:

```
CALL    routine    arg1,arg2,arg3,...
```

This routine will stack the arguments in reverse order and perform a CALLS to the address ROUTINE. If the argument is an address, a PUSHAL is generated, and if the argument is a literal, a PUSHL is generated.

Structured Macro Examples

Now let us see how we might use these macros as we build our Find tool. First, consider the routine getccl, which is used to build the encoded pattern when a character class is used in the search string. For example, when the search pattern is [0-9] indicating that any of the numbers 0 through 9 are valid, the encoded pattern is CCL-(number 10)-0123456789. CCL is a flag byte indicating that a character class follows. The next byte is the size of the class, followed by bytes indicating valid matches. First, here is the RATFOR code:

```
# getccl -- expand char class at arg(i) into pat(j)
    integer function getccl(arg, i, pat, j)
    character arg(MAXARG), pat(MAXPAT)
    integer addset
    integer i, j, jstart, junk

    i = i + 1                # skip over [
    if (arg(i) == NOT) {
        junk = addset(NCCL, pat, j, MAXPAT)
        i = i + 1
        }
    else
        junk = addset(CCL, pat, j, MAXPAT)
    jstart = j
    junk = addset(0, pat, j MAXPAT)        # leave room for count
    call filset( CCLEND, arg,i, pat, j, MAXPAT)
    pat(jstart) = j - jstart - 1
```

```
if( arg(i) == CCLEND)
    getccl = OK
else
    getccl = ERR
return
end
```

Now here is the VAX Macro code using the IF-THEN-ELSE and CALL structured macros:

```
ARG = 4                    ; Address of pattern argument
I = 8                      ; Address of i
PAT = 12                   ; Address of encoded pattern
J = 16                     ; Address of j

.ENTRY GETCCL,^M<R2,R3,R5,R6,R7,R8,R9>
MOVL      J(AP), R5
MOVL      ARG(AP), R3
MOVL      PAT(AP), R6
MOVL      I(AP), R2
INCL      (R2)             ; Skip over [
MOVL      (R2), R7
IF < EQL, (R3)[R7], #NOT, TYPE=B> THEN
          CALL ADDSET   #NCCL, R6, R5, #MAXPAT
          INCL (R2)
ELSE
          CALL ADDSET #CCL, R6, R5, #MAXPAT
ENDIF
MOVL      (R5), R8
CALL      ADDSET #0, R6, R5, #MAXPAT   ; leave room for count
CALL      FILSET #CCLEND, R3, R2, R6, R5, #MAXPAT
SUBL3     R8, (R5), R9
DECL      R9
CVTLB     R9, (R6)[R8]
MOVL      (R2), R7
IF  < EQL, (R3)[R7], #CCLEND, TYPE=B> THEN
          MOVL #OK, R0
ELSE
          MOVL #ERR, R0
ENDIF
RET
```

The next example is locate, which uses the encoded character class pattern when checking the line from the file for a valid match. It takes the pattern and the offset, which is the length inserted by getccl above, and the character to look for. It then moves to the end of the character class and checks character by character as it moves back toward the start of the class. If it finds the character, it returns YES, otherwise it returns NO. Here's the RATFOR code:

```
# locate -- look for c in char class at pat(offset)
    integer function locate(c, pat, offset)
    character c, pat(MAXPAT)
    integer i, offset
    # size of class is at pat(offset), characters follow

    for(i = offset + pat(offset); i > offset; i = i - 1)
        if( c == pat(i)) {
            locate = YES
            return
            }
    locate = NO
    return
    end
```

Here is the VAX macro code. Notice that the for loop in RATFOR is turned into a while loop with the initialization part of the for loop done prior to entering the loop.

```
    C = 4                        ; Value of character
    PAT = 8                      ; Address of pattern
    OFFSET = 12                  ; Value of offset

    .ENTRY  LOCATE,^M<R2,R3>
    MOVAL   PAT(AP), R3
    MOVL    OFFSET(AP), R2
    CVTBL   @(R3)[R2], R0        ; calculate pat(offset)
    ADDL3   R2, R0, R1           ; and offset + pat(offset)
    MOVL    (R3), R0             ; Get address of pattern
    WHILE <GTR, R1, OFFSET(AP)> DO
        IF <EQL, C(AP), (R0)[R1], TYPE=B> THEN
            MOVL    #YES, R0
            RET
        ENDIF
    DECL    R1                   ; Look at previous character
    ENDWHILE
    MOVL    #NO, R0
    RET
```

The final example is index, which is used by the part of the code which builds the encoded pattern. It searches through a string until it finds a character or the end-of-string marker. It also uses the for construct. Here is the RATFOR code:

```
# index -- find character c in string str
    integer function index(str, c)
    character c, str(ARB)
```

```
for( index = 1; str(index) != EOS; index = index + 1)
    if (str(index) == c)
        return
index = 0
return
end
```

Here is the VAX Macro code. Again, the for loop is translated into a do-while loop with the initialization prior to the loop. Notice that the index, R0, is cleared because we are using zero-indexed arrays with Macro. Therefore, instead of returning 0 to indicate failure, we have to return -1 because we could find the character at the 0th position.

```
STR = 4                    ; Address of string
C = 8                      ; Character to locate

.ENTRY  INDEX,^M<R3,R4>
MOVL        STR(AP), R3
MOVZBL  C(AP), R4
CLRL        R0
WHILE < NEQ, (R3)[R0], #EOS, TYPE=B> DO
    IF <EQL (R3)[R0], C(AP), TYPE=B> THEN
        RET
    ENDIF
    INCL    R0
ENDWHILE
MOVL        #-1, R0
RET
```

How It's Done

Before we leave Structured Macros, it might be instructive to take a look at the code that the macros generate. Let us take the index routine above, and assemble it with the following command:

MACRO INDEX.MAR+FIND.MLB/LIB+SMAC.MLB/LIB/NOOB/SHOW=(BINARY)/LIS=INDEX.LIS

and then look at the generated code:

```
                00000004  0000    43      STR = 4
                00000008  0000    44      C = 8
                          0000    45
                    0018  0000    46      .ENTRY  INDEX,^M<R3,R4>
53    04 AC   D0  0002    47      MOVL        STR(AP), R3
54    08 AC   9A  0006    48      MOVZBL  C(AP), R4
```

```
          50   D4  000A   49        CLRL    R0
                  000C    50        WHILE < NEQ, (R3)[R0], #0, TYPE=B> DO
                  000C              _.0.1:
     00   6340   91  000C           CMPB    (R3)[R0],#0
          0C     13  0010           BEQL    _.0.3
                  0012    51        IF <EQL (R3)[R0], R4, TYPE=B> THEN
     54   6340   91  0012           CMPB    (R3)[R0],R4
          01     12  0016           BNEQ    _.1.3
          04     04  0018    52       RET
                  0019    53       ENDIF
                  0019              _.1.3:
          50     D6  0019    54       INCL    R0
                  001B    55       ENDWHILE
                  001B              _.0.2:
          FFEE   31  001B           BRW     _.0.1
                  001E              _.0.3:
  50 FFFFFFFF 8F D0  001E    56     MOVL    #-1, R0
          04     04  0025    57     RET
                  0026    58       .END
```

For those of you not familiar with Macro listings, the column running from 0000 to 0026 is the location counter. The column to the right of it is the source line numbers and the columns to the left are the actual assembler code. The lines of source code without assembler code to the left are macros which get expanded to the code which follow (and have assembler code to the left). Therefore the while-do macro generates three lines of code in this case, the label, the CMPB, and the BEQL. Likewise, you can see how the if-then, endif, and endwhile are expanded to generate the appropriate code.

Notice that the while loop is converted into an if statement checking at the top of the loop with the opposite condition (BEQL rather than BNEQ). In the same manner, the if-equal-then construct is converted to a CMPB and BNEQ combination. This should satisfy you that the code generated is just as you would expect it to be.

Conclusions

I hope I have indicated to you the utility of Structured Macros as a tool to create code which is easier to understand and support. I hope you will agree that the Macro code above is relatively easy to understand. In these simple examples it works quite well. I will admit that Structured Macros are not for everyone and all situations, but in many cases it can help improve the productivity and quality of Macro code.

Again, it is a pleasure to acknowledge the assistance of Marc Blaskie in preparing this article.

FORTRAN 8X – IT'S YOUR TURN
Part 1

Rochelle Lauer
Decus Fortran Standards Representative

1. INTRODUCTION

The revised Fortran standard is in the process of its second public review. The public review period is July 28,1989 - Nov 24,1989, therefore, now is the time to make your input heard. In order to help you along, this article (and the one to follow in November), will present information about the standard, which hopefully will clarify the major features and controversies which have been in the media since the first public review over a year ago. I urge you to send for the standard (address at the end of the article), and respond to the public review. Remember its your turn!

These articles will provide an overview of the language features, but will not be a full technical description(please, send for the standard!). In addition, as its my turn as well, I will present my opinions on the standard, in order to help solidify your own.

This month, I present the features of the language and explain my views as a Fortran programmer. Next month, I will try to answer some of the issues raised by DEC in the August newsletter, presenting my views as a DEC customer.

2. SOME QUESTIONS; SOME ANSWERS

Why a Standard?

Standards have always been important in the computer field. Electronic standards (e.g. RS232, Ethernet) let us connect many types of devices with standard protocols. Standards in programming languages allow us to execute our software on many hardware platforms.

It is clear (as evidenced by the use of VAX extensions) that FORTRAN 77 does not have the functionality required for modern programming. Many of us have sacrificed portability by using VAX extensions because our current system configurations are exclusively VAX/VMS. We are learning however, that future distributed processing will rely on portability. Server/client applications (such as those developed with X windows) will require that software run on many different types of hardware. We will want to chose the hardware which is appropriate for the task, not because it is the only hardware on which the software runs ! For scientific programmers, a Fortran that standardizes modern features is essential.

Why ONE Standard?

There has been some talk in the media about having two standards. I find this talk misleading. There really can be only one standard anything. Two standards really means two languages. A Fortran program will NOT be portable, because there will be NO Fortran; there will be Fortran I

and Fortran II; two different languages. Will ADA programs compile with a Fortran compiler? Will Fortran I programs compile with a Fortran II compiler? Do I want to BUY two Fortran compilers. To me all the answers are no. I want one Fortran standard, so that my Fortran code will compile on any compiler which implements the standard.

Should I Support It Even If I Oppose (or hate?) some of the Features?

Yes! Remember that until a language is used, the desirability (or UNdesirability) of features may not be known. Obviously we know that certain features of Fortran 8X are essential (e.g. long variable names, INCLUDE, extended source form, structures); we have been using them in VAX Fortran for years. Standardizing the use of these features will provide for more portable code; reason enough to support the standard. With a standard in place the desirability of features will be put to the test through use. There is no better way.

Responding to the public review is one way to influence future implementations. A positive response will get the standard going, resulting in the only true test of the features. But, nothing is perfect and Fortran 8X is far from it. Be sure to express your objections, both to influence the current draft, and to get a start on the future.

Does My Voice Count?

You Bet! Many significant changes were made to the standard due to the first public review. In particular, DEC VAX extensions (INCLUDE and DO WHILE) were added. It is therefore both essential and purposeful to respond.

3. FEATURES OF FORTRAN 8X EXPLAINED

3.1 Specifics

This section presents some of the major features of Fortran 8X, stressing how they can be used. The features mentioned here are certainly not all the new features or additions to the language. They are the most interesting ones (to me!) and/or have generated much comment during the first public review.

MODULES

Modules are a new concept in Fortran 8X. They provide information hiding and scope by restricting knowledge of the definitions in the module to the subprograms that USE the module. For many Fortran programmers modules will be the alternative for COMMON.

Definitions in MODULES can replace COMMON if COMMON (as in many cases) is used only to share variables among subroutines. In most cases, the STORAGE ASSOCIATION(offset from the start of the common block) of COMMON is not necessary, and is often detrimental to debugging and maintaining the code. Definitions in modules provide the NAME ASSOCIATION (knowledge of the variable's name), which is the real requirement of many applications now using COMMON.

Some important uses of MODULES include:

- Variables can be known to a defined group of subroutines and be restricted from others. In FORTRAN 77 variables are either local or in COMMON (global in the sense that the name of the common block is global and the variable is an offset from the common block). Restricting variable definitions (scoping)prevents conflicts in naming which might result in execution errors (unintentional sharing of a variable), thereby increasing the reliability of the final product.

- Within a module a definition can be declared private to the MODULE in order to avoid conflicts with subprograms that may USE the module. Fortran programmers and vendors alike will now be able to write packages and subroutine libraries without concern for naming conflicts in the programs that use those packages.

ATTRIBUTE SPECIFICATION STATEMENTS

These statements provide a consistent syntax for defining attributes of the objects you define, thereby making code easier to debug and maintain, resulting in software which is more reliable.

```
Examples:
    real, dimension(5,5) :: a, b
    real, parameter  :: pi=3.14
```

These examples show how to implement existing semantics with new syntax, however new Fortran 8X semantics(e.g. pointers and allocatable arrays), are implemented via attribute specification syntax, and using the new syntax improves the consistency and readability of your code.

```
Example:
    real, allocatable, dimension(:) :: my_array ! a one dimensional
                                               ! allocatable array
                                               ! (deferred shape)
```

PARAMETERIZED TYPES

Fortran 8X has made an attempt to provide consistent syntax for data types which may have more than one representation(e.g. reals can be single precision, double precision, g floating etc.). In addition, for consistency, the specification of character length has been implemented with similar syntax.

```
Examples
    real (kind =1) :: x   ! x is of type kind =1
                          ! intrinsic functions will tell you
                          ! what the characteristics of kind =1
                          ! are
    character(len=10) my_name
```

Note that although code to be ported cannot be guaranteed of the same precision on different architectures, the intrinsics will let you decide (at run time) what KIND (precision) this processor requires. By specifying KIND as a parameter, code can be more easily transported.

Example:

```
integer,parameter :: short = selected_int_kind(4)
                        ! selected_int_kind is an
                        ! intrinsic function
                        ! returns the value of kind
                        ! necessary to represent an integer i
                        ! -10**4<i<10**4

integer (kind=short) :: my_int   ! range at least
                                 ! -9999 to 9999
```

POINTERS

Many public review responses requested pointers. A pointer attribute was added to Fortran 8X as part of the attribute specification statement. Variables defined with the pointer attribute can reference any type of data, and have particular use when combined with allocatable arrays.

Example:
```
real, pointer,dimension(:) ::  p      ! array pointer
real, allocatable, dimension(:) :: e ! p could point to e
```

DEFINED AND OVERLOADED OPERATORS

A programmer can define an operator or define how to use an intrinsic operator (+,-,// etc) on derived data types. This features opens the possibility of providing generic subroutines for data types not provided in Fortran, such as varying length character strings, or data types which are application specific.

ARRAY LANGUAGE

The array language allows us to manipulate whole arrays and sections thereof as data objects. In addition, allocatable arrays provide for dynamic storage, long lacking in Fortran.

The array statements cover a wide range of functionality. Here are few a (hopefully self explanatory) examples.

```
real , dimension (:)   :: x
real, pointer, dimension(:) :: mid
```

```
      n=10
      allocate (x(n),stat=ierr_alloc)  ! check  stat  to  see  if  it
                                        !  was  allocated  correctly
      mid => x(5)                       ! mid points  to  the  middle
                                        ! of  x
```

INTERFACE BLOCKS

Interface blocks are used to explicitly define properties of a subprogram. Such properties include:
    Keywords
    Optional parameters
    Generic calling sequence
    Definition of overloaded operators

Examples:

```
   interface
       subroutine keyword_example (keyword_arg)  ! only defines  the
          real :: keyword_arg                     ! interface
          end subroutine keyword_example          ! actual code defined
                                                   ! elsewhere
   end interface
```

Keyword exampled can be called with:
    call keyword_example(keyword_arg=my_arg)

```
   interface assignment (=)                    ! defines  procedures  to  do
       module procedure assignss,asignsc,assigncs  ! assignment
   end interface
```

3.2 OTHER AVAILABLE FUNCTIONALITY

Fortran 8X has other desirable functionality. The following features are mentioned only briefly because they exist in VAX Fortran and their desirability is known. Note that in some cases the functionality is there, but with different syntax and/or extended semantics. The Fortran 8X functionality is noted in parenthesis. Although the difference in implementation ranges from trivial to ugly (% as the component delimeter in derived data types), there are always technical reasons for the resultant syntax. The standardized feature is more consistent with the language, and provides portability, and therefore will be the syntax of choice if portable code is the goal.

VAX EXTENSIONS AVAILABLE IN FORTRAN 8X:

    IMPLICIT NONE
    INCLUDE

DO WHILE
Long Variable Names
Extended Source Form (free form source)
Structures (derived data types)
BIT manipulation (MIL standard as implemented in
        VAX Fortran)

4. GENERIC ISSUES — RESPONSE TO THE FIRST PUBLIC REVIEW

These generic issues are also issues for compiler writers. They will be mentioned here and addressed further next month.

SIZE -- Many of the first public review responses expressed concern about the size of the proposed standard,stressing the need to retain the simplicity of Fortran.

The introduction of new features does not require their use. To the programmer, the simplicity (if there ever was any) of FORTRAN 77 still remains. The expansion of Fortran allows scientific programmers to combine the numeric features of traditional Fortran (mathematical intrinsics, precision etc) with modern coding techniques,if they wish. This combination gives the Fortran programmer the opportunity to try new features in the context of existing programs.

COMPLEXITY — The first public review responses expressed concern about the complexity as well. New features such as modules and the new form of precision, were hard to understand and use. I agree that some of the new features will require a change in Fortran programming style, and therefore can be viewed as hard to understand at the present time. This change in style however, will result in better, more maintainable code: a bonus for all software development!

Fortran 8X complexity is due in part to duplication of functionality, introduced in order to expand features of FORTRAN 77(e.g. add keywords for parameters in subroutines), increase consistency in syntax and/or semantics(e.g. attribute statements), and to improve the reliability of the language(e.g. name association). Although the new features provide consistent and complete functionality, upward compatibility constraints require the retention of the duplicated FORTRAN 77 features.

I believe that the complexity results from this duplication, and is not necessarily inherent in the features themselves.

DEPRECATED FEATURES -- As noted, some duplication was introduced into the language with the addition of new features. The concept of deprecation was added to reduce complexity in future standards by providing for the possibility of removal of unused features. The public review however was opposed to deprecating features such as COMMON and EQUIVALENCE. The concept of deprecated features was therefore removed from the new revision.

5. WHERE TO GO FROM HERE

This article has given a brief overview of the standard from a programmers point of view. Next month I will respond as a DEC customer.

There is no doubt that Fortran 8X is controversial. It is essential that your opinions are known. Please respond to the public review. The revised standard can be obtained from

Global Engineering
2805 McGaw
Irvine, Calif 92714
800 854-7179

Ask for
Document X3.9 programming language Fortran (revised 1989)

The cost is $50. If for any reason your company will not support you in sending for the document, fee free to contact me at (203) 432 3366. I will try to obtain a copy for you.

See you next month.

TECO WORKING GROUP REPORT

by Pete Siemsen

Yes, there is a TECO Working Group. As the librarian, I've collected enough strange TECO-related stuff to form the TECO Collection, version 1 of which has been submitted to the DECUS Software Library. It includes:

[.DOC]	The newest manual (.MEM) for "Standard" TECO, dated May 1985. This manual is newer than what DEC distributes. Also in here are v39 and v40 release notes, describing all kinds of goodies in TECO11 and TECO32, like callable TECO.
[.EMACS11]	Fred Fish's EMACS subset for TECO-11 v35 or higher.
[.LIDSTER]	Ken Lidster's macros and a documentation file that describes TECO initialization and how to customize it.
[.MACROS]	Best/latest versions of "classic" TECO macros from the rest of the collection.

[.RSTS]	TECO stuff from RSTS/E v9.5, thanks to Mark Derrick. Contains 1982 sources of VTEDIT, SQU, etc. with some documentation.
[.RSX...]	Everything I could find in the RSX SIG tapes relating to TECO.
[.SMITH]	Kelvin Smith's macros for munging BASIC under RSTS, documentation for TECO initialization for RSTS and VMS, and Kelvin's personal VTEDIT with documentation.
[.SOFLIB]	TECO entries from the DECUS Software Library. VTEDIT for VAX TPU, video editors for HP and Tektronix terminals, an EMACS-like package for RSTS/E TECO-11, more.
[.TECO11]	Source code for TECO-11 v36 (mixed mode for VMS).
[.TECO32_FOR_V4]	Native mode TECO32 as released with VMS v5+, but built under v4 so it will run under v4 (executable only).
[.TECOC]	Pete Siemsen's TECO in C for VAX/VMS (almost Unix and MS-DOS).
[.VMS...]	Teco stuff from a VMS SIG CD-Rom disc, 1984-1987.
[.UTECO]	Matt Fichtenbaum's TECO in C (Jul 89) for Ultrix and SunOS.
[.YMILES]	Ya'akov Miles's TECO in C v1.04 (12 Jun 88) for MS-DOS.

Source code comes with everything but what DEC owns. Some things that may be added in version 2 of the collection are: a video TECO written in C that executes TECO commands immediately as they are typed, another TECO in C, a TECO in 6502 assembly language, and a preprocessor that helps write TECO macros: it reads a structured language and produces TECO. Stop laughing.

This collection is also available via anonymous ftp for users of Internet (send me mail for details). Please send complaints, suggestions, additions, insults, etc. to

Pete Siemsen
645 Ohio Ave. #302
Long Beach, Ca. 90814

(213) 433-3059 (home)
(213) 743-0731 (work)
Internet: siemsen@usc.edu

SOURCES OF ADA COMPONENTS

Shirley Bockstahler-Brandt

Introduction

Several people asked about sources for Ada components at the last Ada working group meeting. This paper briefly describes four sources that I am aware of. There may well be others. The information given here represents my own opinions, and does not imply endorsement by DECUS, the L&T SIG, or JHU/APL.

Ada is a language that lends itself especially well to code sharing. Its packaging, stability, and portability are just some of the features that make sharing practical. Many people have realized this, and there are now several publicly-available sources of reusable Ada components. The four sources discussed here are the Booch Components, SIMTEL-20, CAMP, and the GRACE components.

All components are provided in source form, so you can modify the code if necessary.

The Booch Components

This general-purpose component library is available from Wizard Software, a company formed by Grady Booch. It is thoroughly documented in "Software Components with Ada," by Grady Booch, the Benjamin/Cummings Publishing Company, Inc., 1987. In this book Booch presents a tutorial on the concepts of reusability, a taxonomy for reusable components, and a complete description of his components, including data structures, tools (algorithms), and subsystems (cooperating structures and tools).

The Booch Components were developed on a Rational system, but we have had no trouble using them with VAX Ada. The license recognizes that the components will be incorporated into software that may be distributed to customers, and even permits distribution of the embedded source code when properly safeguarded.

> Ordering information: Wizard Software, 835 South Moore Street, Lakewood, Colorado, 80226.

SIMTEL-20

SIMTEL-20 is a general-purpose, free repository available to Arpanet users. It contains both tools and components as well as news. A glance through the directories yields topics such as SQL, GKS, AI, the Ada language reference manual, benchmarks, the components, TCP/IP, debugger, editors, education, kermit, GKS, management tools, math, metrics, news, PDL, PIWG (compiler comparison benchmark programs), and speller.

All the software is donated. Corrected code is sometimes submitted. All documentation is available on-line. Much of the SIMTEL-20 code was written for VAX/VMS.

Some or all of the SIMTEL-20 library was submitted to the Tapecopy Project at the Atlanta

symposium (spring 1989). Contact the DECUS Library or check the L&T and VAX SIG tapes. SIMTEL-20 is not available outside the U.S.

CAMP

CAMP is the Common Ada Missile Packages. It was developed by the McDonnell Douglas Astronautics Company for the U.S. Air Force, which was interested in reusable missile software components. Because of its missile orientation, CAMP is especially strong in mathematical and navigational components, including related data types and constants, navigation, Kalman filters, guidance and control, vector and matrix algebra, and polynomial math.

CAMP offers two Ada software products: the CAMP Parts and the CAMP Armonics Benchmarks. The CAMP Parts include the Ada components, test code, the "CAMP Parts Top-Level Design Document," and the "CAMP Parts Detailed Design Document." The Benchmarks are used to evaluate Ada and processor implementations; they are useful for making performance measurements.

CAMP was written on a VAX host.

> Ordering information: a CAMP information package can be obtained from:
>
> > Data & Analysis Center for Software
> > RADC/COED
> > Griffiss AFB, NY 13441-5700
> > 315-336-0937 or Autovon 587-3395
> > ATTN: CAMP Information Package

CAMP is export-controlled, and may be distributed only to U.S. Government Agencies and to organizations certified as qualified contractors by the Defense Logistics Services Center.

The GRACE Components

This is another general-purpose component library, offered by EVB Software Engineering. It is organized according to Booch's taxonomy, and includes some extensions. The algorithms, data structures, etc. are based on the standard literature. Design documentation is provided for each component.

> Ordering information: A brochure is available from EVB Software Engineering, Inc., 5320 Spectrum Drive, Frederick, Maryland 21701, 301-695-6960.
>
> > Shirley Bockstahler-Brandt
> > The Johns Hopkins University Applied Physics Laboratory
> > Johns Hopkins Road
> > Laurel, Maryland 20707
> > 301-953-6585
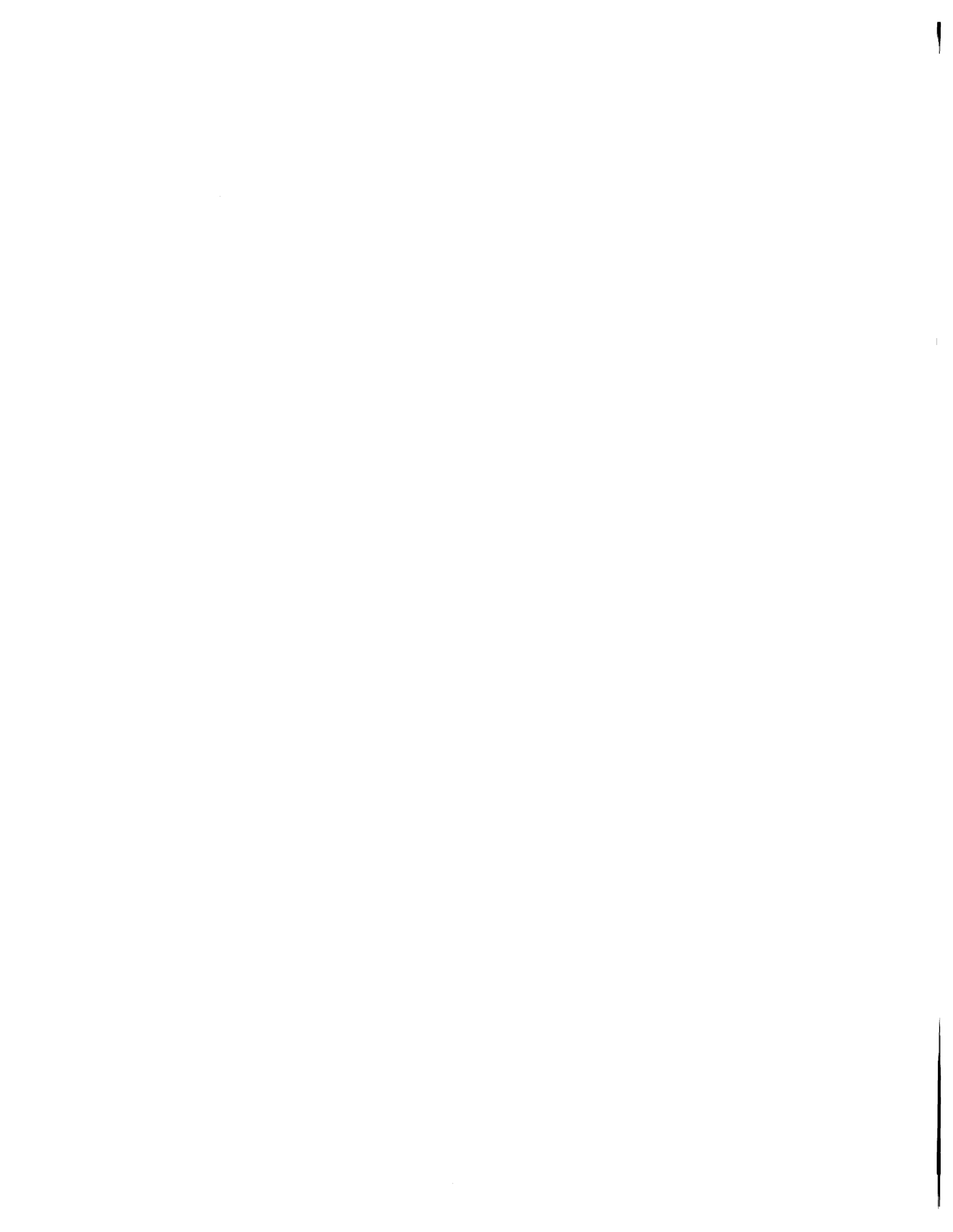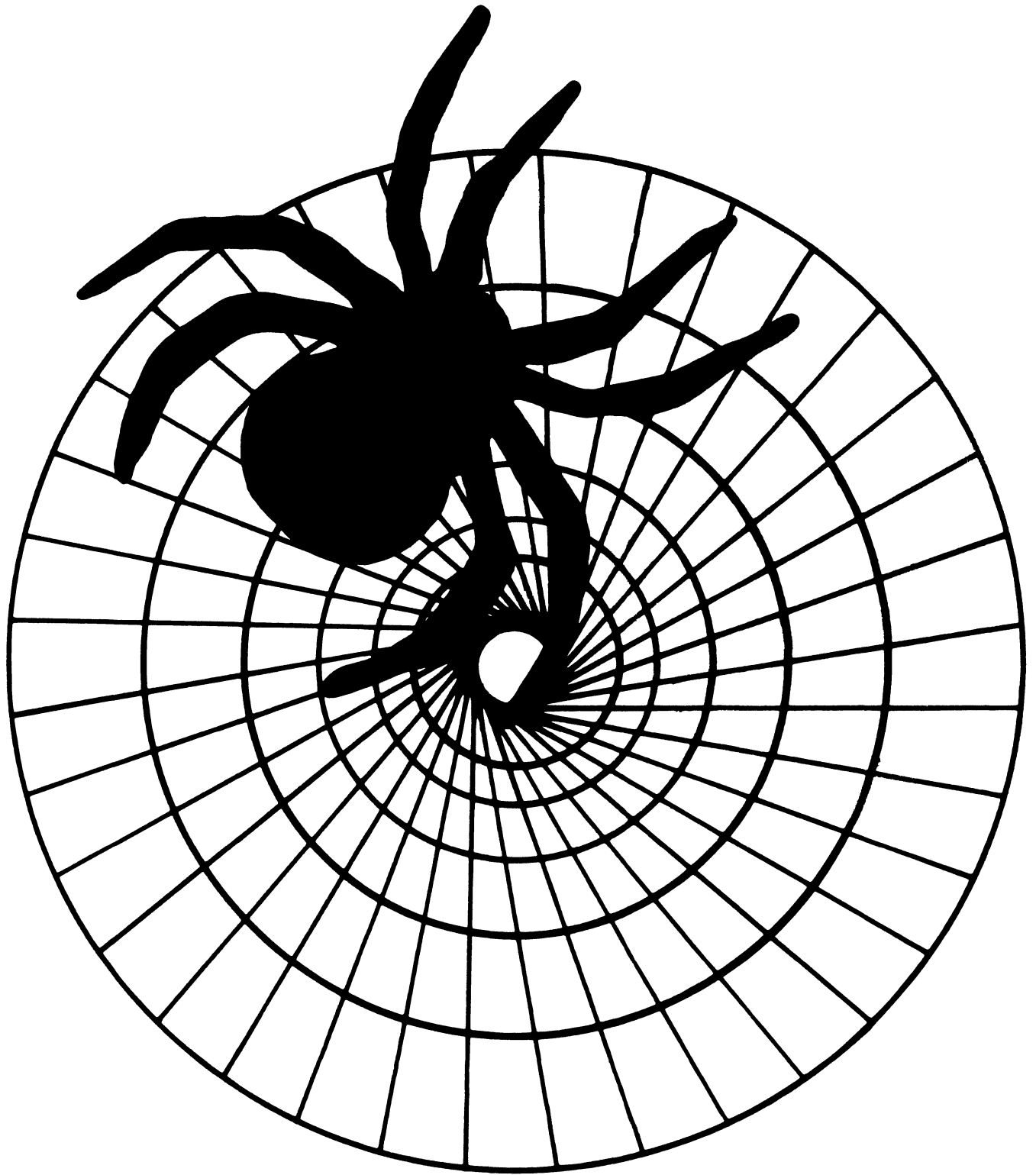
Networks SIG

OCTOBER 1989 NETWORK SIG NEWSLETTER

FROM THE EDITOR'S COBWEB

The Fall 1989 symposium will shortly be underway in lovely Anaheim California. I've taken a peek at the Sessions at a Glance (SAG), and am getting excited already (and I won't even be there!) It appears most of the network sessions will be held in the Convention Center Anaheim Room, or nearby. Just follow the crowds! Tuesday, first thing in the morning, 9:00 in the Center Hall of the Marriott, be sure not to miss the ever-popular "Understanding Ethernet". Have an early
breakfast so there'll be a good seat!

This month features a humorous look at Functional Specifications, submitted by Rick Carter, the Assistant Newsletter Editor, and a review of LTM as published in the CVLUG$OUTPUT, the award-winning news-letter of the Connecticut Valley LUG.

Enjoy the symposium if you are one of the lucky ones. Remember to share your gleanings by submitting articles to the newsletter!

Judi Mandl
Uconn Health Center
Administrative Data Processing
263 Farmington Ave.
Farmington, CT. 06032

FUNCTIONAL SPECIFICATIONS
Submitted by Rick Carter

Several years ago, a company I worked for had a requirement that all documentation (including technical documentation) had to be approved by Marketing. To prove the uselessness of this, someone in the company (no one would tell me who) imbedded the following as a chapter in a Functional Specification for a product. Marketing "read" and approved the entire document unchanged, and, in fact, it stayed in the manual for several years due to apparent oversight on the perpetrator's part. Everyone has denied being the perpetrator of this (it isn't me — honest!), so the author must remain anonymous. In addition, the product and company name have been concealed to protect my skin.

Rick Carter

FUNCTIONAL SPECIFICATIONS

HOST COMMUNICATIONS HOOKS.

The intent is to provide the XXX with the "hooks" for host communication by using the simple Terminal Protocol (TP). A general description of this technology follows: (Ref. = "Tutorial", S. Leibson, June, 1982 COMPUTER).

"TP terminology. Just what is involved in TP? The atomic level of TP, as with most computer technology, is based on the bit. However, just a bit of TP is not very useful. Bits are blocked into "sheets" much like frames in data-link protocols. Each sheet is seperated from adjacent sheets by flag-type characters called "perfs". A complete message may require several sheets forming a TP "roll". Finally, a session may require several rolls, thus making a "carton" of TP.

Although more complex protocols have seven layers, TPs have only one or two, called "plys". A two-ply TP consists of a physical or "bottom" layer and a logical or "top" layer. One-ply TP has only a physical layer. Generally, two-ply is considered superior to the one-ply variety.

Both token-passing and CSMA techniques are allowed in TP. The TP token is called a Generally Asynchronous Symbol (GAS) and in most TP systems is represented by the ASCII character CAN (decimal 24). Thus, token passing is called "passing GAS". It is permissible in TP to hold on to the token for a while if a message is almost ready for transmission. However, some manufacturers have abused this rule, designing products that hold the token far too long. This leads to situations where there is far too much "sitting on the CAN".

The medium used to transmit TP is generically called a "pipe", a term stolen from the Bell Laboratories' Unix system. The installation of a TP system is usually referred to as "installing the plumbing". One of the great advantages of TP over the more complex local network protocols is that many office buildings are already "plumbed" and are ready to accept TP systems without additional cabling.

FUNCTIONAL SPECIFICATIONS

TP NETWORKS.

Several manufacturers have adopted TPs for their local network implementations. Thus we are seeing such new networks at the CharmiNET, and NortherNET, and of course, the CoroNET. These protocols allow some data compression techniques — except for CharmiNET, which allows no squeezing at all.

TP PERFORMANCE.

Although TP is a rather mature technology, no clear measure of performance has surfaced. This has left manufacturers to define their own performance standards. Thus, purchasers of TP systems must compare systems that are not rated equally. For example, CharmiNET is rated by its manufacturer in sheets per second (Whipples).

The manufacturer of NortherNET, which uses CSMA argues that Whipples do not accurately represent a system's performance under even moderate loading. Instead, NortherNET systems are ranked on a "Stroftness" scale to indicate how well they perform at various loading levels. CoroNET, billed as an economy system, has no performance rating at all.

As in any technology, TP manufacturers are always trying to improve the performance of their product. Recently, an advanced TP has been introduced to address the collision problem in CSMA systems. Called "Protocol Link Undoing Network Great Error Rates" (PLUNGER), this new TP is seen by many as a possible replacement for all TPs. As of this writing, however, no PLUNGER has successfully been built."

ABC, Inc's contribution to the TP technology is the recently released, state-of-the-art, Communications Requirements Analysis Program (CRAP). This program allows us to instantaneously determine customer requirements for host communications and respond appropriately.

THOUGHTS FROM YAR (Yet Another Reviewer...) (ANONYMOUS, As published in CONNECTICUT VALLEY LUG NEWSLETTER)

LTM, LAN TRAFFIC MONITOR

This is the first article of a first time series writer in need of direction. In the words of person who is drowning in random thoughts HELP!!! So I have asked for your feedback and removed the first hurdle. Now you're thinking my gosh golly gee what is the guy about to do a series on...how about some of DEC's network management products?

The first product that comes to view on my shelf is LTM. My, another pneumonic you say; right, LAN Traffic Monitor. LAN is Local Area Network for those who care. The hardware requirements include a system on version 5.x, a LAN Bridge 100, ethernet cables, an ethernet segment we need to monitor, and the VT24x, VT330 or VT340 terminal to display the data. The product is installed on a host VAX and is downline loaded into the LAN Bridge. The Bridge at this point is no longer able to perform bridge functions but takes on the role of a monitor. The traffic on your network starts to show LTM packets, which is the monitor reporting traffic statistics back to your host node LTM user interface.

The monitor software on the host node uses menu driven display panels to show traffic summaries, displays of traffic by node and type, and long term, current and peak utilization and throughput. A display on the node gives the addresses as a default. The names for the nodes are supplied by you (a way to detect tappers?). The traffic is broken down by address with counts, frames and % of total displayed. The PEAK LAN utilization and throughput screen is quite useful to see the loading conditions on the segment in the worst case. Like most monitors, it traps the date/time and values on utilization,frames/second and multicast packets. These are but two of many screens available. The interface allows the user to move throught these screens by number and the keypad function keys (another set to learn). The other feature is the ability to graph utilization on a ReGIS terminal. These graphs are fixed to seven types based on current to 48 hours. This is just a brief brush over of LTM. I hope its worth something to you. I'll be back next issue if I'm not heavily trashed by negative feedback.

BY THE WAY, WOULD I BUY LTM? Yes, if I had a backup LAN Bridge 100, or one I could spare for a few days; after all, cost is everything, and bridges aren't cheap, (but then again neither are standalone monitors).

Office Automation

IN THIS ISSUE...

OA Bulletin Board

The OA SIG moderates several on-line bulletin board conferences for discussion of OA problems and solutions. These conferences are available on the DECUServe system.

Watch upcoming issues for more information on DECUServe.

FROM THE EDITOR...

Try your interpretive skills on this pseudo-script:

! changes.scp 21-Aug-1989 18:27

```
for decus_newsletter -
with .section eqs "OASIG" and .editor eqs "Therese LeBlanc" -
do write change decus_newsletter %key=.key, -
editor="Roger Bruner", -
effective_date="October, 1989"
```

If your interpretation of "changes.scp" led you to realize that the OASIG NEWSLETTER is losing Therese LeBlanc as its editor (effective with this month's issue) and getting me, I commend you on your scripting.

I feel like someone who has just adopted an older child. The OASIG NEWSLETTER was Therese's for so long and has taken on a number of her VERY FINE qualities. It's quite a prospect to live up to her standards, but I am looking forward to the challenge.

Actually, the change took place (quite unexpectedly!) a month earlier than Therese or I had anticipated, so my first issue won't quite be what I had hoped. Since I received some good-natured teasing in Atlanta about this being the "Roger Bruner Newsletter", let me assure you that I will take every step possible to prevent that from being true in the future!

One change you will see starting next month is that I hope to make extensive use of the Macintosh and PAGEMAKER. The Macintosh will also allow me to use an HP scanner and DESK GALLERY to include photographs and graphics. The results will be interesting. I would certainly like to be able to include a photographic layout of the OA SIG AT ANAHEIM.

Let me hear from you — what you like and don't like, what's beneficial and what's not. Each of you is an expert on something related to Office Automation, and the rest of us would greatly benefit from your sharing with us in the form of even a short article. It is especially important that we all broaden our view of what Office Automation is.

I have a number of articles that were sent to Therese in June, July and August. They will be appearing in upcoming issues. The November issue will contain several articles related to the Anaheim Symposia; hopefully they will be timely and not after the fact! Topics are getting the most benefit possible from symposia and information about the VTX working group.

Roger Ellis Bruner
OA SIG Newsletter Editor
Foreign Mission Board
Box 6767
Richmond, VA 23230
804/353-0151

UDP LOOPING WITHIN WPSPLUS

Roger Bruner, Foreign Mission Board

Thanks to a 1988 OASIS note (WPSPLUS #29.2) posted by Gail Katagiri of Genentech, Inc., some of my users have a solution to a long-standing need! Gail explained that a symbol, OAMSG__ID, is set by ALL-IN-1 to "ERRORS WPL__ATBOTTOM" when the user tries to go beyond the bottom of the document in WPSPLUS. A UDP can be built to loop until that condition is reached. However, because of the "Already at bottom...press any key to continue." message, the UDP needs to include a key to clear that message and allow ALL-IN-1 to test the value of OAMSG__ID. Gail had included a carat (^) as a throw-away character for this purpose. For her needs, the carat got deleted along with the other things her UDP was searching for and deleting.

Unfortunately, the presence of the carat interfered with what I needed to do. After seemingly hours of combing the keyboard for something that wouldn't display or create havoc, I settled on a {CTRL W}, since it would only refresh the screen unnecessarily, but at least wouldn't add anything to the document itself.

Then I discovered that an unsuccessful search within WPSPLUS sets the "OAMSG__ID" to "ERRORS ERR__SRCHNOTFND" rather than "WPL__ATBOTTOM", so it would be necessary for the UDP to test for both conditions.

I passed all of this information along to our resident UDP expert, and we worked together to create the following UDP. First let me describe its purpose; then the UDP itself will make more sense.

The ladies in our Word Processing Section often create multiple letters in one document. They need to cut the name and address from each one to include in a separate document which is printed on mailing labels, rather than having to key in the addresses separately.

To use this UDP, they put an ASTERISK at the beginning of each ADDRESS as they key it in and a "@" at the beginning of the FIRST address. The UDP then searches for an ASTERISK, deletes it, does a SELECT & GOLD CUT of the paragraph containing the ADDRESS, and PASTEs it at the bottom of the document. The UDP returns to the TOP and looks for the next ASTERISK, repeating the process until there are no more to be found. Then the UDP locates the "@", deletes it, CUTS to the bottom of the document, and goes out and PASTES the addresses in the ADDRESS document.

WARNING: There is some indication that the OAMSG_ID symbol may not be used in 2.3 of ALL-IN-1. We will have to ask our DEC counterparts for their help in identifying the comparable 2.3 symbol names and values.

```
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!Name of UDP:    ADD1
!
!Description of UDP:  Pulls addresses from a document containing several
!letters.  Puts them on bottom of document and then cuts and pastes in
!TENV document.
!
!Programmer's Name:  Bruner and Kellams              Date:  Aug. 3, 1989
!
!Special Instructions for UDP:  An asterisk and an @ sign needs to be put
!in front of the first address and an asterisk in front of addresses
!thereafter.
!
!===============================================================================
!
.LABEL BEGIN
!1ST TIME TO SEE IF ANY ASTERISK
.FX GET $OA$MSG_ID=""
{GOLD ,}
*{CR}
.IF $OA$MSG_ID EQS "ERRORS ERR_SRCHNOTFND" -
       OR $OA$MSG_ID EQS "ERRORS WPL_ATBOTTOM" -
            THEN .GOTO EXIT
.GOTO PROCESS1

.LABEL LOOP
!PROCESS THE DOCUMENT AT THIS POINT IF ASTERISK WAS FOUND
.FX GET $OA$MSG_ID=""
{GOLD .}{CTRL W}
.IF $OA$MSG_ID EQS "ERRORS ERR_SRCHNOTFND" -
       OR $OA$MSG_ID EQS "ERRORS WPL_ATBOTTOM" -
            THEN .GOTO PROCESS2
!REPEAT PROCESS IF ASTERISK IS FOUND

.LABEL PROCESS1
{DELCHR}
{SEL}
{PARA}
{GOLD CUT}
{GOLD B}
{PASTE}
{GOLD N}
{GOLD T}
.GOTO LOOP
```

```
.LABEL PROCESS2
!Find the @ and delete it...
{GOLD ,}
@
{BACKUP}
{DELCHR}
!Now SELect and CUT the rest of the document...
{SEL}{GOLD B}
{CUT}
!File the current document, SELect the ADDRESS document (TENV), and
!PASTE the ADDRESSES in...
{GOLD F}
{RETURN}
SEL
{RETURN}
TENV
{RETURN}
E
{RETURN}
{GOLD B}
{PASTE}
Y
{RETURN}
{GOLD F}

.LABEL EXIT
.EXIT
```

—   —

WANTED: Session Chairs for Fall '89 Anaheim Symposium

Lynda Peach, Mustang Fuel Corporation

If you've been thinking that you might like to get more involved with the OA SIG at the next symposium but don't want or cannot commit to a long-range job, then we have just the thing for you! Become a VOLUNTEER session chair. It's easy and fun. You will meet a lot of new people and feel more involved in what's going on.

What does a session chair do?

Session chairs welcome the group and introduce the speaker. They try to make sure that the session runs smoothly (dim lights if necessary, flip overheads, etc.). They also remind everyone during the question and answer period to speak clearly into the microphone so that the whole group can hear the question. Session chairs also have direct input to the symposium committee through room count and comments about the quality and audience receptiveness of the session.

How long does it take?

That depends on the length of the session. Choose sessions you are interested in, then be there a few minutes before the session begins to meet the speaker, find out how to pronounce their name and a little something for the introduction. That's it!

Will I get any special instructions from DECUS?

There is always a short meeting on Sunday evening before the Welcoming Reception just for Session Chairs. DECUS will give you some specific information and a quick overview of "How to be a good session chair."

It sounds great! How do I sign up?

In order of preference:

1) Write:
Terry Griggs
Strohl Systems/OAS
661 W. Germantown Pike
Suite 100
Plymouth Meeting, PA 19462

2) Call Terry: (215) 825-6220

3) Call me at (405) 557-9513.

For all the above include your full name, mailing address, and phone number. You may request a specific session(s) or type of sessions (technical, WPS-PLUS, mailbus, EDI, ALL-IN-1 programming, etc.). You'll then be contacted by Terry.

NOTE

Requests will be honored on a first-come, first-served basis.

Being a session chair is a great way to meet the speaker(s) and you're definitely guaranteed a "chair"!!!! It's easy to do but a vital job that must be done for every session given at Symposium. Help make Anaheim the best Symposium ever! Participate!

PERSONAL COMPUTER

SPECIAL INTEREST GROUP

NEWSLETTER

TABLE OF CONTENTS

Rainbow Section

PCSA Section

Rainbow Section

Rainbow Bibliography - Part 3: the Letter C
By Dr. Thomas Warren, PC SIG Session Notes Editor
Copyright © 1989, Rainbow News

The Bibliography that follows is reprinted here in serialized form with permission of:
Rainbow News, P.O. Box 567, O'Fallon, IL 62269, (618)632-1143

What follows is a selected bibliography of articles on the Rainbow. It is selective because it is not complete and not complete because I have not seen everything available. It is, however, complete enough to get the interested party started.

That is a small hint. Let me make a bigger one. IF YOU KNOW OF RAINBOW ARTICLES, PUBLICATIONS, BOOKS, ETC. THAT AREN'T LISTED HERE, PLEASE CONTACT ONE OF THE PC SIG STEERING COMMITTEE. Your input to this monumental effort on Tom Warren's part is VERY MUCH DESIRED! Our addresses and phone numbers appear at the back of these Newsletters. Ed.

Each section is headed by a KEYWORD, a list of which are attached in an appendix.

This month, my quota of 25 pages allows me to include the letter "C" of the bibliography. Ed.

CABLES
"Hardware and Software Available", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 1, No. 7 (Nov., 1984), 8-9. (Terminal, E-Mail, Desktop, WordStar, LA100, CP/M, ADA, Graphics, Printer, MS-DOS, Accounting, Cables)

CACHE
"New Products Announcements: Speedcache: Disk Caching Utility Software for the DEC Rainbow", RAINBOW NEWS, Vol. 4 , No. 5-6 (May-June, 1987), 18. (Cache, Speed)

"New Product Announcements: Speedcache: Disk Utility Software for the DEC Rainbow", RAINBOW NEWS, Vol. 4, No. 5-6 (May/June 1987), 18. (Memory, Programs, Cache, RAM)

CALC
"Application Software: SuperCalc3", PERSPECTIVE, Vol. 4, No. 1 (n.d.), 10. (Spreadsheet, Graphics)

Chorney, Victor J. "SuperCalc3: Was it Worth the Wait?" THE DEC PROFESSIONAL, Vol. 5, No. 12 (December 1986), 68, 70-71, 74. (Software, Review, Spreadsheet)

"CodeBlue Reviews", RAINBOW NEWS, Vol. 4, No. 3-4 (March/April, 1987), 22. (Games, CB, HOOPS, Disk Optimizer, Wordprocessing, METHODS, SmallTalk, PC Scheme,PC-File, PC-Calc, BRIEF)

"Software Available for the DEC Rainbow", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 1, No. 3 (July, 1984), 5. Investment Manager, Calc, Graphwriter, Communications, WordEase, Programming, C)

CARDS
Miller, Jerry. "Project Transport", THE DEC MICROLETTER, Vol. 1, No. 3 (n.d. [1987]), 31-32. (Graphics, Monitor, Cards, EGA)

CARE
McAfoos, Bob. "DEC Rainbow Maintenance: Notes From the Rainbow Pacific User Group Meeting--June, 1986", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 3, Nos. 7-8 (July-Sept. 1986), 30-31. (Care, Service)

"More DEC on Diskettes and Drives, "WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 12 (Dec. 1985), 12-13. (Maintenance, Care, Operating Conditions)

CATALOGS
Holman, Katrina. "Rainbow Resources: Free for the Asking", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 3 Nos. 7-8 (July-Sept. 1986), 21-22. (Books, Plastic Bags, PERSPECTIVES, Catalogs)

CB (See also CodeBlue)
"CodeBlue Reviews", RAINBOW NEWS, Vol. 4, No. 3-4 (March/April, 1987), 22. (Games, CB, HOOPS, Disk Optimizer, Wordprocessing, METHODS, SmallTalk, PC Scheme,PC-File, PC-Calc, BRIEF)

Graves, Tom. "Software Review: dBaseIII Plus on the Rainbow with CODEBLUE", RAINBOW NEWS, Vol. 4, No. 3-4 (March/April, 1987), 13-14. (CB, dBaseIII+, SmartKey)

CHARACTER SETS
Camas, Tony. "The I/O Port", THE DEC MICROLETTER, Vol. 1, No. 3 (n.d. [1987]), 24-26, [36]. (Character Sets, COMPOSE, LA100, LQP02, LQP03, Concurrent CP/M, Select, CONDOR, POKE, Video Memory, MDRIVE, MBASIC)

CHARACTERISTICS
Hersh, Sanford. "Why Buy a Rainbow?" WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 11 (Nov. 1985), 9-10. (Hardware, Characteristics)

CHARTS
Saul, Bob. "Product Review: Graftalk", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 6, No. 3 (June 1986), 10-11. (Software, Graphics, Charts)

CHESS INGENUITY
Metsger, D. Scott. "Software Review: Two Computer Chess Programs", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 3, Nos. 7-8 (July-Sept. 1986), 8-10, 12. ("Chess Ingenuity", ChessWright)

CHESSWRIGHT
Metsger, D. Scott. "Software Review: Two Computer Chess Programs", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 3, Nos. 7-8 (July-Sept. 1986), 8-10, 12. ("Chess Ingenuity", ChessWright)

"Software and Hardware", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 1 (January, 1985), 14. (ChessWright, BASIC, Compiler, Graphics, Database)

CHIPS
Barfield, Ed. "Technical Perspectives: Inside the Rainbow . . . A Conversation Between Chips",

PERSPECTIVE, Vol. 2, No. 2 (May 1984), 13-14. (CP/M, BIOS, Programming)

Eliopoulos, Rick. "A Dissertation on the Differences Between 150 and 200 Nanosecond Dynamic RAM Chips", PC-SIG NEWSLETTER, Vol. 2, No. 4 (June, 1985), 32-33. (Hardware, Third_Party)

Eliopoulos, Rick. "A Dissertation on the Differences Between 150 and 200 Nanosecond Dynamic Chips", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 6 (June, 1985), 5-6. (Hardware, Speed, Chips)

"Follow Up: An Addendum to Tom Tugman's Article About Adding Memory Chips", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 7 (July, 1985), 11-12. (Hardware, Speed)

Tugman, Tom. "An Update on Installing 896K", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 6 (June, 1985), 6. (Chips, Hardware, Memory)

CLIKCLOCK
Maaskant, Barbara A. and Ted Needleman. "The ClikClok: Two Views", THE DEC MICROLETTER, Vol. 1, No. 2 (January/February, 1987), 17, 26. (Review, Clocks, ClikClok, Hardware)

Tugman, Tom. "Hardware Review: Clikclok", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 3, Nos. 11-12 (Nov-Dec.1986), 12. (Clocks)

CLIP.COM
Jackson, Bruce. "MS-DOS is not User-Friendly--But I'll Tell You Who is . . . ", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 3, No. 1 (Oct. 1986), 14-17. (Utilities, History.Exe, CP 1.9, DE 1.2, VTREE.COM, TREE.COM, TREEDIR.COM, RN.COM, CLIP.COM, FREE.COM, DISK31.COM, DJINN, AME86.EXE, CLRTSR.ARC [MARK.COM, RELEASE.COM], RR.COM, HEAD.EXE, FASTLAST.EXE STRINGS.COM RB-BUFFER.EXE)

CLOCKS
Maaskant, Barbara A. and Ted Needleman. "The ClikClok: Two Views", THE DEC MICROLETTER, Vol. 1, No. 2 (January/February, 1987), 17, 26. (Review, Clocks, ClikClok, Hardware)

Miller, Jerry. "RDS 8087/Clock Board Fix", RAINBOW NEWS, Vol. 4, No. 10-12 (Oct.-Dec., 1987), 34-35. (Hardware, Modifications, 8087, Clocks)

Needleman, Theodore. "Rainbow Corner", HARDCOPY, Vol. 7, No. 1 (Januayr 1987), 77. (Hardware, Clocks,, I-Drive, Harddisk, RB-Link, Switch-It, Pascal, Newsletters)

Needleman, Theodore. "Improving Rainbow Productivity", HARDCOPY, Vol. 6, Nol. 7 (July 1986), 99-105. (Hardware, RB-LINK, I-Drive, Harddisk, SunDisk, Clocks)

Needleman, Theodore. "Rainbow COrner", HARDCOPY, Vol. 6, No. 7 (July 1986), 109. (FIDO, CodeBlue, Clocks, Formatting, Harddisk, I-Drive, RB-LINK)

"New Rainbow Products Found at DEXPO", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 3, Nos. 11-12 (Nov-Dec. 1986), 5. (Products, Clocks, I-Drive, Harddisk, Tape, Backup, CodeBlue, RB-Link, WPS-DOS)

Trelease, Robert B. "Rainbow Clock Boards", THE DEC PROFESSIONAL, Vol. 5, No. 6 (June 1986), 66-68. (Hardware, Clocks)

Tugman, Tom. "Hardware Review: Clikclok", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 3, Nos. 11-12 (Nov-Dec. 1986), 12. (Clocks)

CLOCKSPEED

Strybos, Gabe. "Update: Benefits of Installing the V-20 in the Rainbow", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 3, Nos. 7-8 (July-Sept. 1986), 29. (Hardware, Modifying, Clockspeed, Modifications)

CLONE

Vince, Paul. "Running DEC Printers on an IBM or Clone Computer", RAINBOW NEWS, Vol. 4, No. 5-6 (May-June, 1987), 20-21. (Hardware)

Vince, Paul. "Running DEC Printers on an IBM or Clone Computer", RAINBOW NEWS, Vol. 4, No. 5-6 (May/June), 20-21. (LA50, Hardware, Modifying)

CLRTSR.ARC

Jackson, Bruce. "MS-DOS is not User-Friendly--But I'll Tell You Who is . . . ", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 3, No. 1 (Oct. 1986), 14-17. (Utilities, History.Exe, CP 1.9, DE 1.2, VTREE.COM, TREE.COM, TREEDIR.COM, RN.COM, CLIP.COM, FREE.COM, DISK31.COM, DJINN, AME86.EXE, CLRTSR.ARC [MARK.COM, RELEASE.COM], RR.COM, HEAD.EXE, FASTLAST.EXE STRINGS.COM RB-BUFFER.EXE)

COBOL

Freeman, Neil. "Programming Languages", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 3, No. 3 (March 1986), 13-14. (BASIC, PASCAL, COBOL, RPF, C, LISP, PROLOG, MUMPS, DIBOL, SPSS, FORTRAN, SNOBOL, APL, PL/1, ALGOL, ADA, PILOT)

"Product Announcement: STSC Announces APL*PLUS for DEC Rainbow", WASHINGTON AREA USERS GROUP NEWSLETTER, Vol.2, No. 2 (February, 1985), 2-3. APL, Langauges, BASIC, FORTRAN, COBOL, Pascal, Graphics)

CODEBLUE

"Code Blue and Compatible News", RAINBOW NEWS, Vol. 4, No. 7-9 (July-Sept., 1987), 20. (CodeBlue, RB-LINK)

"CodeBlue Reviews", RAINBOW NEWS, Vol. 4, No. 3-4 (March/April, 1987), 22. (Games, CB, HOOPS, Disk Optimizer, Wordprocessing, METHODS, SmallTalk, PC Scheme,PC-File, PC-Calc, BRIEF)

"Compatibility", RAINBOW NEWS, Vol. 4 No. 10-12 (Oct.-Dec., 1987), 25. (CodeBlue, RB-LINK)

Graves, Thomas E. "Software Review: Code Blue, Version 2.0", RAINBOW NEWS, Vol. 4, No. 10-12 (Oct.-Dec., 1987), 17-21. (CodeBlue, Compatibility)

Graves, Tom. "Software Review: dBaseIII Plus on the Rainbow with CODEBLUE", RAINBOW NEWS, Vol. 4, No. 3-4 (March/April, 1987), 13-14. (CB, dBaseIII+, SmartKey)

Macmillian, Michael. "Preliminary Comments on Code Blue", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 3, No. 10 (Oct. 1986), 12-14. (CB, CodeBlue)

Miller, Jerry. "Project Transport", THE DEC MICROLETTER, Vol. 1, No. 2 (January/February, 1987), 31-32. (Editors, Editing, WordPerfect, SEDT, EDT, DED, BRIEF, Review, CodeBlue)

Miller, Jerry and Julie Starr. "Project Transport", THE DEC MICROLETTER, Vol. 1, No. 1 (Nov.1986), 6, 15. (IBM, Compatibility, Software, Codeblue, CB, PC-Outline)

Needleman, Theodore. "Rainbow Corner", HARDCOPY, Vol. 6, No. 9 (September 1986), 88. (CodeBlue, Bulletinboards, FIDO, Public Domain Software, Jackson, Compuserve, Formatting, WUTIL)

Needleman, Theodore. "Rainbow COrner", HARDCOPY, Vol. 6, No. 7 (July 1986), 109. (FIDO, CodeBlue, Clocks, Formatting, Harddisk, I-Drive, RB-LINK)

"New Product Announcements", RAINBOW NEWS, Vol. 4, No. 3-4 (March, 1987), 18-19. (Lotus, CodeBlue, VENIX, LC-TERM, IBM, Shareware)

"New Rainbow Products Found at DEXPO", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 3, Nos. 11-12 (Nov-Dec. 1986), 5. (Products, Clocks, I-Drive, Harddisk, Tape, Backup, CodeBlue, RB-Link, WPS-DOS)

Starr, Julia B. "adding Functionality to the Rainbow", RAINBOW NEWS, Vol. 4, No. 10-12 (Oct.-Dec., 1987), 22-25. (Hardware, PC-DOS, MS-DOS, Harddisk, IBM, I-DRIVE, RB-LINK, CodeBlue, Memory)

COGEN
Needleman, Ted. "Easing the Pain of Application Creation", HARDCOPY, Vol. 14, No. 10 (October 1985), 70-74, 76-77, 78-80, 82-83. (Kaleidoscope, Cogen, Quepro, Data Management, Data Flex, Datavu, Dataease)

COMDEX
Starr, Julie. "Observations", THE DEC MICROLETTER, Vol. 1, No. 2 (January/February, 1987), 23, 32. (DEXPO, COMDEX, IBM, DECUS, VAX, VAXmate, Swap)

COMMUNICATIONS
"An Extra Telephone Line for the Computer--How Much will it Cost?" WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 1, No. 7 (Nov., 1984), 5. (Communications, Modem)

"Application Software: Communications", PERSPECTIVE, Vol. 1, No. 2 (July 1983), 23-24. (POLY-XFR, POLY-TRM)

"Application Software: Context MBA: An Integrated Softwear Package for the Rainbow 100", PERSPECTIVE, Vol. 2, No. 3 (October 1984), 32-33. (Spreadsheet, Graphics, Wordprocessing, Data Management, Communications)

Barron, Donna. "Smartcom", THE DEC PROFESSIONAL, Vol. 5, No. 10 (October 1986), 100, 102-103. (Communications, Software, File-Transfer, Modem)

Bernstein, Amy. "A Common VAX/PX-Based Spreadsheet is the Right Fit for One Telecommunications Manufacturer", HARDCOPY, Vol. 6, No. 11 (November 1986), 144, 146-148. (20/20)

"Books", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 1, No. 7 (Nov., 1984), 9. (Bibliography, Communications, MultiPlan, Lotus, Graphics, WordStar)

Brown, Mark. "Can We Talk?" HARDCOPY, Vol. 6, No. 4 (April 1986), 26-28, 36, 38, 40. (Communications, Modem)

Chorney, Victor J. "Blast", THE DEC PROFESSIONAL, Vol. 6, No. 4 (April 1987), 112, 114-116. (Communications, Software, File-Transfer, KERMIT)

Chorney, Victor J. "Communications: Blast", DEC PROFESSIONAL, Vol. 6, No. 4 (April, 1987), 112-116. (File-transfer)

Chorney, Victor J. "Look", THE DEC PROFESSIONAL, Vol. 5, No. 7 (July 1986), 72-73. (Graphics, Communications)

Chorney, Victor J. "MOBIUS: Establishing a Software Rapport Between the VAX and your Rainbow", THE DED PROFESSIONAL, Vol. 5, No. 6 (June 1986), 98-101, 104-105. (Communications, Emulator)

Chorney, Victor J. "Reflection 2 Plus", THE DEC PROFESSIONAL, Vol. 6, No. 2 (February 1987), 82, 84-87. (File-Transfer, Emulator, Communications)

Chorney, Victor J. "Symphony", THE DEC PROFESSIONAL, Vol. 4, No. 11 (November 1985), 61-62, 64-66, 68-69. (Spreadsheet, Wordprocessing, Review, Database Management, Communications)

"Communicating With Your Personal Computer", PERSPECTIVE, Vol. 2, No. 3 (October 1984), 20-23. (Communications, Networks)

"DEC Oriented Bulletin Boards", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 1, No. 7 (Nov., 1984), 9. (BBS, Communications)

Edwards, Brian. "The 2400bps Modem Age", HARDCOPY, Vol. 14, No. 10 (October 1985), 42-46, 48-50. (Hardware, Communications)

Edwards, Brian. "Share and Share Alike", HARDCOPY, Vol. 6, No. 2 (February 1986), 90, 92. (Communications, POLY-SHARE, POLY-XFR)

Etzbach, Donald. "Integrating PC LANs into the VAX Environment", HARDCOPY, October, 1986, pp. 159-164. (Communications)

Ference, Jack. "POLY/XFR MS-DOS Compatible Upgrade", PC-SIG NEWSLETTER, Vol. 2, No. 4 (June, 1985), 26. (Communications, CP/M)

Ference, Jack. "POLY/XFR Now MS-DOS Compatible", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 1, No. 4-5 (August- Sept., 1984), 6. (Poly-XFR, POLY-TRM, POLYCOM, Communications)

Fitzgerald, Dennis. "Rainbow Communications", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 3, No. 10 (Oct. 1986), 27-29. (Modem, Tutorial)

Fitzgerald, Dennis K. "Software Review: Mobius", RAINBOW NEWS, Vol. 4, No. 3-4 (March/April, 1987), 11-13. (Communications, VAX, IBM, Terminal, Integration)

Garbera, Barbara. "Rainbow to DECmate and Back Using POLYCOM", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 7 (July, 1985), 10. (Communications, CP/M)

Glossbrenner, Alfred. GOING ONLINE: COMMUNICATIONS ON THE DEC RAINBOW. Billerica, MA: Digital Press, 1984. (BBS, Bibliography, Modem, POLY-XFR, Games)

Glossbrenner, Alfred. PERSONAL COMPUTER COMMUNICATIONS FOR THE DEC RAINBOW. Bedford, MA: Digital Press, n.d.

Graves, Thomas E. "Communications Tutorial: Navigating the Compuserve Information Service", RAINBOW NEWS, Vol. 4, No. 3-4 (March/April, 1987), 22-29. (BBS)

Garbera, Alexander R. "DECmate and Rainbow Communications", PERSPECTIVE, Vol. 3, No. 3 (n.d.), 21. Graham, Chad. "Communications: Electronic Mail", RAINBOW NEWS, Vol. 4, No. 5-6 (May-June 1987), 21-22. (CompuServe, The Source, EasyPlex, FIDO)

Holman, Katrina. "Linkware for Rainbow and IBM Personal Computers", WAHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 3, Nos. 7-8 (July-Sept. 1986), 16-17. (Communications, VAX)

Jack, Steve. "Communications Tip", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 7 (July, 1985), 12. (LCTERM, FIDO, Hardware)

Jackson, Bruce. "Communications: Pibterm Available for the Rainbow", RAINBOW NEWS, Vol. 4, Nos. 1-2 (Jan-Feb, 1987), 13-14. (Public Domain Software)

Jackson, Bruce. "Public Domain Software Review: MINITEL (DECMini)", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 3, No. 3 (March 1986), 6-7. (Communications, File Transfer, FIDO)

Jackson, Bruce. "Public Domain Software Update: PIBTERM Modifications", RAINBOW NEWS, Vol. 4, No. 3-4 (March/April, 1987), 18. (Communications)

Jackson, Bruce. "Review: GTE's PC Pursuit . . . All you do is Run", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 3, Nos. 11-12 (Nov-Dec. 1986), 16-17. (Communications, modems)

"January Meeting Notes", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 2 (February, 1985), 1-2. (Communications, PolyCom, PolyTrm, Kermit, Terminal, Emulator, ReGIS, Graphics, MS-DOS, Knowledgeman, Multiplan, CP/M, FIDO)

"The Kermit Communications Protocol", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 1, No3 (July, 1985), 4. (Software, Modem)

"Kermit: A File Transfer Protocol for Universities: Part I: Design Considerations and Specifications", BYTE, June, 1984, pp. 255-278. (Communications, Modem, Software)

Leehouts, Mark. "Rainbow to DECmate and Back--Without Communications Software", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 12 (Dec. 1985), 10. (Modem, Hardware, File Transfer)

Leeman, Bill. "Going Online: Compuserve", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 3, Nos. 7-8 (July-Sept. 1986), 23. (Communications, Databases)

"Letters to the Editor", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 8-9 (August-Sept., 1985), 13-14. (LISP, Communications, Modem, LA50, Hardware, Genealogy, Pascal)

"Modem/Multiplexer Buyer's Guide", HARDCOPY, Vol. 14, No. 10 (October 1985), 52-60. (Hardware, Modem, Communications)

Needleman, Ted. "Communicating with your Rainbow", HARDCOPY, Vol. 6, No. 6 (June 1986), 153-154. (NOTE: Seems to have been a Volumn change. Use date.) (Modems, Communications, Hardware, LCTerm, DECMini, MiniTel, Xmodem, Kermit)

Needleman, Theodore. "Rainbow Corner", HARDCCOPY, Vol. 15, No. 1 (January 1986), 110. (Software, Accounting, Communications)

"November Meeting Notes", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 1, No. [8] (Dec., 1984), 1. (Public Domain Software, CP/M, MS-DOS, Utilities, Wordprocessing, Communications, Modem, ASCII, FIDO, BBS)

Orr, Brian. "DECnet Rainbow: Part II", THE DEC MICROLETTER, Vol. 1, No.3 (n.d.[1987]), 18-23, 28. (DECnet-Rainbow, Networking, NDU, Printers, Printing, VAX, Laser, FAL, TFA, TNT, Task-to-Task, Communications, Transparent-to-Task, TTT)

"PC to PC Transfer Using KERMIT", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 3, No. 2 (Feb. 1986), 20. (Communications, Modem)

"Q/A: Poly-XFR", PERSPECTIVE, Vol. 2, No. 1 (January 1984), 17-18. (Communications)

"Questions and Answers", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 5 (May, 1985), 9-10. (Diskdrives, Communications, WordStar, Multiplan, ASCII, Terminal, Lotus, CP/M-86, Graphics, ReGIS)

Reisler, Kurt. "Product Update: New Firmware Upgrade for US Robotics Courier 2400", WASHINGTON

AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 3, No. 3 (March 1986), 7. (Communications, Modem)

Scott, Larry D. "Going OnLine: Remote Operation of the DEC Rainbow Using MS-DOS", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 3 No. 10 (Oct, 1986),24-26. (Communications, Modem)

"Software and Hardware", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 6 (June, 1985), 15. (Communications, Graphics, Modem, Autocad, Knowledgeman, Dr. Logo, Utilities, CP/M, Hardware)

"Software Available for the DEC Rainbow", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 1, No. 3 (July, 1984), 5. Investment Manager, Calc, Graphwriter, Communications, WordEase, Programming, C)

Tugman, Tom. "FIDO Bulletin Board Commands", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 1, No. 7 (Nov., 1984), 10. (BBS, Communications)

Tugman, Tom. "Going Online: How to Use FIDO", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 3, Nos. 7-8 (July-Sept. 1986), 24-26. (Communications, Bulletin Boards)

Weinberg, Paul. "Transparent Integration between PCs and VAXs", HARDCOPY, Vol. 7, No. 7 (July 1987), 116-120. (Communications, Networking,

Winston, Mark. "Some LEssons Learned During my First Big dBaseII Project", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 1, No. 4-5 (August-Sept., 1984), 2-3. (Database Management, Communications, Modem, Bibliography)

Wszolek, Donald. "Proliferation of PCs brings New Concepts in Processing to the Corporate Envirnoment", HARDCOPY, Vol.7, No. 6 (June 1987), 85-87. (Networking, Communications)

COMPATIBILITY
Bassin, N. Jay. "Software Review: MediaMaster", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 1, No. 3 (July, 1984), 1-2. (Compatibility, WordStar, CP/M, MS-DOS, Formatting)

"Big Plans for DEC Rainbow", INFOWORLD, January 14, 1985, p. 17. (Hardware, Compatibility, IBM)

Chattaway, Alan. "Rainbow--IBM Compatibility Revisited: Making Your IBM PC Compatible with the Rainbow", RAINBOW NEWS, Vol. 4, No. 10-12 (Oct.-Dec., 1987), 22. (Uniform-PC, Software)

"Compatibility", RAINBOW NEWS, Vol. 4 No. 10-12 (Oct.-Dec., 1987), 25. (CodeBlue, RB-LINK)

Crayne, William. "Much of What You wanted to Know about the Rainbow . . . and Have Been Asking", RAINBOWNEWS, Vol. 4, No. 5-6 (May-June, 1987), 23-24. (Code Blue, IBM, Compatibility, Formatting)

Crayne, William. "Much of What You Wanted to Know About the Rainbow. . . And Have Been Asking--Part II", RAINBOW NEWS, Vol. 4, No. 7-9 (July-Sept., 1987), 24-29. (Compatibility, Programming, MS-DOS)

Deahl, Thomas F. "Guest Editorial: Tapping the Motherlode of PC Software", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 1 (January, 1985), 10. (Compatibility, IBM)

Fitzgerald, Dennis K. "Book/Program Review: Pascal Programs for Data Base Management", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2 No. 8-9 (August-Sept., 1985), 9. (Programming, Languages, Bibliography, Database Management, Compatibility)

"The Grapevine", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 10 (Oct., 1985), 17. (MediaMaster, Compatibility, Memory, Networking, Lotus)

Graves, Thomas E. "Software Review: Code Blue, Version 2.0", RAINBOW NEWS, Vol. 4, No. 10-12 (Oct.-

Dec., 1987), 17-21. (CodeBlue, Compatibility)

Miller, Jerry and Julie Starr. "Project Transport", THE DEC MICROLETTER, Vol. 1, No. 1 (Nov.1986), 6, 15. (IBM, Compatibility, Software, Codeblue, CB, PC-Outline)

"New Rainbow is Months Away", INFOWORLD, January 21, 1985, pp. 20-21. (Hardware, Compatibility, IBM)

"Product Announcement: Transfer Between the Rainbow and the IBM-PC", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 2 (February, 1985), 4-5. (Compatibility, IBM, Disks, CP/M-86, PC-DOS)

"Rainbow Rumors", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 1 (January, 1985), 13. (IBM, Compatibility, Hardware)

"ROBIN (VT 180) Notes", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 7 (July, 1985), 12. (Disks, Compatibility, CP/M)

COMPILER
Camas, Tony. "The I/O Port", THE DEC MICROLETTER, Vol. 1, No. 2 (January/February, 1987), 11-12, 26. (CP/M, IBM, Harddisk, UNIX, C, MS-DOS, Compiler,VENIX86, RB-Link, LA100, Printers)

"The Grapevine", PC-SIG NEWSLETTER, Vol. 2, No. 4 (June, 1985), 57. (MS-DOS, Compiler, C, Printers, RBase, DECminiSidekick)

"The Grapevine", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 5 (May, 1985), 10. (Compiler, C, Printers, RBase, dBaseIII, MS-DOSv2.11, MiniTel, Pascal)

Olson, Paul. "Software Review: Dr. Pascal", RAINBOW nEWS, Vol. 4, No. 7-9 (July-Sept., 1987), 11-14. (Language, Compiler, Editors)

Shepherd, F. "Comments on Programs Which Have Worked with the Rainbow", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 4 (April, 1985), 8-9. (Fortran, Compiler, Terminal Emulation, LA50)

"Software and Hardware", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 1 (January, 1985), 14. (ChessWright, BASIC, Compiler, Graphics, Database)

COMPOSE
Bowman, Carl. "Software Review: Benchmark Composer", RAINBOW NEWS, Vol, 4, No. 5-6 (May/June 1987), 16. (Wordprocessing, Graphics)

Camas, Tony. "The I/O Port", THE DEC MICROLETTER, Vol. 1, No. 3 (n.d. [1987]), 24-26, [36]. (Character Sets, COMPOSE, LA100, LQP02, LQP03, Concurrent CP/M, Select, CONDOR, POKE, Video Memory, MDRIVE, MBASIC)

Myers, W.H. "Special Characters amd the use of the Compose Character Key in WordPerfect Using the LA-50 Printer", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 3, No. 10 (Oct. 1986), 18-20. (Printers, Wordprocessing)

COMPUSERVE
Byrd, Brent A. "Letter", RAINBOW NEWS, Vol. 4, No. 3-4 (March/April, 1987), 42-43. (Hack, FreeStyle, Compuserve)

Graham, Chad. "Communications: Electronic Mail", RAINBOW NEWS, Vol. 4, No. 5-6 (May/June 1987), 21-22. (CompuServe, The Source, EasyPlex, FIDO)

Graves, Thomas E. "Communications Tutorial: Navigating the Compuserve Information Service", RAINBOW NEWS, Vol. 4, No. 3-4 (March/April, 1987), 22-29. (BBS)

Leeman, Bill. "Going Online: Compuserve", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 3, Nos. 7-8 (Jul-Sept. 1986), 23. (Communications, Databases)

Needleman, Theodore. "Rainbow Corner", HARDCOPY, Vol. 6, No. 9 (September 1986), 88. (CodeBlue, Bulletinboards, FIDO, Public Domain Software, Jackson, Compuserve, Formatting, WUTIL)

CONCURRENT CP/M
Camas, Tony. "The I/O Port", THE DEC MICROLETTER, Vol. 1, No. 3 (n.d. [1987]), 24-26, [36]. (Character Sets, COMPOSE, LA100, LQP02, LQP03, Concurrent CP/M, Select, CONDOR, POKE, Video Memory, MDRIVE, MBASIC)

Kempton, Willett. "Software Update: Whatever Happened to Concurrent CP/M", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 8-9 (August-Sept., 1985), 7-9. (CP/M-86, MS-DOS)

Needleman, Ted. "Multitasking for the Rainbow", HARDCOPY, Vol. 14, No. 12 (December 1985), 57-60. (CP/M, Concurrent CP/M, Multi-Tasking)

CONDOR
Camas, Tony. "The I/O Port", THE DEC MICROLETTER, Vol. 1, No. 3 (n.d. [1987]), 24-26, [36]. (Character Sets, COMPOSE, LA100, LQP02, LQP03, Concurrent CP/M, Select, CONDOR, POKE, Video Memory, MDRIVE, MBASIC)

Garbera, Alexander. "Condor to Samna Transfers", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 11 (Nov. 1985), 12. (Spreadsheet, Wordprocessors, Mailmerge)

Rhodes, Robert, P. "Software Review: Is Condor an Endangered Species?" RAINBOW NEWS, Vol. 4, No. 3-4 (March/April, 1987), 16-17. (Bibliography, Database Management)

Rhodes, Bob. "Software Review: Is Condor an Endangered Species?:2" RAINBOW NEWS, Vol. 4, No. 10-12 (Oct.-Dec., 1987), 16. (Database, dBase, Graphics)

CONOFONTS
Jackson, Bruce. "Desk Top Publishing on a Rainbow--It can Be Done! Bookmaking", RAINBOW NEWS, Vol. 4, No. 3-4 (March/April, 1987), 3-5. (Desktop Publishing, Rainbow Freeware, WordPerfect, IBM,Typesetting, LA50, Typefaces, Laser, Conofonts)

CONTEXT MBA
"Application Software: Context MBA: An Integrated Softwear Package for the Rainbow 100", PERSPECTIVE, Vol. 2, No. 3 (October 1984), 32-33. (Spreadsheet, Graphics, Wordprocessing, Data Management, Communications)

CONTROLLER
"New Product Announcement: Rainbow Hard Disk Partioning Program", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 6, No. 3 (May, 1986), 7. (Hardware, Controller)

Olson, Paul. "Hardware Review: The Microchs Hard Disk Controller and the Seagate ST-255 20MB Hard Drive", RAINBOW NEWS, Vol. 4, No. 10-12 (Oct.-Dec., 1987), 5-6. (Controller, Formatting)

Roberts, William Lee. "The Complete Guide to Hard Disk Options for the DEC Rainbow", THE DEC MICROLETTER, Vol. 1, No. 2 (January/February, 1987), 18-22. (Harddisk, RD50, Controller, Rainbo Works, Backup, RD31, Seagate, Utilities, MS-DOS, WUTIL, Installing, 100A, 100B, Formatting, Initialization, COPY, PIP, CP/M, Backup, Field Service)

Wheeler, John E. "The Rainbow 100+2: Installing the MicroCHS Hard Disk Controller with Two Disk Drives", RAINBOW NEWS, Vol. 4, No. 10-12 (Oct.-Dec., 1987), 7-8. (Harddisk, Formatting, WUTIL)

White, Joe. "Notes on Installing a 20MB Hard Disk with an Internal I-Drive and MicroCHS Dual Controller", RAINBOW NEWS, Vol. 4, No. 10-12 (Oct.-Dec., 1987), 8-9. (Harddisk, Install)

COPY
"December Meeting Notes", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 1 (January, 1985), 1-2. (MS-DOS, Formatting, Copy, Directories, Paths, Batch, Bibliography)

Roberts, William Lee. "The Complete Guide to Hard Disk Options for the DEC Rainbow", THE DEC MICROLETTER, Vol. 1, No. 2 (January/February, 1987), 18-22. (Harddisk, RD50, Controller, Rainbo Works, Backup, RD31, Seagate, Utilities, MS-DOS, WUTIL, Installing, 100A, 100B, Formatting, Initialization, COPY, PIP, CP/M, Backup, Field Service)

COVER
Stewart, Steven. "The DOS Side: Directory Utilities", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 1, No. 6 (Oct., 1984), 4-5. (MS-DOS, Public Domain Software, Cover, FDIR, SDIR, TD, TREEDIR, WHEREIS, Z)

Stewart, Steven. "The DOS Side: Five Utilities", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 1, No. 7 (Nov., 1984), 3-4. (MS-DOS, Public Domain Software, Cover, Z, TD, Delete, VDL, WASH, SWEEP)

Stewart, Steven. "The Fine Points of MS-DOS: Public Domain Directory and File Maintenance Utilities for MS-DOS", PC-SIG NEWSLETTER, Vol. 2, No. 4 (June, 1985), 27-30. (Cover, FDIR, SDIR, TD, TREEDIR.COM, WHEREIS, Z, Delete, VDL, WASH, CSWEEP)

CP/M
Anthony, Piers. "Wielding a Rainbow", RAINBOW NEWS, Vol. 4 No. 5-6 (May-June 1987), 8-11. (Wordprocessing, SmartKey, Hardware, Software, CP/M, MS-DOS)

Barfield, Ed. "Technical Perspectives: Inside the Rainbow . . . A Conversation Between Chips", PERSPECTIVE, Vol. 2, No. 2 (May 1984), 13-14. (CP/M, BIOS, Programming)

Bassin, Jay. "The Chairman's Corner", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 10 (Oct., 1985), 3. (Crosstalk, AME86.exe, WordPerfect, MS-DOS, CP/M, Trump, Hardware)

Bassin, N. Jay. "Guest Editorial: 'DEC Dares to be Different,'" WASHINGTON AREA RAINBW USERS GROUP NEWSLETTER, Vol. 2, No. 1 (January, 1985), 9-10. (MS-DOS, CP/M, Lotus, Service, Mouse)

Bassin, N. Jay. "Screen Controls from CP/M Operating System", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 3, (March, 1985), 13. (Graphics, Printers)

Bassin, N. Jay. "Software Review: MediaMaster", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 1, No. 3 (July, 1984), 1-2. (Compatibility, WordStar, CP/M, MS-DOS, Formatting)

Bassin, N. Jay. "Software Review: Media Master", PC-SIG NEWSLETTER, Vol. 2, No. 4 (June, 1985), 25-26. (Utilities, Formatting, Disks, CP/M, MS-DOS, IBM)

Bassin, N. Jay. "Using CP/M's 'PIP' to Concatenate Files", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 1 (January, 1985), 10-11. (ASCII)

Bassin, N. Jay. "Viewing Wordprocessor Files in CP/M", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 1 No. [8] (Dec., 1984), 6. (Wordprocessing, WordStar)

Bunn, W. "Extending the CP/M Directory Size", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 3, Nos. 7-8 (July-Sept., 1986), 28.

Camas, Tony. "Right on Track: A Floppy Disk Primer", THE DEC MICROLETTER, Vol. 1, No. 3 (n.d.[1987]), 7-14. (MS-DOS, CP/M)

Camas, Tony. "The I/O Port", THE DEC MICROLETTER, Vol. 1, No. 3 (n.d. [1987]), 24-26, [36]. (Character Sets, COMPOSE, LA100, LQP02, LQP03, Concurrent CP/M, Select, CONDOR, POKE, Video Memory, MDRIVE, MBASIC)

Camas, Tony. "The I/O Port", THE DEC MICROLETTER, Vol. 1, No. 2 (January/February, 1987), 11-12, 26. (CP/M, IBM, Harddisk, UNIX, C, MS-DOS, Compiler,VENIX86, RB-Link, LA100, Printers)

"Case of the Missing Technical Documentation, The" WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 1, No. 4-5 (August-Sept., 1984), 5-6. (Manuals, Programming, MS-DOS, CP/M, Bibliogrphy, Communication)

Chattaway, Alan. "Notes from Down Under", RAINBOW NEWS, Vol. 4, No. 3-4 (March/April, 1987), 40-41. (ASCII, WordStar, CP/M, ENSOFT, TEXTSOFT, PrintMAster, PrintShop, CP/M-80, Graphics)

"DEC Announces 20 Meg Hard Disk for Rainbow", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 10 (Oct., 1985), 5. (Hardware, Harddisk, CP/M, MS-DOS)

Ference, Jack. "POLY/XFR MS-DOS Compatible Upgrade", PC-SIG NEWSLETTER, Vol. 2, No. 4 (June, 1985), 26. (Communications, CP/M)

Fineman, Mark. "Software Review: Freestyle", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 3, No.5 (May 1986), 7-8. (Wordprocessing, CP/M)

Fitzgerald, Dennis K. "The Standalone Formatter--An Alternative to the Word Processor", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 1 (January, 1985), 7-9. (Formatting, UNIX, ROFF, IBM, RED, EDLIN, MS-DOS, CP/M, WordStar, NRO, Fonts, Printing)

Fitzgerald, Dennis. "The Standalone Formatter--An Alternativ to the Word Procesor", PC-SIG NEWSLETTER, Vol. 2, No. 4 (June, 1985), 18-23. (Formatting, Wordprocessor, WordPerfect, WordStar, CP/M, RED, MS-DOS, EDLIN, Pascal, Editors, NRO, ROFF, UNIX)

Franswick, Rus. "CP/M Product Review: Out-Think--Outlining on the Rainbow", RAINBOW NEWS, Vol. 4, No. 5-6 (May-June, 1987), 25-27. (Software, Wordprocessing)

Garbera, Barbara. "Rainbow to DECmate and Back Using POLYCOM", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 7 (July, 1985), 10. (Communications, CP/M)

Geremia, Ken. "Software Review: In Search of a Sharper `Image'", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 8-9 (August-Sept., 1985), 5-6. (CP/M, Wordprocessing, Graphics)

Glixon, Roy. "How to Recover Accidentally Deleted Files with DU", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 1, No. 4-5 (August-Sept., 1984), 3-4. (CP/M, Utilities)

Glixon, Roy. "How to Recover Accidentally Deleted CP/M Files with DU", PC-SIG NEWSLETTER, Vol. 2, No. 4 (June, 1985), 34-35. (Utilities, Public Domain)

Gordon, Ken. "Software Review: Personal Planner", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 10 (Oct., 1985), 6-7. (MS-DOS, CP/M, Taxes, Financial Planner)

"Grapevine, The", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 2 (February, 1985), 9. (CP/M, Public Domain Software, AME-86, C)

Graves, Thomas E. "Yes, Virginia, You Can Finally Run WordStar, Version 4, on the Rainbow in CP/M, But, . . .", RAINBOW NEWS, Vol. 4, No. 7-9 (July-Sept., 1987), 19. (Wordprocessing, Software)

Hall, Carl E. "Software Review: look Version 1.1", RAINBOW NEWS, Vol. 4, No. 10-12 (Oct.-Dec.,1987), 12-13. (Graphics, CP/M)

"Hardware and Software Available", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 1, No. 7 (Nov., 1984), 8-9. (Terminal, E-Mail, Desktop, WordStar, LA100, CP/M, ADA, Graphics, Printer, MS-DOS, Accounting, Cables)

Houseman, Carl. "Installing a 40MB Hard Disk in Your Rainbow", RAINBOW NEWS, Vol. 4, No. 10-12 (Oct.-Dec., 1987), 6-7. (Hardware, Harddisk, MS-DOS, Formatting, CP/M)

Howard, Fritz. "Duncan MacDonald: DM110 Tape Backup Subsystem", THE DEC MICROLETTER, Vol. 1 No. 2 (January/February, 1987), 29, 32. (Backup, Harddisk, MS-DOS, CP/M)

"January Meeting Notes", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 2 (February, 1985), 1-2. (Communications, PolyCom, PolyTrm, Kermit, Terminal, Emulator, ReGIS, Graphics, MS-DOS, Knowledgeman, Multiplan, CP/M, FIDO)

Joseph, Peter M. "Letter", RAINBOW NEWS, Vol. 4, No. 3-4 (March/April, 1987), 42. (CP/M, DDT, RX50, Errors)

"July Meeting Notes", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 1, No. 4-5 (August-Sept., 1984), 1-2. (CP/M, Bibliography, WordPerfect, LA50)

"June Meeting Notes", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 1, No. 3 (July, 1984), 1. (DECUS, MS-DOS, CP/M)

Kempton, Willett. "Software Update: Whatever Happened to Concurrent CP/M", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 8-9 (August-Sept., 1985), 7-9. (CP/M-86, MS-DOS)

"Letters to the Editor", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 10 (Oct., 1985), 15-16. (Memory, Genealogy, LA50, WordPerfect, WordStar, CP/M)

Mack, Caroline. "A Fool and His/Her (Choose One) Words are soon Parted, Or, How to Avoid the Agony of Delete", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 10 (Oct., 1985), 13-14. (Wordprocessing, Files, WordPerfect, CP/M, MS-DOS, LCTERM, DECMini, BAckup, DU)

McClinton, Arthur Jr. "KERMIT Communciations", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 1 (January, 1985), 11-12. (CP/M)

McClinton, Arthur Jr. "Product Review: rainbow ReGIS", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 3 (March, 1985), 7-8 (Graphics, KERMIT, POLYCOM, MS-DOS, CP/M, CP/M-86)

McSweeny, Jim. "Rainbow Graphics--GSX/RGI", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 1, No. 6 (Oct., 1984), 5-6. (CP/M, RGI)

Mandle, John. "Boot can Scoot Between Systems", RAINBOW NEWS, Vol. 4, No. 7-9 (July, Sept., 1987), 20. (Software, MS- DOS, CP/M, Booting)

"March Meeting Notes", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 4 (April, 1985), 2. (Software, Networking, MS-DOS, CP/M Harddisk, WPS-Plus, Bibliography)

Maroun, David P. "DOS as a Word Processor", RAINBOW NEWS, Vol. 4, No. 5-6 (May/June 1987), 27. (MS-DOS, Wordprocessor, CP/M)

Maroun, David P. "Notes on Programming", RAINBOW NEWS, Vol. 4, No. 7-9 (July-Sept., 1987), 29-30. (CP/M, CP/M-86)

Maroun, David. "Notes on Programming", RAINBOW NEWS, Vol. 4, No. 10-12 (Oct.-Dec., 1987), 33. (Editing, MBASIC, CP/M, Printing)

"May Meeting Notes", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 6 (June, 1985), 2-3. (BBS, FIDO, CP/M, MS-DOS, Lotus, Printing, WordPerfect, Diskdrive)

"Micro-Sources Designs Print-Screen", WASHINGTON AREA RANBOW USERS GROUP NEWSLETTER, Vol. 1, No. [8] (Dec., 1984), 12. (Printing, CP/M, MS-DOS)

Needleman, Ted. "Multitasking for the Rainbow", HARDCOPY, Vol. 14, No. 12 (December 1985), 57-60. (CP/M, Concurrent CP/M, Multi-Tasking)

Needleman, Theodore. "Rainbow Corner", HARDCOPY, Vol. 7, No. 7 (July 1987), 70. (Hardware, Diskdrives, CP/M)

"November Meeting Notes", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 1, No. [8] (Dec., 1984), 1. (Public Domain Software, CP/M, MS-DOS, Utilities, Wordprocessing, Communications, Modem, ASCII, FIDO, BBS)

Olson, Paul. "CP/Mer", RAINBOW NEWS, Vol. 4, No. 3-4 (March/April, 1987), 29-30. (CP/M, RED)

Olson, Paul. "The CP/M Attic", RAINBOW NEWS, Vol. 4, No. 5-6 (May-June, 1987), 24-25. (Boot, Replace, SystemsCP/M-86, Software, RED)

Olson, Paul. "The CP/M Attic: Disk Formats, Disk Data Blocks,a nd File Control Blocks (Part 2)", RAINBOW NEWS, Vol. 4, No. 7-9 (July-Sept., 1987), 26-29. (CP/M-86, Pascal)

Olson, Paul. "The CP/M Attic: Disk Formats, Disk Data Blocks, and File Control Blocks", RAINBOW NEWS, Vol. 4, No. 10-12 (Oct.-Dec., 1987), 30-32. (Disks, CP/M-86, Memory, RED, Editors)

Pasini, Mike. "Guest Editorial: The End of the Rainbow", RAINBOW NEWS, Vol. 4, No. 5-6 (May-June, 1987), 4-5. (Announcements, Software, CP/M, Multiplan, WordStar)

Pfaff, Judy L. "Software Review: SL-Micro-An SPSS Look-Alike", RAINBOW NEWS, Vol. 4, No. 5-6 (May-June, 1987), 113, 17. (Statistics, Statistical Software, CP/M, MS-DOS, CP/M-86)

"Product Alert", RAINBOW NEWS, Vol. 4, No. 5-6 (May-June, 1987), 32. (Hardware, Software, MS-DOSv3.1, Windows, CP/M, Pascal, 20/20, DataEase, WordStar, Open Access, Financial Planner)

"Product Announcement: Transfer Between the Rainbow and the IBM- PC", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 2 (February, 1985), 4-5. (Compatibility, IBM, Disks, CP/M-86, PC-DOS)

"Q/A: CP/M", PERSPECTIVES, Vol. 1, No. 1 (February 1983), 28-29. (Systems)

"Q/A: CP/M-86/80", PERSPECTIVE, Vol. 3, No. 1 (January 1985), 42. (CP/M)

"Q/A: MS/DOS and CP/M", PERSPECTIVE, Vol. 4, No. 2 (n.d.), 28. (MS-DOS)

"Questions and Answers", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 1, No. 2 (June, 1984), 4. (CP/M, WPS-80, LA50, WordPerfect)

"Questions and Answers", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 3 (March, 1985), 14-15. (dBaseII, MBASIC, Lotus,, CP/M, Samna, Laptop, Tandy Model 100, Bibliography)

"Rainbow Questions and Answers", PC-SIG NEWSLETTER, Vol. 2, No. 4 (June, 1985), 43-45. (Terminal, Lotus, Spreadsheets, dBaseII, CP/M-86, Multiplan, MS-DOS, CP/M, LA50, LA100, MBASIC, Samna, Graphics)

Reisler, Kurt L. "RCPM, RBBS, 'Freeware'--Public Domain Software for the Rainbow Personal Computer", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 3 (March, 1985), 6-7. (CP/M, BBS, Patch, Games, Accounting, KERMIT, XMODEM, Bibliography)

Roberts, William Lee. "The Complete Guide to Hard Disk Options for the DEC Rainbow", THE DEC MICROLETTER, Vol. 1, No. 2 (January/February, 1987), 18-22. (Harddisk, RD50, Controller, Rainbo Works, Backup, RD31, Seagate, Utilities, MS-DOS, WUTIL, Installing, 100A, 100B, Formatting, Initialization, COPY, PIP, CP/M, Backup, Field Service)

"ROBIN (VT 180) Notes", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 7 (July, 1985), 12. (Disks, Compatibility, CP/M)

"September Meeting Notes", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 10 (Oct., 1985), 4. (FIDO, ARC, Freestyle, WordStar, CP/M, MS-DOS, WordPerfect, Samna, WPS- PLUS, Sideways, SEDT)

"Software and Hardware", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 6 (June, 1985), 15. (Communications, Graphics, Modem, Autocad, Knowledgeman, Dr. Logo, Utilities, CP/M, Hardware)

"Software and Hardware", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 7 (July, 1985), 15. (Milestone, Managing, MS-DOS, CP/M)

Stewart, Steven. "KnowledgeMan--Review", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 1, No. 2 (June, 1984), 2. (Database, CP/M, MS-DOS)

"User to User: Converting Hitachi Color Monitors for the Rainbow", RAINBOW NEWS, Vol. 4, No. 7-9 (July-Sept., 1987), 44. (Hardware, Disks, CP/M)

Vince, Paul. "Exchanging CP/M Versions of WordStar and dBaseII for MS-DOS Versions", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 10 (Oct., 1985), 8-9.

Vince, Paul. "Setting Printer Features", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 7 (July, 1985), 7-8, 17-22. (Wordprocessing, WordStar, LA50, Lotus, Printers, Printing, CP/M, EDLIN, LA100)

Waite, M. CP/M BIBLE. Indianapolis: Howard Sams, 1983. (Bibliography)

Waite, M. THE SOUL OF CP/M. Indianapolis: Howard Sams, 1983. (Bibliography)

"WordStar Releases CP/M Version of 4.0", RAINBOW NEWS, Vol. 4, No. 7-9 (July-Sept., 1987), 3. (Wordprocessing)

"WPS-80 Upgrade Available", PC-SIG NEWSLETTER, Vol. 2, No. 4 (June, 1985), 26. (Wordprocessing, CP/M)

CP/M-86

"February Meeting Notes", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 2 (March, 1985), 4-5. (Languages, Programming, PAscal, C, BASIC, CP/M-86, MS-DOS, APL, ADA)

Kempton, Willett. "Software Update: Whatever Happened to Concurrent CP/M", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 8-9 (August-Sept., 1985), 7-9. (CP/M-86, MS-DOS)

McClinton, Arthur Jr. "Product Review: rainbow ReGIS", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 3 (March, 1985), 7-8 (Graphics, KERMIT, POLYCOM, MS-DOS, CP/M, CP/M-86)

Maroun, David P. "Notes on Programming", RAINBOW NEWS, Vol. 4, No. 7-9 (July-Sept., 1987), 29-30. (CP/M, CP/M-86)

Olson, Paul. "The CP/M Attic", RAINBOW NEWS, Vol. 4, No. 5-6 (May-June, 1987), 24-25. (CP/M-86, Software, RED)

Olson, Paul. "The CP/M Attic: Disk Formats, Disk Data Blocks, and File Control Blocks", RAINBOW NEWS, Vol. 4, No. 10-12 (Oct.-Dec., 1987), 30-32. (Disks, CP/M-86, Memory, RED, Editors)

Olson, Paul. "The CP/M Attic: Disk Formats, Disk Data Blocks,a nd File Control Blocks (Part 2)", RAINBOW NEWS, Vol. 4, No. 7-9 (July-Sept., 1987), 26-29. (CP/M-86, Pascal)

Pfaff, Judy L. "Software Review: SL-Micro-An SPSS Look-Alike", RAINBOW NEWS, Vol. 4, No. 5-6 (May-June, 1987), 113, 17. (Statistics, Statistical Software, CP/M, MS-DOS, CP/M-86)

"Product Announcement: Transfer Between the Rainbow and the IBM- PC", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 2 (February, 1985), 4-5. (Compatibility, IBM, Disks, CP/M-86, PC-DOS)

"Q/A: CP/M-86/80", PERSPECTIVE, Vol. 3, No. 1 (January 1985), 42. (CP/M)

"Questions and Answers", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 5 (May, 1985), 9-10. (Diskdrives, Communications, WordStar, Multiplan, ASCII, Terminal, Lotus, CP/M-86, Graphics, ReGIS)

"Rainbow Questions and Answers", PC-SIG NEWSLETTER, Vol. 2, No. 4 (June, 1985), 43-45. (Terminal, Lotus, Spreadsheets, dBaseII, CP/M-86, Multiplan, MS-DOS, CP/M, LA50, LA100, MBASIC, Samna, Graphics)

"Software and Hardware", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 2 (February, 1985), 9-10. (Printers, Taxes, CP/M-86,MS-DOS,GWBASIC, MBASIC, POLYCOM, POLY-XFR)

CROSSTALK
Bassin, Jay. "The Chairman's Corner", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 2, No. 10 (Oct., 1985), 3. (Crosstalk, AME86.exe, WordPerfect, MS-DOS, CP/M, Trump, Hardware)

"Vaporware", WASHINGTON AREA RAINBOW USERS GROUP NEWSLETTER, Vol. 1, No. 7 (Nov., 1984), 5. (Software, Hardware, Fortran, Pascal, Languages, Crosstalk)

CSWEEP
Stewart, Steven. "The Fine Points of MS-DOS: Public Domain Directory and File Maintenance Utilities for MS-DOS", PC- SIG NEWSLETTER, Vol. 2, No. 4 (June, 1985), 27-30. (Cover, FDIR, SDIR, TD, TREEDIR.COM, WHEREIS, Z, Delete, VDL, WASH, CSWEEP)

PCSA Section

More Atlanta Symposium Slides
By Don Call, Digital Equipment Corporation

DECUS

ATLANTA, GEORGIA

MAY 8 - MAY 12 1989

Don Call
PCSG Technical Suppt

NETWORK

PCSA HINTS

To Be Discussed

- NETWORK
- MEMORY
- VAX TUNING

PCSA HINTS

CHANGING VAX NODE NAME AND ADDRESS

STEPS TO TAKE:

1. RUN NETCONFIG.COM FROM SYS$MANAGER DIRECTORY

2. MODIFY SCSNODE

3. MODIFY SCSSYSTEMID

4. TURN ON SERVICE ENABLED

5. MODIFY PIPELINE QUOTA

PCSA HINTS
NET CONFIGURATION

RUN NETCONFIG.COM "@SYS$MANAGER:NETCONFIG"

ANSWER FIRST QUESTION WITH NEW NODE NAME

ANSWER SECOND QUESTION WITH NEW NODE ADDRESS

CREATE/EDIT MODPARAMS.DAT IN SYS$SYSROOT:[SYSEXE

1st LINE SHOULD READ SYSSYSTEMID=(1024*AREA+ADDRI

2nd LINE SHOULD READ SCSNODE="NCDENAME"

3rd LINE SHOULD READ VAXCLUSTER=0

RUN @SYS$UPDATE:AUTOGEN GETDATA REBOOT [FEEDBA

PCSA HINTS
KNOWN NODES

NEW WAY TO DEFINE KNOWN NODES AS PCs ARE ADDED

- UP THROUGH V2.1 KNOWN NODES WERE
  STORED IN AN ACCOUNT NAMED PCSA ADMIN
- V2.2 CHANGES THINGS
- KNOWN NODES ARE NOW STORED IN A
  TEXT FILE NAMED DECNODE.TXT
- TO ADD NEW NODES USE EDT
- EX: EDIT SYS$COMMON[PCSA]:DECNODE.TXT

PCSA HINTS
NCP SETUP

IF REMOTE BOOT WILL BE USED SERVICE MUST BE ENABLI

SERVICE ENABLED ALLOWS LOADING AND LOOPBACK TES1

DO THE FOLLOWING: MC NCP

SET CIRC QNA-0 STATE OFF

SET CIRC QNA-0 SERVICE ENABLED

SET CIRC QNA-0 STATE ON

DEF CIRC QNA-0 SERV ENABLED

PCSA HINTS
GOTCHAS

WHEN UPGRADING FROM V1.2 TO V2.2 MAKE SURE
ALL ACCOUNTS THAT YOU HAVE GRANTED PUBLIC ACCE
TO ARE REMOVED FROM THE UAF DATABASE

IF NOT MAY GET MESSAGE
"INVALID SERVICE OR PASSWORD"

FILE SERVER AND LAD MAY NOT START
AFTER UPGRADING TO V2.2
START UP FILES ARE NOW LOCATED IN SYS$STARTUP
USED TO BE SYS$MANAGER
NEVER CONNECT FILE SERVER TO A CLUSTER ALIAS
WHEN INSTALLING PCSA FOR THE FIRST TIME LEAVE
THINGS ALONE. DO NOT MOVE FILES AROUND. FILES ARE
DIRECTORY DEPENDANT

PCSA HINTS
NCP SETUP

PIPELINE QUOTA:

PIPLINE QUOTA IS USED FOR MULTIBUFFERING
AT THE NSP LEVEL

- DEFAULT IS SET AT 3000
- VERSION 2.1 RECOMMENDED 6000
- VERSION 2.2 WILL RECOMEND 15000

ENTER:

  NCP SET EXEC PIPELINE QUOTA 15000
  NCP DEF EXEC PIPELINE QUOTA 15000

PCSA HINTS
LOGICAL LINKS

- IS SET USEING NCP EXECUTOR CHARACTERISTICS
- DEFAULT VALUE IS 32
- MAXIMUM LINKS IS 960
- NUMBER OF LOGICAL LINKS DEPENDS ON
  NUMBER OF PCs TO BE CONNECTED
- CALCULATE LINKS AS FOLLOWS:
  LINKS= $3 \cdot$ (NUMBER OF PCs) +
  NUMBER OF NODES IN CLUSTER +
  NUMBER OF ADD'L LINKS FOR OTHER APPS

PCSA HINTS
LOGICAL LINKS
EXAMPLE

ASSUME IT IS NOT A CLUSTER

70 PCs WILL BE CONNECTED

5 LINKS FOR OTHER APPLICATIONS

LINKS = 2*70+1+5

LINKS = 146

PCSA HINTS
MEMORY TERMS

- UPPER MEMORY
  BLOCKS OF MEMORY FROM 640K TO 1024
  ONLY AVAILABLE USING SPEC MEM EXPAND OR
  386 MEM MANAGERS

- HIGH MEMORY
  FIRST 64K BEYOND 1 MEG
  CALLED XMS EXTENDED MEM SPEC
  DEALS WITH EXTENDED NOT EXPANDED MEMORY

MEMORY

TUNING

MEMORY MAP

ROM	FFFF
	F0000
RESERVED IN AT	E0000
OPEN	
OPEN	D0000
EGA / VGA	C8000
	C0000
CGA	
	B0000
TEXT	A0000
LOWER	* PS/2 USES THIS
640K	AREA FOR ROM
	0

PCSA HINTS
TUNING DOS MEMORY USAGE

MS-DOS SIZE:
DOS 3.2   40K
DOS 3.3   45K
DOS 4.0   55K

, CONFIG.SYS:
LASTDRIVE - SET TO ¤ OF PHYS DEVICES + ¤ FILE SERV
FILES - 8 SHOULD BE ENOUGH, BUT CHECK YOUR APP
BUFFERS - 8 SHOULD BE ENOUGH, BUT CHECK YOUR APP
SHELL - THE E:XXX SWITCH SPECS SIZE, USE MIN POSS

PCSA HINTS

Tuning Considerations

- File Server
- Disk Server
- PC workload
- Types of Network Cards

PCSA HINTS

Disk Tuning

- First type "PCSA Show disk counter/cache"
- Look at Cache hits /Rate %
- The higher the Percentage the better
- Means you are making more use of Cache

PCSA HINTS

Tuning Considerations

Be Careful - Go one step at a time

Changed parameters can affect other processes

Touch the system only if you have to

In most cases you don't have to touch it

PCSA HINTS

Disk Tuning

Effect of varying Dsik Server Cache size

Cache Size	Cache Hits/Rate %
512	28624/62
768	28589/67
1024	24791/73
1280	36798/94

Notice that by increasing the cache size by 2.5 times
we increased the cache hit ratio from 62% to 94%

PCSA HINTS

Disk Tuning

- The Disk Server uses Cache
- Modify size for better Cache usage

PCSA HINTS

Disk Tuning

To adjust Cache - modify "LAD Startup.com"

Increase Cache size for a better hit rate

Monitor NPAGEDYN

Readjust as nesessary using autogen

Keep adjusting Cache until hit rate is 70% or better

leave enough time between adjustments to gather data

PCSA HINTS

File Server Tuning

Goal is to
always have file server in HIBernation

Too many PCs can cause problems

PCSA HINTS

File Server Tuning

- The lower the page fault rate the better
- Edit "PCFS STARTUP.COM"
- Increase "/MAXIMUM WORKING SET"
  to lower page faults

PCSA HINTS

File Server Tuning

- Do a "show system"
- Determine if PCFS SERVER is in ASTWAIT
- Modify "PCFS STARTUP.COM"
- Increase /AST LIMIT
- Guideline is to make it 4X the ¤ of PCs

PCSA HINTS

File Server Tuning

- On VMS "Monitor File"
- look at "File Hdr"
- The higher the hit rate % the better
- If too low:
  Increase SYSGEN parameter ACP HDRCACHE
  This increases ¤ of file headr blocks cached

PCSA HINTS

File Server Tuning

Next step is to check the page fault rate
- Type "show system"
- Look at PCFS SERVER page fault rate

PCSA HINTS

File Server Tuning

- On VMS "Monitor File"
- Look at DIR DATA hit %
- The higher the percentage the better
- If low:
  Increase SYSGEN parameter ACP DIRCACHE

PCSA HINTS

Third Party Cards

- Disable Multicast
- Config Aide/Menu does this for you
- Run NCP set / def service disable
- Run NCP set / def multicast disable

PCSA HINTS

MEMORY USAGE

COMPONENT	VER 2.1	VER 2.2
LADDRV	2K	5K
SCH	3K	3K
DLL (DEPCA)	3K	3K
LAST	8K	8K
LAD	5K	5K
DNPETHPC	50K	49K
SESSION	24K	29K
RED!?	23K	23K
LAT	12K	12K
TOTAL	130K	137K

PCSA HINTS

Using Third Party Network Cards

- Start LAT /M:disabled
- Start session /M:d
- Start LAD /M:d /R:1
- NCP set pipeline quota 1

PCSA HINTS
EXTENDED vs EXPANDED MEMORY

EXTENDED:
- MEMORY ABOVE ONE MEGABYTE
- ADDRESSED BY 286 & 386 IN PROTECTED MODE
- ACCESSED BY SOME DRIVERS USING EXT TO INT 15h
- 286 MODE SWITCH IS TIME CONSUMING

EXPANDED:
- CONFORMS TO TO LIM SPEC
- ALLOWS FOR UP TO 32 MEG TO BE ACCESSED THROUG A PAGE FRAME WITHIN THE 1 MEG AREA
- APPS CAN NEVER ACCESS MORE THAN 64K AT A TIME
- NEEDS EMM DRIVER WHICH USES SOME OF LOWER 640K

PCSA HINTS

General Tuning Considerations

- Use LAD drives for system / applicat software
- Put file server drives at end of path
- Distribute user services amongst available systems using rating etc
- Spread services over multiple VAX disks

VAX/VMS Services for MS-DOS

Summary — Versions 2.2 New Features

VMS Version 5.1 support

VMS NETBIOS support

VMS Access to LAD

Floppy remote boot capability

Ability to set qualifiers on VMS print commands

Ability to delete print queue from PC

Ability to limit repeat log-ins from PC

Broadcast send

System administration menus

Layered software = PCLAN/Server software

DECnet/PCSA Client Version 2.2

Summary — New Features

PCMail, SEDT

Mail notification at boot time

New version of terminal emulator (VT320)

More PCs configurations supported

MS-Windows/286 (Version 2.1)

PC DECwindows

PC DOS Version 4.0 support

Floppy remote boot support

MS-Windows LJ250 printer support

Ease of installation/configuration

DECnet/PCSA Client Version 2.2

PC DECwindows

MS-DOS application which implements an X server that uses the industry-standard X Window Systems Version 11 (X11) protocol

Allows user to execute DECwindows application running on VAX/VMS or VAX/ULTRIX system to display on a PC

Initial release allows software developers and end users to evaluate, develop, and test DECwindows applications

DECnet/PCSA Client Version 2.2

PC DECwindows Requirements

VMS Version 5.1 or ULTRIX Worksystem Software Version 2.0

80286 or 80386

512K memory free memory recommended before starting applications

RT-11

AN IDEA WHOSE real TIME HAS COME

RT-11 MINITASKER
October, 1989

Contents

****

From the Editor: (a real editorial this time)

Most of us are really looking forward to Version 5.5. We've had previews of some of the new features - extended unit numbers, Unibus mapping register support, wildcard utilities in SYSLIB, new features in BUP and KED, etc. These enhancements do not come without cost, however. There are some new things happening that I think show some unfortunate trends. As **RT-11** approaches "stabalization" I hate to see features creep in that I feel are detrimental. I'll mention a couple here, and encourage readers to send notes to the Newsletter describing things you want to see in **RT-11** and things you particularly do not want to see in **RT-11** before we approach "the end of the world."

Supporting the Unibus mapping registers (UMRs) is a really tough problem, and I appreciate some of the headaches the developers had to suffer. But as a result of including UMR support, whenever any job issues an .ABTIO, <u>ALL</u> handlers in memory will be entered at their abort entry point regardless of of the state of the ABTIO$ or HNDLR$ bits in their status word (The names of these bits no longer correlate to their function.) and regardless of whether the offending job has I/O outstanding on the device. This adds a lot of system overhead, especially for internally-queues handlers that must uselessly scan their queues to see if the I/O abort is intended for them. What's worse, some handlers (certainly not mine!) may raise their priority to block interrupts during their abort-I/O code which can have serious effects. I really think that this behavior should be restricted to systems that have UMRs either as a SYSGEN option or with smarter abort-I/O code in the monitor. (Frankly, I prefer the former.)

There have been a lot of changes in block zero of **RT-11** device handlers. We've had plenty of warning about these, and most of them have been "backward compatible" so they didn't bother us if we didn't try to use the new features, or get in their way. With Version 5.5, the utilities, particularly DUP, take "advantage" of some of these new features. So for those of you who have handlers with stuff in block zero, particularly between 0 and 200, be prepared to make changes. This is fair, and I have no beef about it. But there are a lot of third-party handlers in use out there for which the users do not have sources. Worse, the manufacturers of such handlers may no longer support them, and have no intention of updating them. Just be aware that there may be a problem.

There is another change, however, that will cause a lot of problems. The Software Support Manual says that a handler should return a hard error immediately if it receives a SPFUN request that it doesn't recognize. This was modified in subsequent Release Notes to read that handlers should ignore invalid SPFUNs. With Version 5.5, handlers, particularly disk handlers, MUST ignore unrecognized SPFUNs. It seem that DUP will issue some SPFUNs, (e.g. some used by handlers that support bad-block replacement) without first checking whether the device can accept them. If the handler returns an error, according to the V5.2 policy, tough luck! Again, this is only an inconvenience to those of us who have sources to our handlers, but will be a major problem for users of third-party handlers that can't be easily modified.

So, the bottom line, is be aware of possible problems in moving up to Version 5.5. If you do encounter problems, please in addition to sending in your SPR, write me a note for the Newsletter so others can be warned and so that your fellows can help with solutions and workarounds.

$\rightarrow$ flame off $\leftarrow$

Now what's in this month's Minitasker? Ian Hammond is alive and well! He provides an informative (and entertaining) article about how he circumvented some of RT-11's shortcomings when talking to VAXen. Nick Bourgeois has useful hints and tools for those of us still using FORTRAN IV. The new features of KED are shown in the vugraphs from Megan Gentry's talk at the Atlanta symposium. Megan also provided the patch to modify USR's file-fitting algorithm. And I always like to get letters as well as full articles. Thanks to Bob Meister and Billy Youdelman for writing in.

Keep sending good stuff to me:

> John M. Crowell
> RT-11 Newsletter Editor
> P.O. Box 128
> Davis, CA 95616

John M. Crowell
RT-11 Newsletter Editor
P.O. Box 128

Davis, CA 95616
USA

August 10, 1989

Dear John

The two technical notes in the last mini-tasker finally stirred me into
writing an article about our networks and other things.

I have spent the last three weeks rewriting the HELP text file for our next
release of SHAREplus (it started as a simple cleanup). This article was more
or less a reaction to finally getting the damn thing finished (along with a
new HELP compiler and HELP utility). I guess the HELP project finally got me
back into a lost habit of writing.

After ten years of active service for DECUS Europe I have cut back to being
an committee member of the European RTSIG and a more active member for the
DECUS München RTSIG. Hopefully this will give more leisure for an occasional
submission to the Mini-Tasker.

I chaired the European RTSIG last year. We merged with the local version of
the DAARC SIG at the European level. A lot of our emphasis has moved from
RT-11 to 'realtime' in general (i.e. VAXs). We find that we are still
serving much the same DECUS audience (albiet a larger one). The multi-user
RT-11 component has almost entired dissappeared at the European level (in
fact, that occurred about three years ago). But, in general the realtime and
the RT world are in good shape over here.

At present I am working on a MACRO compiler that outputs C code.

Say hello to Milton for me.

All the best

Ian Hammond

RT-11 CONNECTIONS

Ian Hammond
HAMMONDsoftware
Stiegbreite 1
D-3400 Göttingen
West Germany

A recent issue of the mini-tasker had a couple of short notes about our technique of forwarding logical names across a network under RT-11. This article looks at problems and solutions in general for RT-11 networks.

We have accumulated these techniques over the past ten years or so in six projects:

1. STAR-eleven - an RT-11 cluster
2. VAMP - an RT-11 to VAX/VMS single-line connection
3. SHAREnet - a multi-user RT-11 version of (2) for SHAREplus.
4. VRT - A combination of 2 and 3 as RT-11 emulator under VAX/VMS
5. TPnet - A multi-node Ethernet network connecting all the above
6. A (read-only) RSX F11A ACP for SHAREplus

The Eight Unit Problem

VAMP was designed to connect RT machines to VAX's over a 'VX:' handler. The first idea was that units VX0: thru VX7: would associate with eight VMS directories. However, we already had STAR-eleven sites with over 20 nodes. Eight directories was not enough. This was the eight-unit problem.

But we had other problems too. It is a good idea for a network server to know which job of which node has a particular file open. This helps at clean-up time when an image terminates and stops wires from getting crossed when two jobs want the same thing at the same time.

Under RT-11 this was no real problem. The VX: handler gets the job number from the queue element. Under multi-user RT-11 there are other techniques. But, the STAR-eleven host had to pass node numbers to the VX: handler. The question was: Where?

We avoid special functions in handlers. They tend to break in multi-user systems and in new versions. Thus, we encode everything in standard I/O requests.

Well, we knew that RT-11 passed the VX: handler a pointer to the file name specification in the program. So, we tacked (hacked) some magic numbers and the node information on the end of the file spec. And during debug we noted that we also had a copy of the logical name available in the device handler. The lights went on in our little brains and we had solved the eight-unit problem. Out of a hack came an elegant solution.

At the local node we make an assignment like this:

```
. assign vx dat
```

And at the remote (VMS or other) system we have something like:

```
$ assign du:[data] dat
```

The VX: handler picks up the logical name DAT: and passes it to the VMS server. The VMS server translates the name locally to find the target directory.

The technique gives us access to thousands of directories and means we do not need to use crude names like VX0: etc. Later on it meant that we could use the unit numbers for a different purpose.

Penultimate Logical Names

Our systems support recursive logical names. In the example below TST: translates to DAT: which translates to VX:

```
. assign vx dat
. assign dat tst
```

This created a problem. Did a reference to TST:MYDATA.DAT mean the directory TST: or DAT:? We decided to use the last logical name in the translation path - which we call the penultimate logical name. In other words, we use the logical name that is directly assigned to the device name VX: - DAT: in the example.

Our monitors automatically translate logical names and pass the VX: handler (or other such) the penultimate logical name. The replacements for PIP and DIR perform this work internally to work out which directory to search.

ACP Transparency

The basic idea behind these networks is transparency. A user or program should not be concerned about the difference between local devices and remote directories. They shouldn't need to know where the directory is.

On the whole RT-11 helped us more than it hindered us. But some of the problems were difficult to solve.

RT-11 defines two kinds of file structure. The first is RT11A, which RT-11 knows about and handles internally. The second type is called special and includes everything else (we call it RT11S, where 'S' stands for 'Stranger').

RT11S was initially developed to handle magtape (and later cassettes). The I/O path for RT11S is slightly different to RT11A - and these slight differences caused the problems. An overview of the pre-processing and post- processing paths describes these.

Pre-processing

- Determine the issuing job number
- Work out if process recently aborted for inits
- For STAR, get the node and its job number
- Get the I/O function (including device size function)
- Filter and handle local ACP requests
  These setup and report the internal node name & address table
- Normalize rename by forcing both to same directory
- Get optional file format setup information from Q$BLKN
  This is the .ENTER or .LOOKUP file sequence number
- Get the logical name via Q$BUFF/Q$PAR
  For RT-11/XM decrement the page number and add 64 to the address.
- For .ENTER get file allocation from Q$WCNT (RT-11 convention)
- For .CLOSE get final file size from C.USED (KED truncates files)
  Treat C.USED=0 as a .PURGE - a hack

Post processing

- Copy network file id to C.SBLK
- Send back file size in C.LENG for .LOOKUP and .ENTER
  RT-11 does not setup C.LENG for RT11S channels
  Convert file size 0 to -1 for some historic reason
- Send back file size in Q$WCNT (RT-11 convention)
- If requested send back .LOOKUP file format information in C.USED
- Clear C.USED for .ENTER requests
- Clear the RT-11 tentative file flag in the CSW
  This makes it possible to use .SAVESTATUS/.RESTORE on the channel

RT-5

- Send back device size for special function 373
- Send back ACP status in monitor SPUSR variable

With this approach only one real problem remains: .READ and .WRITE requests do not properly report truncated transfer counts. For some reason this has never been a real problem.

Another issue concerns job aborts. This is a deep-fudge area of RT-11 that I suspect no one single individual has ever completely understood for more a than a few painful days of their lives. But, the basic task here is to register each job abort as it happens and pass this information to the remote server on the next request.

A special issue here is that aborts should purge all channels for a job. All channels? No, a REENTER or START command could need the image channel for overlays. But, this is a problem for the server.

Job aborts are very important for virtual terminal connections: otherwise the target system can stack up unterminated subprocesses.

Server Transparency

The server has perform a lot more work in maintaining transparency. But, happily, the server has more space to get its work done. One of the real problems for the RT-11 networks is the three-rule system requirement:

- It should occupy no space
- It should take no time
- It should cost nothing

Any RT-11 handler that takes up more than a thousand words will break about fifty thousand FORTRAN applications (a handler that occupies just one word will break about a thousand such).

Most of the server problems are related to mapping RT-11 requests on to the native file structure. This problem barely exists if the host is an RT-11 based machine. VMS is another story.

The temptation with VMS is to try an perform file format conversions in the server. We gave up on that one. Operating systems are murky enough. We settled for straight image-mode access with only two kinds of recognised files: Stream and fixed-length 512-byte records.

The default mode is stream (i.e. straight ASCII with embedded returns and line-feeds). The servers main task is to work out two little RSX/VMS variables called first-free-byte and end-block. It is actually non-trivial.

The second task under VMS is to handle RT-11's concept of tentative file. Under VMS a file name becomes visible when you .ENTER it. Under RT-11 the file name becomes visible after the .CLOSE. Some RT-11 programs rely on this behaviour.

In general the RT-11 technique is better. One problem with RSX/VMS is that a program abort can leave a half-created file lying around. In fact we use the tentative file approach now in our VMS software.

But, the biggest problem for the VMS server is RT-11's cavalier approach to closing, purging, or not closing or not purging files. Or closing them twice. Add .SAVESTATUS and .REOPEN (the GOTO's of the file space) to this problem.

The solutions took some time to develop. To summarise, we rarely believe RT- 11 when it closes (or purge's) or doesn't close (or doesn't purge) a .LOOKUP file. Instead we maintain a cache of open files, just in case an RT-11 program decides it still really needs the file after all. The same approach is used to solve .SAVESTATUS and .REOPEN (which I sincerely wish had never been invented - and if Early RT had not had to run on slow DECtapes they never would have been invented).

Generally speaking it is not desirable to give an RT-11 program non-file- structured access to a VMS disk. VMS is quite sensitive to minor changes of its disk structure (he said). However, many RT-11 programs

routinely perform .LOOKUPs to devices simply to get some device information. We let these through and only clobber them when they actually try to read or write the disk (unless the disk is mounted /FOREIGN).

Utilities
The techniques described above solve all the problems for most of the programs. However, some programs like to break RT-11 rules. Indeed, an RT-11 rule is not a rule until it has an exception (DECUS rules go the other way around I think).

Most of the programs that break the rules come from a group of developers who produce utilities such as KED, SIPP et al. There are really only two problems.

The first is that RT11S (the 'special' file structure) was initially developed to support magtapes. KED and SIPP simply do not believe that it is useful to edit or patch modules on magtapes. One PASCAL runtime has a similiar check for what it believes to be magtape. Finally, the RT-11 monitors refuse to RUN a program from magtape.

For RT-11 systems we solve this problem with a TW: handler and a TWIST utility. TWIST marks an image with a magic number. The TW: handler intercepts .DSTATUS requests and simply lies to twisted programs. When the program (e.g. KED) asks: what kind of device is VX: or TP:? TW: replies: it is a disk. KED is happy. TW: is happy. Everyone is happy. TW: automatically fixes the KMON RUN problem on the fly.

The other problem remains unsolved. IND manipulates the device name of control files in wierd and wonderful ways that cannot be solved with current techniques. IND control files must be local. However, ordinary DCL command files can be remote.

An unsolved problem for RT-11 is reassigning SY: to a remote directory. RT-11 expects to perform some non-file structured operations to its system disk. Further, it does not like any ASSIGN commands directed at SY:. This is not a real problem, which is why we have not worked hard at a real solution. Most RT-11 systems have at least a small hard disk available or can boot out of VM:.

On our own systems we can reassign SY: to a remote directory. A few years ago we had two simultaneous disk crashs and were left with floppies-only for our PDP-11 systems (yes, that was the year that we actually had a real summer here and it had nothing to do with smoking which I believe most disks actually enjoy - particularly the floppies). We booted from the floppies and assigned SY: to a VMS directory. We worked that way for about half a year (the maintenance company went under) without noting much difference.

(For in-house development it makes sense to use a single master system directory (on a PDP-11 however) so that we do not have to run around updating directories all the time. But, we also use directory search paths to help out here. Directory search paths would really help on RT-11 because they would let us selectively place a small number of things on VM:).

VIP & VIR
Our transparency does not extend to directory structures. PIP and DIR do not understand VAX/VMS directories. We did consider converting VMS directories to RT-11 directories on the fly. For about five seconds.

Instead, we rewrote PIP and DIR as VIP and VIR. These are the only two mandatory replacement parts for the networks. VIP and VIR support many extensions for the networks (and other things). After about five years development and redevelopment these utilities are basically plug-compatible with PIP and DIR.

SRCCOM and BINCOM also support wildcards. We have not done anything about those (the project is still lying around some where 'almost finished'). VAX/VMS does not help here: it does not support wildcard DIFFER operations. Explicit DIFFER filenames work fine across the connection.

File Conversion
VIP handles some automatic file conversions (for text files going to printers or terminals), and semi-automatic (/ASCII and /BINARY), and explicit conversions.

RT-7

95% of work on the networks concerns files that belong to the PDP-11 systems. Anything that they create they can read without problems. RT-11 uses Stream ASCII which VMS editors understand quite happily. Fixed-length 512-byte record files handle most of the remaining 5%. The users who need the 1% left over have to write some kind of conversion program at the VAX end. This is better than having us work out dense algorithms for conversion that would affect the masses.

VIP understands VAX/VMS variable length, VFC (and some other one whose name I forget at the moment). COPY/ASCII or a transfer with a target of LP:, LS: or TT: performs automatic conversion of these files. COPY/BINARY creates fixed-length binary files at the VAX. Finally, there are options for performing explicit file conversions.

The important point about these file conversion techniques is that they are rarely required. The whole point of these networks is to get away from thinking about networks as something different to local devices. A lot of people implicitly conjure up a picture of KERMIT or TRANSF when they think about RT-11 networks: they tend to guess that most of the network operations take place with COPY commands in a 'special file transfer utility' (and a slow one). I would guess that typically 5% to 10% of network traffic with these systems takes place with COPY commands. Most of it happens in editors, compilers and applications who can directly access remote directories.

Another point that needs to be made is that on VAX/VMS the files are stored in ordinary VAX/VMS directories: not in files that look like RT-11 logical disks. This means that the files are directly available to VAX/VMS applications.

A final point about file conversion concerns VAX/VMS EXCHANGE. When EXCHANGE copies ASCII files from RT-11 media to a VAX/VMS directory it always creates variable-length records; it refuses to create stream ASCII files. This is a drag because the VMS CONVERT utility still gets the conversion to stream wrong (it adds a newline after form-feed and has done ever VAX-time began). Now, the VIP utility can solve this problem with COPY/ASCII but it is a painful procedure. More recently we added some functionality to let us use a PDP-11 utility to read RT-11 media located on VAX's across the network.

In general the lesson we have learned with conversion is that it is smart to be dumb.

Ethernet

The initial system to use these techniques was VAMP. A single-line connection from RT-11 (or our systems) to a single remote VAX/VMS system. With SHAREnet we extended VAMP to handle remote SHAREplus hosts (and nodes).

Ethernet presented new possibilities and new problems. The main issue was the ability to handle multiple node addresses instead of a single target machine.

To handle addressing we used the TP: unit numbers TP0: to TP7:. Some sites have more than eight nodes: but, none of them has more than eight servers. We believe eight is enough for an RT-11-based system.

To handle network node assignments, and other new functionality, we developed five utilities:

CONNECT   TPu: node-name

This utility sends an internal message to the local TPnet handler (or ACP) to associate a TP: unit with a specified node name. CONNECT can also report node-unit associations.

CONNECT can also be used to break a connection, define the name of the local node, and perform a wildcard connect (to ANYONE) to the first TPnet server that replies.

NODES

The NODES utility tests and reports connections. It also reports Ethernet station addresses and the operating system running at the target node. NODES is basically a SHOW NETWORK tool.

JOIN node-name

Creates a virtual terminal connection to the node. Basically the same as 'SET HOST node-name'.

LOGON and LOGOFF
These two implement some basic UIC-based file protection for multi-user system connections.

Under TPnet assignments are made to particular nodes with the TP: unit number:

```
. connect tp0: quark
. connect tp1: mungo
. assign tp0: dat:       ! QUARK::DAT:
. assign tp1: tmp:       ! MUNGO::TMP:
```

For TPnet we also developed new virtual terminal software to support fully interactive logins across the connection. To do this we basically emulate a SET HOST to self at the target server. The main implication of this work is translating VAX/VMS terminal QIO's under RT-11. The result is not perfect but more than enough for editors etc. We could do more work in this area - but, if we did it would go in the direction of a LAT implementation since that has the speed.

There is not a lot of pressure on us for more virtual terminal support. The reason is simply that the network connection is so transparent that it is rarely necessary to go across to VMS. For example, it is usually faster to use KED locally to edit a remote file than to use a native VMS editor. Another reason is that many sites have some form of terminal switch which gives them a direct terminal connection to their VAX's.

Node Capabilities
TPnet is designed to run under RT-11. It has to be small, fast and easy to use. It does not have to be all things to all people.

RT-11 nodes are end-nodes. They support out-going connections only. It is not possible to access files at an RT-11 node from another machine. Only about 5% of RT-11 systems would be willing to pay the 4kw to 6kw necessary to implement a server.

RT-11 nodes use the TP.SYS or TPX.SYS device handler. This provides its own support for Q-bus hardware. Unibus is not supported. The amazing thing about this handler is that it packs all the ACP code and the Ethernet hardware handling into about 1200 words. All other Ethernet handlers I have seen so far need about 3k words buffer space. TP: uses none.

VMS nodes are server-only nodes. Anyone can talk to them. But, they cannot make out-going requests. One goal for VMS was not to require any special system privileges at all. Some RT-11 sites are sometimes fairly low on the totem-pole and cannot "persuade her with gun or lariat, to come across for the proletariat" (Dorothy Parker).

SHAREplus nodes support both ACPs and servers. Full transparency is achieved in both directions. Separate ACP and server processes are used. The RT-11 NQX: handler is used under SHAREplus. SHAREplus provides the fullest support for TPnet but, the process-process-handler ACP path makes it a little slower than the RT-11 implementation.

The NQX: handler works well, but, it has an awful habit of blocking the system for long periods during I/O completion (under rare circumstances). Synchronising I/O completion is problematic under RT-11. NQX basically says: "Now, I raise CPU priority to seven and block anyone who would get in my way and I guess someone, somewhere in the future will eventually lower it."

The RT-11 Unibus handler is not supported. It could be, but, I have been hoping for a cleanup of the initialization code in this handler which breaks more rules than I can count (which is about up to seven).

The VAX/VMS and SHAREplus servers both support virtual terminals. These are described either above or below in this article.

We call some of these restrictions features. Any attempt to support ACP functionality at the VAX end would require system code, which would require updates at 'Major New Release Time', which would cause everybody concerned headaches. An alternative is to provide one of those 'special file transfer utilities' at the VAX end and maybe a library. We have looked at it, but there is little demand.

RT-9

The same goes for the RT-11-end server capability. We could move the server into a system job. But, realistically this would only help XM users and we don't know many. Most RT-11 sites that use a network have gone back to unmapped systems (for their realtime work). They use the VAX/VMS hosts to do the things they would have done under XM.

We have requests for support for TSX-plus which we are scheduling time to look at. I am not sure how many of the techniques described above would move to TSX-plus.

An RT-11 ACP
There is another way of looking at the projects described here. What we have basically developed is an 'ACP' for RT-11 environments.

We call an ACP an 'access processor'. The original name of 'ancilliary control process' seems a bit turgid. An ACP is a general-purpose interface to a file system. Something that RT-11 has classically omitted.

To prove the point we used the same basic ACP front-end to implement a read-only ACP for RSX F11A disks under SHAREplus (we did this as part of a project to implement a subset RSX emulator). We barely needed to change the front-end. We added some support for default and explicit directory UIC's (in the piggy-back area).

In the meantime we prefer a utility (F11) that we developed to read RSX, VMS and RT-11 disks. But, the F11A ACP showed that it is possible to support foreign file systems under RT-11 with almost no restrictions. We have toyed around with the idea of supporting VMS or MS/DOS file structures.

We put the F11 utility on the DECUS 1989 München Symposium collection - it will probably filter across to the U.S. symposium tape. We added RT-11 directory support as an afterthought because we noticed that the form:

```
DY:[MYDISK]
```

could be used to access RT-11 logical disks without having to MOUNT them. This makes it easy to look at backup disks with lots of unknown logical disks and pick-off files. In particular DECUS distributions. F11 (now a misnomer) supports nested RT-11 logical disks, such as DY:[MYDISK.SUBDSK]. (This utility is also our current workaround for the VMS EXCHANGE problem).

Two Asides
Since I seem to be wandering all around the place in this article I may as well talk about two vaguely related projects.

The first is a system we did a long time ago called VRT - an RT-11 emulator for VAX/VMS. We built this by taking a little single-user implementation of RT-11 we had done and by hooking it up to the VAX/VMS server code for VAMP. The VAMP code required almost no modification. The whole project took less than three days to get running.

The second concerned the subset RSX emulator for SHAREplus. SHAREplus was designed to handle alternative monitors so there were no real problems catching RSX executive directives. Writing the emulator was an unexpected delight. If RSX is one thing it is consistant. Its rules are almost never broken. This meant that the emulator could be put together a module at a time with almost no need to revisit code already written. Like a jigsaw puzzle. EDT and TKB were up and running in about 10 days.

But, the RSX emulator did present one major problem: we wanted to use RT-11 directories. Now, mapping RT-11 onto RSX or VMS is not all that hard. But, going the other way around is, to put it mildly, more difficult. RSX simply does not have simple notions like .LOOKUP and .ENTER (VMS does). RSX treats a file-structure as an add-on to a disk subsystem. There are dozens of ways to attach and detach files. And all of them get used all the time. We put a verbal executive directive trace into the emulator and gasped at what we saw happening. Some files get opened about ten times before they get used. EDT updates its journal file about every twelve bytes.

RT-10

The other basic problem is that RT-11 directory entries do not store all the information required by RSX programs. In particular the file format and last-used-byte information. We solved this partially with a list of filetypes and some pattern matching routines. The 'file system' took up most of the work. The terminal was also interesting.

Neither of these emulators (or the TSX emulator) have ever had a lot of use. I suspect that only one site ever used the RSX emulator. But, they were fun to write. And after all those years it was nice to see an RSX emulator running in an RT-11 environment.

Everyday Use
Using the techniques outlined above we have achieved 99% transparency under RT-11 and 100% under our own systems. What does this mean?

At our own site we rarely use RT-11 logical disks. 99% of our source, object and data files are stored remotely in VMS directories. We don't ever need to SQUEEZE. We use over 100 remote directories at present.

Most of us have a sub-process that is permanently connected to our local VAX. [ctrl/f] is enough then to work under VMS.

All our editors and compilers run on the PDP-11 machines. There is no noticable speed difference (except that PDP-11 editors appear to be faster). Very long LINKs with over 50 modules can cause problems - in two cases we keep the object files local.

Our document processor runs under PDP-11s or VAXs. PDP-11 output is sent to the VAX/VMS spooler for output to the laser printer.

We also use TPnet to talk between our PDP-11's. We keep floppies etc. off the main machine. In particular we can configure very small PDP-11 systems which have access to all system resources across the network. This simplifies software debug.

Some of sites run large numbers of large VAX's (you know, where SHOW NETWORK takes about half an hour). TPnet lives quite happily with all kinds of broadband-baseband converters, DECnet and TCP/IP (to whom it is not related).

I think our main goal has been achieved: we rarely notice the network is there. Creating a new directory at the VAX means entering assignments in two startup command files. Apart from that, there is no real difference between local and remote devices. Networks should be neither seen, nor heard.

RT-11

A FORTRAN IV PROGRAMMING STYLE
FOR
RT-11 AND TSX-PLUS

Nick Bourgeois
NAB Software Services, Inc.
Albuquerque, NM

Why use FORTRAN IV when FORTRAN 77 has so much to offer? There are many reasons for this, several of which follow. FORTRAN 77 requires floating point hardware, and there are many PDP-11s still in service and available on the used market that do not have such hardware. FORTRAN IV is significantly less expensive to buy than is FORTRAN 77. The executable image of a FORTRAN IV program is generally smaller than is the executable image of the same program compiled with FORTRAN 77.

Careful attention to style and the use of a preprocessor offers some of the advantages of FORTRAN 77 while using FORTRAN IV in the RT-11 and TSX-Plus environments. Also, additional features that are not available with FORTRAN 77 are available when using a preprocessor. The preprocessor utilized here is named MP and is distributed with DECUS C [1].

This paper uses a sample program, SNOOPY, designed to run under TSX-Plus to illustrate the the use of the preprocessor, MP, and the programming style. The program is made up of the modules: SNOOPY.PGM, SNOOPY.DAT, GTDATA.SUB, INIT.SUB, SETTIM.SUB, WODSTK.SUB, WORK.SUB, and the indirect command file SNOOPY.COM to build the program. Copies of these modules are included at the end of this paper.

MP PREPROCESSOR

The MP preprocessor offers the ability to strip out comments, include files, define variables, and conditionally use a block of code or an alternate block of code. MP is case sensitive, hence, both MP directives and MP variables must be consistent in case. MP directives are all in lower case, must

begin with a # character, and start at the beginning of a line.

The comments that are stripped by MP are delimited by /*
and */. Everything between and including the two slashes does
not appear in the output file. Comments are inserted into the
input file as follows:

/* This is a comment */

FORTRAN COMMONs may be included in all pertinent routines
with the MP #include directive. The #include directive is used
as follows:

#include "DEV:FILNAM.EXT"

Note that the device, filename, and extension are contained in
double quotation marks.

There is no PARAMETER statement in FORTRAN IV. However,
the MP #define directive may be used as a substitute for this
FORTRAN 77 feature. The #define directive is used as follows:

#define VARIABLE xx
            or
#define variable xx

Where xx is the numeric value to be assigned to the variable or
parameter.

Even FORTRAN 77 does not permit conditional compilation.
This may be accomplished with MP's #ifdef, #else, and #endif
directives as follows:

#ifdef VARIABLE
    <block of code>
#endif
          or
#ifdef VARIABLE
    <block of code>
#else
    <block of code>
#endif

In the first case above, the block of code is included if the
VARIABLE has been defined. In the second case, either one or
the other blocks of code are included in the output file.

When executed the MP preprocessor requires two arguments,
an input file specification and an output file specification.
Execution is initiated as follows:

RUN dev:MP
dev:input.fil dev:output.fil

FORTRAN statements such as "READ (lun'rcd) [list]" that

make direct access to data files will cause MP to output an
error message because only one apostrophe or single quote mark
has been encountered on the line. This may be averted with a
comment containing an apostrophe mark following the statement as
follows:

                   READ (lun'rcd) [list] ! ' Fool MP

                        PROGRAMMING STYLE

        All program modules should be kept short, typically less
than a page. This permits easier comprehension of just what the
code it doing. Each module should start with its identification
and be followed with a description of just what the module will
do. Write this before writing any code!

        In addition, the main module should contain a list of all
the files required to build the program, an annotated edit
record, and a hierarchical diagram or some other overview of the
structure and flow of the program. This overview should include
the relationship of all the subroutine and function calls that
are external to the FORTRAN language such as calls to library
routines.

        The code in the main module should be kept to a minimum.
This module is used primarily for documentation and tying the
whole group of modules together. A convenient structure for the
code in this main module would be as shown below:

                        CALL INIT
                        CALL WORK
                        CALL QUIT

        This permits a simple overlay structure that has little to
no impact on execution speed. Place the MAIN module in the root
segment and each of the other three modules along with their
supporting modules, if any, in separate segments of the same
overlay region. Even for real time applications this approach
is quite usable. If more than six modules are required for any
one overlay segment, concatenate that segment's modules with the
COPY command. Then place the concatenated set in the LINK
string.

        The INIT module should contain or control the entry of any
initialization data from the keyboard and the opening of any
files required for the program. This includes all things
required to get ready to begin the work to be performed by the
program. The WORK module performs or controls all the processes
to be accomplished. This may also include interaction with the
keyboard, and the opening and closing of files. The QUIT module
controls all those things necessary to terminate the program in

a clean fashion, such as closing any open files. On occasion
the INIT and/or the QUIT modules are trivial, and the code is
included in the MAIN module. Also, there are many times that
the whole program is so short that it all goes into a single
MAIN.

The use of COMMONs is frowned upon by current structured
programming proponents. However, COMMONs do offer much faster
execution of the program, frequently a necessary feature for
some real time applications. This is accomplished by
eliminating the construction of the argument block on subroutine
and function calls. As penance for this sin of using COMMONs,
explicitly declare each variable on a separate line and include
a short descriptive comment. Such explicit declarations mean
that it is not absolutely essential to follow FORTRAN's INTEGER
and REAL conventions. This may or may not be an acceptable
feature.

The code should be indented to show the looping and
conditional execution structure. Use the CONTINUE statement for
all labeled statements except for FORMAT statements. The
blocked IF, ELSE, and ENDIF structure of FORTRAN 77 can be
partially simulated as follows:

```
              IF (condition)
        *        GO TO xxx
  C        ELSE
                 <block of code>
                 GO TO yyy
  C        ENDIF
       xxx CONTINUE
                 <block of code>
       yyy CONTINUE
```

While this approach is a bit stilted, it does help to make the
code somewhat more readable. It should be noted that if the
condition is satisfied, FORTRAN IV will execute only one
statement. In the example above, that statement is the "GO TO
xxx" on the continuation line following the "IF (condition)"
statement.

Where the code is the least bit complex or obscure, it
shoule be annotated as follows:

                 ! This is a comment

Such annotations will prove invaluable later when you, or
especially someone else, has to modify the program. If at all
in doubt, add a comment.

Use an indirect command file to control the build process.
It avoids the frustrations of typing errors. Add to the command
file as modules are added to the program during development.
The command file should perform the preprocessing, compilation,
linking, and for TSX-Plus the setting of the memory allocation.

Create listing files when compiling. These are useful if
any error diagnostics are produced by the compiler. The command
file execution will abort if any diagnostics are generated.
Delete both the MP output and compiler listing files next. Then
build the executable image with the linker. Delete the compiler
output object files following the linking step. The generation
and deletion of a link map file is your choice. However, the
map file is useful when debugging some large applications,
especially if they contain routines written in MACRO.

DISCUSSION OF THE SNOOPY MODULES

SNOOPY was originally written to test the implementation of
the system status EMTs in TSXLIB [2]. The main control module
source is SNOOPY.PGM. Note that the list of required files
contains a short description of each of the files. SNOOPY
modules and library routines are differentiated in the
hiearchical diagram. The program identifies itself on
initiation and announces its termination. It aborts on return
from initialization if an attempt to run under RT-11 was made.

SNOOPY.DAT is the file that contains the COMMON
declarations that must be included in the main and all
supporting routines. A description of the conditional
compilation segments is given in this module. Each variable
used within any of the several modules is explicitly declared
and briefly described. Once included into any of the other
modules, this module adds significantly to the length of a
compiler generated listing. In the attached example, the LIMIT
conditional is defined and the TSRUAP conditional is not.

The INIT.SUB module first determines that the program is
being run under TSX-Plus. It then handles the keyboard entry of
the sample interval, and conditionally the entry of the
iteration limit. It also sets the initial time values. The
description section includes information about the conditionals
in this module.

WORK.SUB controls the flow of the functional operation of
the program. The LIMIT conditional changes the flow from an
infinite loop to a finite loop. Once the next wakeup schedule
is established, this module suspends program execution. Who
else but WODSTK.SUB, would wake up SNOOPY? WODSTK obtains and
displays the system date and time before waking up the sleeping
SNOOPY. Once awakened the WORK control module then calls for
the collection and display of the data, and loops back for
another cycle.

The SETTIM.SUB module handles the incrementing of the time
parameters for establishing the next wakeup time.

After determining that a line is active, the GTDATA.SUB
module collects the statistical data about that line. It
performs the required conversions and then displays the data.
This process is repeated until there are no more lines generated
into the TSX-Plus system.

Construction of the program is accomplished by the indirect
command file, SNOOPY.COM. To execute this command file under
RT-11, first define the following UCL command:

DISPLAY :== TIME

The MP preprocessor output files all have the extension of
.FOR. This permits simple input to the compiler.

The compiler generates both listing and object output
files. If any compilation errors occur, the command file
aborts, allowing perusal of the diagnostic statements contained
in the listing file(s). Once the compilations are successfully
completed, the .FOR and .LST files are deleted.

The linking step is now performed. It creates a root
segment and two overlay segments in the first overlay region.
Segment 1 is the INIT module, and segment 2 contains the WORK
and its supporting modules. A map file is also created. Next
the object and map files are deleted.

Then the memory partition size is set for TSX-Plus.
SNOOPY's memory requirements are generally much smaller than is
the typical default memory partition size.

Finally, the SNOOPY program is executed with a sample
interval of 5 seconds for 3 iterations.

CONCLUSION

This programming style has evolved over the years. It has
proven to be quite workable and very useful to the author. Like
all software development processes this one is probably still
evolving. The only thing constant seems to be change.

REFERENCES

1. David Conroy, Robert Denny, Charles Forsyth, Clifford Geshke,
    and Martin Minow, C Language System, #11-SP-18, Digital
    Equipment Computer Users Society, Marlboro, MA, November 1983.

2. Nick Bourgeois, TSXLIB: A FORTRAN Callable Library
    Implementation of EMTs for TSX-Plus, #11-490, Digital
    Equipment Computer Users Society, Marlboro, MA, May 1989.

                              NOTES

        PDP-11 and RT-11 are registered trademarks of
        Digital Equipment Corporation, Maynard, MA.

        TSX-Plus is a registered trademark of S & H
        Computer Systems, Inc., Nashville, TN.

                         <89g08a>

```
        PROGRAM SNOOPY
C
C       Nick Bourgeois
C       NAB Software Services, Inc.
C       Albuquerque, NM
C
C
C DESCRIPTION:
C
C       Report TSX-Plus system activity and the status of all logged
C       on lines to the terminal at some regular interval.  The
C       interval is specified at run time in seconds.  The data
C       displayed is the line number, line status code, project and
C       programmer numbers, program name, execution status code,
C       memory use and position, connect and cpu time.
C
C       The following files are required to build this program:
C
C               SNOOPY.COM      Indirect command file to build this
C                                 program.
C
C               SNOOPY.DAT      Unlabeled COMMON for all modules.
C
C               FORLIB.OBJ      FORTRAN IV OTS library.
C               MSCCHG.OBJ      Routine to notify the TSRUAP charge-
C                                 back data logger.
C               SYSLIB.OBJ      RT-11 system subroutine library.
C               TSXLIB.OBJ      TSX-Plus system subroutine library,
C                                 DECUS #11-490.
C
C               SNOOPY.PGM      This main control module.
C
C               FORTRA.SAV      FORTRAN IV compiler.
C               LINK  .SAV      RT-11 linker.
C               MP    .SAV      Preprocessor to include COMMON,
C                                 from DECUS C, DECUS #11-SP-18.
C
C               GTDATA.SUB      Data gathering and display module.
C               INIT  .SUB      Initialization module.
C               SETTIM.SUB      Wakeup time setting module.
C               WODSTK.SUB      Completion routine module.
C               WORK  .SUB      Work control module.
C
C       See SNOOPY.DAT for an explanation of the conditional code
C       embedded in this program.
C
C
C EDIT RECORD:
C
C       29-Aug-83:  Original creation.
C
C       05-Jan-84:  Added code for notifying the TSRUAP chargeback
C                     system that this program has been run.
C
C       17-Oct-87:  Changed from fixed 10 second sample interval to
C                     entry of sample interval from keyboard.
C
C       06-Jul-89:  Revised to allow MP to include COMMONs.  Added
C                     conditional code to allow entry of maximum
C                     iteration count and notification of execution
C                     to chargeback system.
C
C
C HIERARCHICAL DIAGRAM:
C
C       In the diagram below, items listed in capital letters are
C       routines contained in the source modules for this program,
C       and items listed in lower case letters are contained in
C       either SYSLIB or TSXLIB.
C
C       SNOOPY
C               INIT
C                       iserr
C                       itslin
C                       iherr
C                       gtlin
C                       len
#ifdef  LIMIT
C                       gtlin
C                       len
#ifdef  TSRUAP
C                       mscchg
#endif
#endif
C                       gtim
C                       cvttim
C               WORK
C                       SETTIM
C                       isched
#ifdef  LIMIT
#ifdef  TSRUAP
C                       mscchg
#endif
#endif
C                       suspnd
C
C                               WODSTK
C                                       date
C                                       time
C                                       resume
C
C                               GTDATA
C                                       ilnsts
C                                       ippnum
C                                       ipgnam
C                                       r50asc
C                                       iexsts
C                                       memuse
C                                       mempos
C                                       icontm
C                                       jicvt
C                                       jmul
C                                       jjcvt
C                                       timasc
C                                       icputm
C                                       timasc
C               exit
C
C
#include "SNOOPY.DAT"
C
C
    1 FORMAT (/,' SNOOPY:  89g06a',/)
    2 FORMAT (/,' SNOOPY:  All done',/)
```

```
      C
      C
            TYPE 1
            CALL INIT
            IF (IERR .EQ. 0)
          *     GO TO 100
      C     ELSE
            CALL WORK
      C     ENDIF
        100 CONTINUE
            TYPE 2
            CALL EXIT
            END
```

```
/*      SNOOPY.DAT

DESCRIPTION:

  COMMON declatations for SNOOPY.PGM and its supporting modules.

  NOTE: Remove the comment delimiters bracketing the
        LIMIT and TSRUAP definitions to allow compilation
        of the iteration limit and chargeback code into
        the program.  LIMIT must be included to include
        TSRUAP.  When TSRUAP is allowed, the maximum
        iteration count must not exceed 39.  Also,
        when TSRUAP is defined, the module MSCCHG.OBJ
        must be included in the root segment of the
        link command in the file SNOOPY.COM.                 */

#define LIMIT   1

#ifdef  LIMIT
/*#define          TSRUAP  1*/
#endif

      C
            COMMON              /* Word bounded declarations          */
          *     I,              /* Loop index                         */
          *     I3600,          /* Time conversion constant           */
          *     IAREA,          /* Argument area for scheduler        */
          *     ICON,           /* Connect time in minutes            */
#ifdef  LIMIT
          *     ICOUNT,         /* Iteration count                    */
#endif
          *     ICPU,           /* CPU time in clock ticks            */
          *     ID,             /* Completion routine argument        */
          *     IDELTA,         /* Sample interval in seconds         */
          *     IERR,           /* Error flag                         */
          *     IHOUR,          /* Hour in the range of 0 to 23       */
          *     ILEN,           /* Length of an ASCII string          */
#ifdef  LIMIT
          *     IMAX,           /* Maximum iteration count            */
#endif
          *     IMIN,           /* Minute in the range of 0 to 59     */
          *     IPGM,           /* Prpgram name in RAD50              */
          *     IPOS,           /* Memory position, half kilobyte units */
          *     IPPN,           /* Project and programmer numbers     */
          *     ISEC,           /* Second in the range of 0 to 59     */
          *     ISTS,           /* Execution status code              */
          *     ITICK,          /* Clock ticks, range 0 to 59         */
          *     IUSE,           /* Memory use, half kilobyte units    */
          *     JCON,           /* Connect time                       */
          *     JRES,           /* Result of an INTEGER*4 operation   */
          *     JTIME,          /* System time in ticks past midnight */
          *     LINUM,          /* TSX-Plus line number               */
          *     LNSTS,          /* TSX-Plus line status code          */
          *     POS,            /* Memory position in kilobytes       */
          *     USE             /* Memory use in kilobytes            */
      C
            COMMON              /* Byte bounded declarations          */
          *     CONTIM,         /* Connect time in hh:mm:ss           */
          *     CPUTIM,         /* CPU time in hh:mm:ss               */
          *     LDATE,          /* System date string                 */
          *     LTIME,          /* System time string                 */
          *     PGNAM,          /* Program name in ASCII string       */
```

```
      *     STRING        /* An ASCII string                  */
C
      INTEGER*2
      *     I,
      *     I3600(2),
      *     IAREA(4),
      *     ICON,
#ifdef LIMIT
      *     ICOUNT,
#endif
      *     ICPU(2),
      *     ID,
      *     IDELTA,
      *     IERR,
      *     IHOUR,
      *     ILEN,
#ifdef LIMIT
      *     IMAX,
#endif
      *     IMIN,
      *     IPGM(2),
      *     IPOS,
      *     IPPN(2),
      *     ISEC,
      *     ISTS,
      *     ITICK,
      *     IUSE,
      *     LINUM,
      *     LNSTS
C
      INTEGER*4
      *     J3600,
      *     JCON,
      *     JCPU,
      *     JRES,
      *     JTIME
C
      LOGICAL*1
      *     CONTIM(10),
      *     CPUTIM(10),
      *     LDATE(10),
      *     LTIME(10),
      *     PGNAM(8),
      *     STRING(20)
C
      REAL*4
      *     POS,
      *     USE
C
      EQUIVALENCE
      *     (I3600,J3600),
      *     (ICPU,JCPU)
C
      EXTERNAL
      *     WODSTK        /* Completion routine module        */
C
      DATA
      *     I3600 /3600,0/,
      *     ID    /0/
C
/*    End of file, SNOOPY.DAT.                          */


      SUBROUTINE INIT
C
C DESCRIPTION:
C
C     Initialization module for SNOOPY.PGM.  If the program has
C     been run under RT-11, issue an error message, and return to
C     the caller.  If the program has been run under TSX-Plus,
C     prompt for and accept the sample interval, initialize the
C     time values, and return to the caller.
C
C     If the conditional, LIMIT, has been defined, prompt for and
C     accept a value for the maximum number of iterations.
C
C     If the conditional, TSRUAP, has also been defined, notify
C     the chargeback logger that this program has been run.
C
C
#include "SNOOPY.DAT"
C
C
    1 FORMAT (I5)
C
C
      CALL ISERR              ! Intercept system errors
      CALL ITSLIN (IERR)      ! Get TSX line number
      CALL IHERR              ! Restore system error intercept
      IF (IERR .EQ. 0)
      *     GO TO 100
C     ELSE
      CALL GTLIN (STRING,'Enter the sample interval:')
      ILEN = LEN (STRING)
      DECODE (ILEN,1,STRING) IDELTA
#ifdef LIMIT
      CALL GTLIN (STRING,'Enter the maximum number of iterations:')
      ILEN = LEN (STRING)
      DECODE (ILEN,1,STRING) IMAX
#ifdef TSRUAP
      CALL MSCCHG ('SNOOPY','PR')    ! Tell SYSNAB
#endif
#endif
      CALL GTIM (JTIME)       ! Get system time
      CALL CVTTIM (JTIME,IHOUR,IMIN,ISEC,ITICK)
      GO TO 110
C     ENDIF
C
  100 CONTINUE
      TYPE *,'SNOOPY:  Must be run under TSX-Plus'
  110 CONTINUE
      RETURN
      END
```

```
        SUBROUTINE WORK
C
C DESCRIPTION:
C
C       Work control module for SNOOPY.PGM.  Call a routine to set the
C       time values for the next wakeup interval.  Schedule the next
C       wakeup time and go to sleep.  Upon awakening, call a routine
C       to gather and display the data.  Then loop back for another
C       cycle.  Abort with an error message on a scheduling error.
C
C       If the conditional, LIMIT, has been defined, change from an
C       infinite loop of iterations to a maximum number of iterations.
C
C       If the conditional, TSRUAP, has been defined notify the charge-
C       back data logger of each iteration.
C
C
#include "SNOOPY.DAT"
C
C
#ifdef  LIMIT
        DO 120 ICOUNT = 1,IMAX
#else
  100 CONTINUE
#endif
        CALL SETTIM
        IERR = ISCHED (IHOUR,IMIN,ISEC,ITICK,IAREA,ID,WODSTK)
        IF (IERR .EQ. 0)
   *        GO TO 110
C       ELSE
            TYPE *,'SNOOPY:  ISCHED error'
            GO TO 130
C       ENDIF
C
  110       CONTINUE
#ifdef  LIMIT
#ifdef  TSRUAP
        CALL MSCCHG ('SNOOPY','PU')    ! Update SYSNAB
#endif
#endif
        CALL SUSPND                    ! Go to sleep
        CALL GTDATA
#ifdef  LIMIT
  120 CONTINUE
#else
        GO TO 100
#endif
C
  130 CONTINUE
        RETURN
        END
```

```
        SUBROUTINE WODSTK
C
C DESCRIPTION:
C
C       Completion routine for SNOOPY.PGM.  When executed, get and
C       display the system date and time.  Then wake up the main
C       program.
C
C
#include "SNOOPY.DAT"
C
C
    1 FORMAT (/,' ',9A1,'    ',8A1)
C
C
        CALL DATE (LDATE)
        CALL TIME (LTIME)
        TYPE 1,(LDATE(I),I=1,9),(LTIME(I),I=1,8)
        CALL RESUME                 ! Wake up SNOOPY
        RETURN
        END
```

```
        SUBROUTINE SETTIM
C
C DESCRIPTION:
C
C       Time setting module for SNOOPY.PGM.  Set the time values
C       for the next wakeup time.
C
C
#include "SNOOPY.DAT"
C
C
        ISEC = ISEC + IDELTA
        IF (ISEC .LT. 60)
     *     GO TO 100
C       ELSE
            ISEC = ISEC - 60
            IMIN = IMIN + 1
            IF (IMIN .LT. 60)
     *         GO TO 100
C           ELSE
                IMIN = IMIN - 60
                IHOUR = IHOUR + 1
                IF (IHOUR .LT. 24)
     *             GO TO 100
C               ELSE
                    IHOUR = IHOUR - 24
C               ENDIF
C           ENDIF
C       ENDIF
  100 CONTINUE
        RETURN
        END
```

```
        SUBROUTINE GTDATA
C
C DESCRIPTION:
C
C       Data gathering and display module for SNOOPY.PGM.  Collect
C       and display statistics for each logged on line using TSX-Plus
C       system services available via TSXLIB routines.
C
C
#include "SNOOPY.DAT"
C
C
    1 FORMAT ('  ILNSTS        IPPNUM  IPGNAM  IEXSTS',
     *        '  MEMUSE  MEMPOS     ICONTM    ICPUTM')
    2 FORMAT (2X,I2,1X,I3, 2X,I5,',',I5, 2X,6A1, 6X,I2,
     *        4X,F4.1, 3X,F5.1, 2X,8A1, 2X,8A1)
C
C
        TYPE 1
        LINUM = 0
  100 CONTINUE
        LINUM = LINUM + 1                ! Line number
        CALL ILNSTS (LINUM,LNSTS,IERR)! Line status
        IF (IERR .EQ. 0)
     *     GO TO 100                    ! Line not logged on
C       ENDIF
        IF (IERR .EQ. 2)
     *     GO TO 110                    ! No more lines
C       ELSE
            CALL IPPNUM (LINUM,IPPN)    ! Project-programmer number
            CALL IPGNAM (LINUM,IPGM)    ! Program name
            CALL R50ASC (6,IPGM,PGNAM)!  Convert to ASCII
            CALL IEXSTS (LINUM,ISTS)    ! Execution status
            CALL MEMUSE (LINUM,IUSE)    ! Memory use
            USE = FLOAT(IUSE) / 2.      !  Convert to kilobytes
            CALL MEMPOS (LINUM,IPOS)    ! Memory position
            POS = FLOAT(IPOS) / 2.      !  Convert to kilobytes
            CALL ICONTM (LINUM,ICON)    ! Connect time
            I = JICVT (ICON,JCON)       !  Convert to INTEGER*4
            I = JMUL (J3600,JCON,JRES)!  Convert to seconds
            CALL JJCVT (JRES)           !  Swap words
            CALL TIMASC (JRES,CONTIM)  !  Convert to ASCII
            CALL ICPUTM (LINUM,ICPU)    ! CPU time
            CALL TIMASC (JCPU,CPUTIM)  !  Convert to ASCII
            TYPE 2,LINUM,LNSTS,IPPN(1),IPPN(2),(PGNAM(I),I=1,6),
     *            ISTS,USE,POS,(CONTIM(I),I=1,8),(CPUTIM(I),I=1,8)
C       ENDIF
        GO TO 100
C
  110 CONTINUE
        RETURN
        END
```

```
! SNOOPY.COM     NAB     09-Jun-89/21-Jun-89.
!
!       Build SNOOPY.SAV from its component parts and then
!       execute it.
!
DATE
TIME
!
DISPLAY         <Build the .FOR files>
RUN     SY:MP
SNOOPY.PGM      SNOOPY.FOR
RUN     SY:MP
INIT.SUB        INIT.FOR
RUN     SY:MP
WORK.SUB        WORK.FOR
RUN     SY:MP
SETTIM.SUB      SETTIM.FOR
RUN     'SY:MP
WODSTK.SUB      WODSTK.FOR
RUN     SY:MP
GTDATA.SUB      GTDATA.FOR
!
DISPLAY         <Compile>
RUN     SY:FORTRAN
SNOOPY,SNOOPY=SNOOPY/W
INIT,INIT=INIT/W
WORK,WORK=WORK/W
SETTIM,SETTIM=SETTIM/W
WODSTK,WODSTK=WODSTK/W
GTDATA,GTDATA=GTDATA/W
^C
DEL     SNOOPY.(FOR,LST)
DEL     INIT.(FOR,LST)
DEL     WORK.(FOR,LST)
DEL     SETTIM.(FOR,LST)
DEL     WODSTK.(FOR,LST)
DEL     GTDATA.(FOR,LST)
!
DISPLAY         <Link>
RUN     SY:LINK
SNOOPY,SNOOPY=SNOOPY,SY:TSXLIB/F/G/W//
INIT/O:1
WORK,SETTIM,WODSTK,GTDATA/O:1
//
^C
DEL     SNOOPY.(MAP,OBJ)
DEL     INIT.OBJ
DEL     WORK.OBJ
DEL     SETTIM.OBJ
DEL     WODSTK.OBJ
DEL     GTDATA.OBJ
!
DISPLAY         <Set the memory allocation>
RUN     SY:SETSIZ        SNOOPY/D:4
RUN     SY:SETSIZ        SNOOPY
!
TIME
!
DISPLAY         <Execute>
RUN     SNOOPY
5
3
```

```
!
TIME
!
!       End of file, SNOOPY.COM.
```

John M. Crowell
RT-11 Newsletter Editor
P. O. Box 128
Davis, CA   95616

08-Aug-89

Dear Mr. Crowell:

If Ed Judge can get a letter published, I guess I can give it a try.

I obtained my LSI-11 computer back in 1978 as the Heathkit
version.   Since then, I've written several (probably hundreds) of
programs in MACRO, C, FORTRAN, and even BASIC. I started out with a
paper-tape system and did one edit, assembly, link, and run per night
with a Teletype Model 33. Used to go out to dinner for two hours just to
avoid the noise.   Finally bought an RX-01 and a video terminal, then a
hard disk, and printer. I don't want to tell you what I   spent on some of
that hardware back in the late 70s and early 80s.

Getting to the point, your letter in the August Newsletter   has
prodded me into telling about my dealings with mag tape formats. You
mentioned that you could get RT-11 to VMS fairly well, but had
problems going from VMS to RT-11.

About 5 years ago, a lot of nifty software was available from the
DECUS library on DOS/11-format mag tapes only. Since necessity is the
mother of invention, I purchased a package in DOS format. I used the RT-
11 DUMP utility to discover the format of the data on the tape. Fairly
trivial: a 7-word header that had, among other things, three RAD50
words   that   were   the file name, some other integers that were most likely
a PPN ([nnn,mmm]), maybe a date in some Julian format, and I think there
might have even been a size in blocks. I wrote a small program in MACRO
to read these headers, then the files, one block at a time, and write them
onto the default disk (DK:) with the same name. As most of the library
tapes had a relatively small number of files, I often wrote directly to a
floppy disk, or cleared a logical unit on my hard disk and assigned that to
DK:. This only copied the files in binary mode, block for block, which
often required one pass with a text editor to fixup end of line characters
or trailing blanks. Cheap, dirty, but it did solve the problem and I got
a few good programs converted to run on RT-11 out of it.

On another occasion, I had to extract some documentation off of a VAX
TU-58 (DECtape II) cartridge. I didn't have a running VAX to use, in fact,

the information on the tape was necessary to get the machine booted in the first place. I had a TU-58 drive on my system at home, and, again using the DUMP utility, I was able to deduce the structure of the data on the tape. This meant figuring out the index blocks, directories, block usage tables, and finally the data blocks themselves. I wish that I had access to some of the excellent documentation that is now available for this format, which I believe is FILES-11 or ODS. Anyway, another MACRO program to the rescue. It had to do a little more processing of the text blocks, since these were stored as a 16-bit word (the string length), followed by the string itself. The only trick was that the word that held the length was always on an even (word) boundary, and null padding was used between strings if necessary. I only had to extract a small number of files, but the program did its thing.

Now, I use a UNIX (11/73) system at work. Occasionally, I have to bring files from home (RT-11) to the office where I can do some editing and testing, then bring them back home again. There are only three mag tape data formats that exist in the UNIX world: TAR, DUMP, and just plain raw data. None of these are compatible with RT-11, much less any ANSI standard. I found the easiest way to get stuff from RT to UNIX was to use the UNIX "DD" utility, which is able to read or skip blocks or entire files. Since an RT-11 mag tape file is placed on tape as a header block, a file mark, the data blocks, a file mark, and a trailer block and another file mark, I have a small command script file that reads (skips) one "file" (the header block and its file mark) into the "null" device, then reads the next "file" (the data blocks up until the file mark) into my specified output file, and finally reads (skips) one more "file" (the trailer block and its file mark) into the "null" device again. The tape is then left positioned at the beginning of the next file, if any. RT-11 also writes a volume header block on the tape, but since this is immediately followed by the first file's header block, it gets skipped along with the header as if it isn't there. I find this method extremely reliable as long as I know what the names of the files are on the tape, which I solve by getting a hard copy directory listing of the tape under RT-11 when I make it. Going back from UNIX to RT-11 is a little more painful. As there is also a lot of software written on UNIX machines, they have more or less standardized on the "TAR" format. In this format, each file is preceded by a one-block header that contains the full path name of the file, its size in bytes, various dates, protection modes, and owner identification numbers. Each file takes one or more full 512-byte blocks on tape. Each subsequent file is laid out exactly the same. The end of the medium is marked by a file header block that is all nulls. Tapes then have a few tape marks on them, written by the device driver when the tape is closed. The TAR program also allows information on mag

tape to be written as larger blocks, up to 10240 bytes each. However, each of these big blocks is handled in 512-byte chunks at a time; information is just packed in memory and written to the tape less often. For example, ten files of 512 bytes (or less) can be written into one 10240-byte tape block. A 20-block file will utilize two of these 10240 byte blocks, since the header information occupies one block. I wrote a C program that directly manipulates the TM-11 hardware on my RT-11 system to read tape blocks and pass them back 512 bytes at a shot to the main program that figures out what the RT-11 file name should be, opens the file, writes the data in binary mode, and closes the file when it has written enough data out.

These aren't really difficult programs to implement. I propose that you (or someone you know) should be able to come up with something under RT-11 that can write and/or read a VMS-compatible tape. Using the SPFUN requests, you can open the tape drive, read or write almost any block size, and close the tape drive. If you don't want to go through all the hassle of making an ANSI formatted tape, you could design your own format that is simple to implement on both RT-11 and the VAX. You could even write the program in C so the same program can be used on both systems. As someone famous once said, "there are always alternatives", and with a little programming effort, you should be able to get information between the systems so that you can then use the utilities provided with VMS or RT-11 to write their own tapes.

I enjoy reading the Newsletter and find myself going first to the RT-11 section, then IAS/RSX, Unix, Hardware, and when time permits, the VAX section, primarily to see what sort of problems the other groups are having, as well as their solutions.

Please feel free to edit this letter as you see fit. It was formatted with a rather old version of standard RUNOFF (not Bonner Labs). Keep up the excellent work on the Minitasker. I look forward to the next issue.

Bob Meister
Connsult Associates
104 Twin Brook Road
Hamden, CT 06514

Billy Youdelman
Consulting Electrical Engineer
PO Box 1207
Culver City, CA 90232
(213) 839-7673

15-Aug-89

Mr. John Crowell
RT-11 Newsletter
PO Box 128
Davis, CA 95616

Dear John,

Due to an unusually busy summer this letter is much later than I'd prefer,
however ..

Shortly after I sent a copy of my anti-tailgating program to the RT-11
Newsletter version 6.4 of TSX-Plus was released. For some years now I've
maintained programs on two systems which handle about 250 calls per day,
365 days a year. Prior to v6.4 I'd hear of some tailgating incident once
or twice a month. Since installing v6.4 there has not been one single
incident, nor can it be forced to occur even under conditions purposefully
made favorable. The 0.5sec interval for testing modem line status is more
than adequate, as merely getting a modem to answer another call and assert
its carrier takes at least that long.

Notail.mac may still be useful to sites wanting some (possibly public)
lines more tightly controlled than others, however the exact same thing
(for all modem lines) can now be accomplished merely by making timout=1
and offtim=1 in tsgen, or with tsxmod. It should be mentioned most modems
incorporate some small (ca. 0.7sec) delay between actuall loss of carrier
and bringing dcd false, thus operating with timout and offtim at their
minimums with such modems does not lead to lines being killed due to minor
dropouts - however, as was the case at one of my sites, any glitching from
the modems (such as in the dcd line when going off hook) may be taken as
a failed call. The same modems had worked fine with the last few releases
of TSX-Plus.. My solution was to set the modems to answer on the second
ring, which for whatever reason eliminated the dcd glitch. This has the
additional benefit of ensuring the system has reset the line before the
phone is answered again.

I'd like to thank Mr. Stephen Brenner of S&H not only for his help in
uncovering the bug in the above modems, but for his very thorough
treatment of this situation in TSX-Plus, and far faster than I'd ever
expected..

Regards,

Billy Youdelman

New KED Features

RT-11 Version 5.5

RT-29

Megan B Gentry
Senior Software Engineer
Small Systems Software Engineering

New Features

- KED command files
- Work-session initialization files
- Journaling
- Expanded macro support

KED command files

KED command files are text files which contain KED commands that you would normally type in response to the 'Command:' prompt.

- Command lines may be commented using '!'
- File may be loaded and executed as a KED macro or
- Invoked by @filespec at Command: prompt

Work-session initialization files

Work-session initialization files are KED command files that may be used to:

- Customize KED operation prior to each editing session
- Perform a complete editing session without further user intervention

Defaults to KEDINI.KED unless otherwise specified or suppressed

Journaling

Allows recovery of an editing session if abnormally terminated due to

- power failure
- system failure
- user failure

Keystrokes entered during the editing session are stored first in the journal buffer and later in the journal file. Journal buffer contents are moved to the journal file when:

- a certain number of modifications have been made (default of 10)
- the journal buffer becomes full
- the user requests the buffer to be written
- the editing session is complete and the user exits from KED

The frequency with which KED moves information from the journal buffer to the journal file may be increased at the cost of system performance, or decreased at the risk of losing more modifications should a failure occur.

The journal file is normally discarded when KED is terminates. You may request that the journal file be retained on exit.

RT-31

Expanded macro support

KED now supports:

- definition and use of more than one macro

- saving macros to files

- loading macros from files

- deletion of individual macros

Macros saved in files may be executed as KED command files

Macros may be nested

Changes to KED

Entering multiple commands

If a command entered to the 'Command:' prompt is terminated with a <RETURN>, the command will be executed and KED will return to the 'Command:' prompt for additional commands.

No cursor movement on failed searches

By default, KED no longer moves the cursor to the beginning or end of a file whenever a search target is not found. The cursor now stays in its current position.

Enhanced HELP

Describe the new commands

Optional suppression of full screen HELP displays

It is now possible to disable display of the full-screen HELP texts when <PF2> is pressed. SET [NO]HELP ([disables]/enables)

Optional disabling of selection during FIND operation

By default, KED selects the target in a successful FIND operation, this may be disabled. SET SEARCH [NO]SELECT ([disables]/enables)

Positioning RT-11 Files

John M. Crowell
Multiware, Inc.

In last month's Minitasker there was a discussion about how to force files to the beginning of a disk. Starting, I think, with Version 5.0, the RT-11 USR puts files of known length (i.e. copied files, or files opened with fixed maximum length) in the empty space that most closely fits the file size. In the olden days, the first free space on the disk large enough to accomodate the file was used. Some of us like it better that way - especially when we're upgrading versions of RT and just want to copy .SYS files and new utilities to a target system disk as close to the beginning of the disk as possible.

Megan Gentry, of the RT-11 Development Team, provides the following patch to RT-11 monitors. This patch will change the "best fit" algorithm back to "first fit."

```
.SIPP RT11zz.SYS/A !or whatever monitor file name you have
Base?           ;S
Search for?     101322
Start?
End?
Found at nnnnnn
Base?           0
Offset?         nnnnnn
Base        Offset      Old         New?
000000      nnnnnn      101322      1404
000000      nnnnnn+2    001403      721
000000      nnnnnn+4    010425      ^Y
```

This seems to work for all monitors. Thanks, Megan.

For those of you who do SYSGENs and want this patch as the default, the change to USR.MAC is as follows:

```
            OLD                         NEW

       BHI  9$                     BEQ  131$
       BEQ  131$                   BR   9$
  132$:MOV  R4,(R5)+          132$:MOV  R4,(R5)+
```
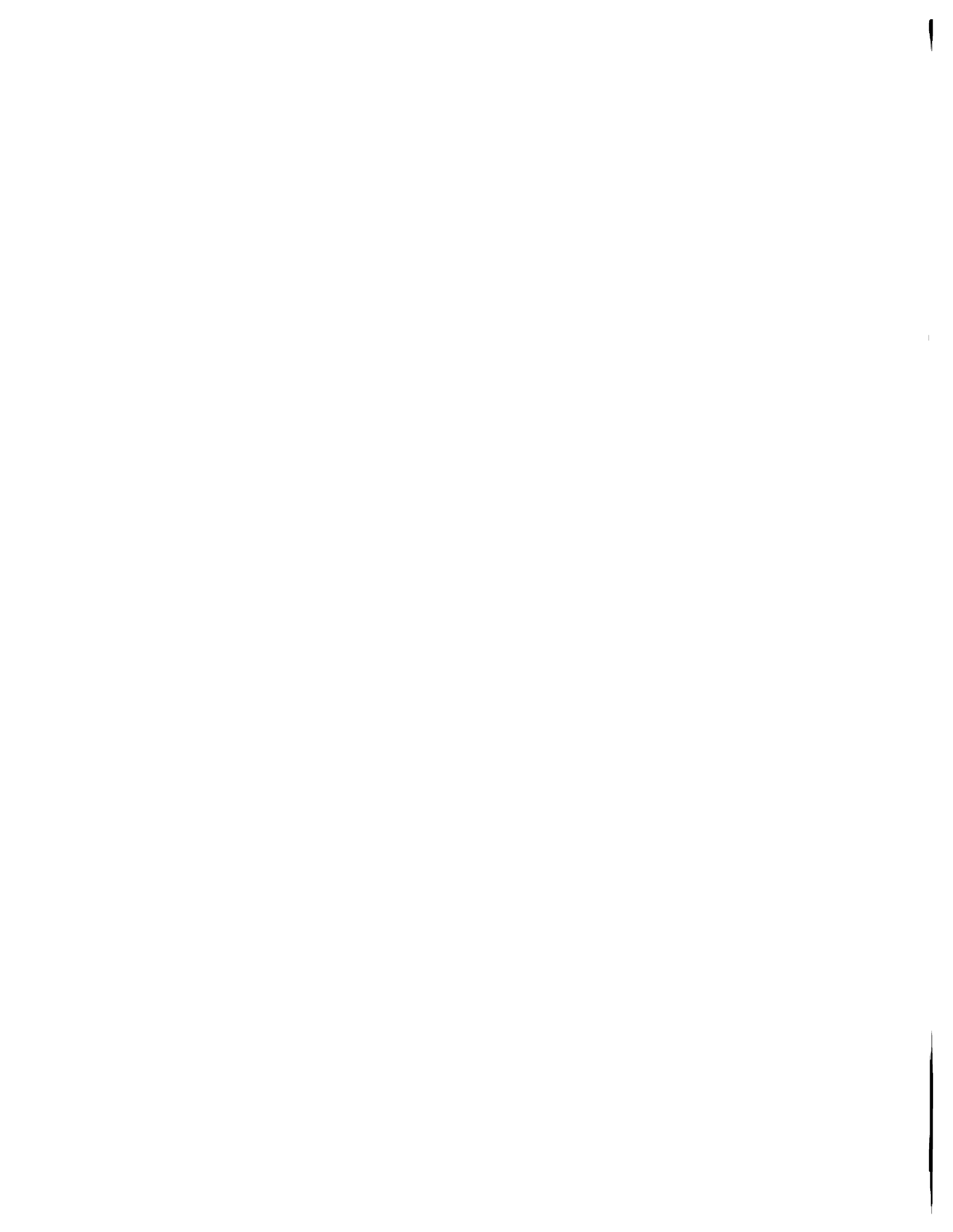
UNISIG

From the Editor *by Sharon Gates-Fishman*

This month we have a very full issue, so I won't bore you with my usual editorial drivel. Instead, let's get right down to business.

First, we have

● **Bug in 'brk'**

by Mike Mitchell of the Research Triangle Institute. This is something I saw on Usenet and thought might be of interest. If you want to contact Mike for more information about this, his address is decvax!mcnc!rti!mcm, and his phone number is (919) 541-6098.

Next, we have

● **UNISIG Sessions - Anaheim '89**

This is a preliminary schedule, subject to change at any time. There are a lot of good sessions scheduled for Anaheim, so many that there are a few time slots when Unisig sessions actually conflict with each other! Many thanks to Michael Angelo, the Unisig Symposia Committee representative, for getting me this list. For those of you who won't be joining us in Anaheim, this list might help you decide which audio tapes you might like to order.

Next, is Part I of

● **Monte Carlo OSF Meeting Trip Report**

by Steve Lazarus and Dorothy Geiger. As you may know, DECUS is a member of the Open Software Foundation, and Steve and Dorothy have been putting a lot of time and energy into representing our interests to the OSF. This is the first half of their trip report, the second half will be in next month's newsletter. Many thanks to them both for their work and for this report.

Next is

● **LAT-Telnet Problems**

by Jim Karsten of ULTRIX CSSE. This article is from a Digital internal Ultrix/Unix newsletter. Thanks very much to Sharon MacDonald for

passing this article along to me.

And last, we have

● **CERT Internet Security Advisory**

by Kenneth Van Wyck of Carnegie-Mellon University's Software Engineering Institute. This article is also culled from Usenet. CERT can be reached electronically at cert@SEI.CMU.EDU, or on their 24-hour hotline at (412) 268-7090.

If you are going to be at Symposium in Anaheim, introduce yourself! Steering Committee members will be easily identifiable by their badges (as if we need badges to make us stand out in a crowd!). There will be a UNISIG questionaire in the Tri-SIG Campground, with a nifty gift for everyone who fills one out. In the meantime, send in those articles!! Send hardcopy to:

> Sharon Gates-Fishman
> NDC Systems
> 730 E. Cypress Ave.
> Monrovia CA 91016

or e-mail to:
> amdahl!cit-vax!ndc!sgf

Bug in 'brk' *by Mike Mitchell, Research Triangle Institute, RTP, NC*

I have run across a bug with Ultrix 3.1 on both the DecStation 3100 and the bigger vaxes. It involves using *brk()* to allocate and free memory. The problem is that a process' PTE's are not invalidated properly when freeing memory. That means that a program can access memory it has just freed. It does not show up on microvaxes because their TLB cache is so small. The DecStation 3100 has a 64-entry TLB, so it does have the bug. The bug also shows up on 8600's and 785's, so you might want to test your own system. *Figure 1* shows a program that will demonstrate the bug.

I have verified that the bug is present in BSD 4.2 and BSD 4.3, but I know it has been fixed in BSD 4.3 tahoe. This bug has been reported in several times in several Usenet newsgroups! A fix for the bug has been known for several years, yet few vendors have incorporated the fix.

```
#include <signal.h>

main()
{
    char *old_break, *cp;
    int i;
    extern char *sbrk(), *brk();
    void segv();

    signal(SIGSEGV, segv);

    i = getpagesize();
    old_break = sbrk(0);                /* get the current "break" */
    (void) brk(old_break + 2*i);        /* bump it up 2 pages */

    cp = old_break + i + 256;
    *cp = 1;                            /* write into a new page */

    (void) brk(old_break);             /* return the memory */

    *cp = 2;                            /* write into the page again.  This */
                                        /* time, you should get a sigsegv */
    printf("Your brk routine is broken!\n");
    exit(1);
}

void segv()
{
    printf("Your brk routine works correctly.\n");
    exit(0);
}
```

Figure 1

....

Bug in 'brk' *continued...*

The problem is in the *vm_proc.c* file, in the routine
expand(). The starting address for the PTE's to
invalidate is not calculated correctly when freeing
memory. The code in error looks something like:

```
            if (change < 0)
                change = -change;
            else {
```

The code should read:

```
            if (change < 0) {
                change = -change;
                v -= change;
            } else {
```

Further down in the code v is passed on to *newptes()*,
and it sets up the PTE's

UNI-2

UNISIG Sessions - Anaheim '89 *preliminary schedule*

Time	Session	Title	Speaker	Length
Monday:				
9:00	UN001	UNISIG Roadmap / ULTRIX Product Panel	Reisler Kurt	1 hr
10:00	UN042	RISC Architecture and ULTRIX	Delorey Al	1 hr
11:00	UN060	New RISC DECstations	Blount Susan	1 hr
12:00	UN009	Performance Comparison of Digital's RISC-Based Systems	Lo James	1 hr
1:00	UN044	Disk Performance on the HSC/CI	Seagraves James	½ hr
1:30	UN007	UNIX/ULTRIX File System Tutorial	Stepanek Steven	1 hr
2:30	UN008	UNIX/ULTRIX File System Question-and-Answer Session	Stepanek Steven	½ hr
3:00	UN049	Increasing the Performance of the ULTRIX Filesystem	Group ULTRIX	1 hr
4:00	UN029	You Don't Have to be Root	Bartelt Mark	1 hr
5:00	UN041	Common ULTRIX to VAX Messaging	Kilman Howard	½ hr
5:00	UN057	Security Development in ULTRIX	Hall Henry	½ hr
5:30	UN032	UNIX Security Wishlist	Tihor Stephen	½ hr
6:00	UN052	Integration of Project Athena Services Into ULTRIX	Logcher Suzanne	1 hr
7:00	UN050	Kerberos: An Internet Distributed Authentication Service	Brown William	1 hr
8:00	UN015	DECsystem 5400 Technical Session	MacLean Roseann	1 hr
9:00	UN066	X Window System Overview	Wingerd Joe	1 hr
9:00	UN016	DECsystem 5800 Technical Session	MacLean Roseann	1 hr
10:00	UN019	AI, L&T, and UNISIG Joint Reception	Reisler Kurt	1 hr
Tuesday:				
9:00	UN024	X Window System-From Stock to Fully Customized: Making the Most of Your Windows on the World	Reisler Kurt	1 hr
10:00	UN062	Migrating to C for RISC	Maxwell Sid	1 hr
11:00	UN061	Migrating to FORTRAN for RISC	Maxwell Sid	1 hr
12:00	UN010	DECstation 3100 FORTRAN User's Panel	Denning Bill	1 hr
1:00	UN030	Standing at the Crossroads: Lamentations of a Mail Gateway Keeper	Avolio Frederick M.	1 hr
2:00	UN031	Introduction to Sendmail and Sendmail.cf	Avolio Frederick M.	1 hr
3:00	UN027	UNIX for VMS Users: Fundamental Concepts and Getting Started	Bourne Philip E.	1 hr
4:00	UN012	Network File System Concepts and Implementations	Kashtan David	1 hr
5:00	UN048	Detailed View of ULTRIX Network File System (NFS) Implementation	Group ULTRIX	1 hr

UNISIG Sessions - Anaheim '89 *preliminary schedule, continued...*

Time	Session	Title	Speaker	Length
Wednesday:				
9:00	UN028	UNIX for VMS Users: Convert DCL Procedures to Shell Scripts	Bourne Philip E.	1 hr
10:00	UN025	UNIX for VMS Users: Converting VMS Applications to UNIX	Bourne Philip E.	1 hr
11:00	UN011	Building Software Using Pipes and Filters, Including an Introduction to Grep, Sed and Awk	Avolio Frederick M.	1 ½ hr
1:30	UN038	Real-World Projects on UNIX, Or You and Software Code Control System (SCCS)	Gates-Fishman Sharon	½ hr
2:00	UN021	ULTRIX-32 System and Network Setup: A Cookbook Approach	Ring Nolan	1 hr
3:00	UN023	Managing Your UNIX Workstations: Congratulations You Are Now a System Manager!	Reisler Kurt	1 hr
4:00	UN039	Hierarchical Systems Management	Romero Louis	1 hr
5:00	UN055	An ULTRIX Performance Information Collection Utility	Hurley Carolyn	1 hr
		Remote Procedure Call (RPC)	Glaser Ray	1 hr
Thursday:				
3:00	UN053	Kernel Debugging on ULTRIX/RISC	Delorey Al	1 hr
4:00	UN013	The Story and Face Behind the New RISC-Based Products	Maclean Roseann	1 hr
5:00	UN017	RISC Family Hints and Kinks	MacLean Roseann	½ hr
5:30	UN043	ULTRIX System Performance on the DECsystem 3100	Seagraves James	½ hr
6:30	UN046	Open Systems Standards and Digital	Bismuth Robert	1 hr
7:00	UN004	Open Software Foundation Progress	Lazarus Stephen M.	½ hr
8:00	UN025	UNIX for VMS Users: Converting VMS Applications to UNIX	Bourne Philip E.	1 hr
9:00	UN002	UNIX/ULTRIX Hints and Kinks	Rod F.	1 hr
Friday:				
9:00	UN037	UNIX Remote Printing and VMS	Allen Daniel	1 hr
10:00	UN059	Managing and Using Printservers in an ULTRIX/UNIX Environment	Epstein Bruce	1 hr
12:00	UN033	MACH, the Universe, and Everything	Gould Ed	1 hr
1:00	UN045	Configuring Virtual Memory in the ULTRIX Operating System	Amato Joseph	1 hr
2:00	UN047	Simulating O/S Scheduling and Dispatching Algorithms	Peterson Michael T	1 ½ hr
3:30	UN035	Bringing up 2.10 BSD	Lowenstein Carl	½ hr
4:00	UN003	UNISIG/ULTRIX Feedback and Wrapup	Reisler Kurt	1 hr

Monte Carlo OSF Meeting Trip Report *by Steve Lazarus and Dorothy Geiger*

1. Introduction

The May meeting of the Open Software Foundation was held in Monte Carlo, Monoco from May 21 - May 24 1989. DECUS was represented at this meeting by Dorothy Geiger and Steve Lazarus.

The meeting was divided into two major tracks: a review of the OSF operating system component (OSC) and a managerial track. SIG meetings were also scattered throughout the four day meeting. For example, the CASE SIG ran an extensive series of meetings which ran in parallel with most of the rest of the sessions. Due to the importance of both the OSC review and CASE, DECUS efforts largely focused on these two areas, with Dorothy Geiger attending the OSC review while Steve Lazarus covered the CASE SIG meetings.

This meeting was a watershed for the OSF. The organizers of the meeting set an exhausting schedule, with more than 600 pages of technical documentation for the OSC sent to members for review less than a month before the meeting itself. With staffing nearly complete, the OSF is now addressing technical issues in a manner which more closely follows the requirements of the member organizations.

The OSF is also attempting to more clearly define its relationships with end users. At OSF's request, DECUS participated in two lunch time meetings to discuss these issues and to review a draft of OSF's plan for establishing end user relations.

2. OSC Review

2.1. Introduction

This was an intense review of all major components of the OSC. As such, participation in the OSC technical review sessions was restricted to those members holding valid AT&T System V source licenses. (DECUS members participated under the terms of their respective employer's AT&T source license agreements). During the course of the review session, OSF referred to the process of open technical review as "Engineering in a Fishbowl" or "Open Kimono Engineering". The technical review included a full discussion of schedules, issues, and technical risks. Currently, the development staff consists of 80 full time staff members, of whom 58 are in Cambridge, 7 are in Munich, 13 are contractors, and 2 interim (member company) employees.

The schedule for delivery of the OSF operating system, OSF/1, is as follows:

Vendor Kit	10/89
Application Kit	3/90
University Platform	5/90
Full Release	7/90

The *Vendor Kit* represents a stable base from which vendors may begin their porting effort and is composed of code from which OSF itself is working. The *Application Kit* is intended to be comprised of a functioning system that meets the complete OSF/1 specification; its release is, in effect, a beta test of the OSF/1 product. The *University Platform* represents a completely functional pre-release. Frequent snapshots will be provided throughout the development stage, beginning with delivery of the vendor kit.

OSF is also setting up a portability lab to assist member companies in porting OSF/1 to their individual hardware platforms. The portability lab will be set up at OSF headquarters to facilitate direct interaction with OSF developers, provide early identification of portability problems, and shorten the time required for member vendors to incorporate OSF/1 into their own product lines.

2.2. The AES and the OSC

It is important to continue to distinguish between the actual operating system provided by the OSF, (known by description as the OSC and by product name as OSF/1) and the Applications Environment Specification (AES), which represents a specification for the functionality (i.e. the system call

Monte Carlo OSF Meeting Trip Report *continued...*

interface) of an operating system.

The OSC meets the AES and provides functionality which extends beyond that specified by the AES. In so doing, the OSF represents the current "state of the standards" in that, as a practical implementation of a Unix based operating system, it includes those elements that are standardized as well as providing backward compatibility with current AT&T System V and Berkeley Software Distribution (BSD) derived systems. Additionally, OSF/1 includes features such as Sun's Network File System (NFS) which are not part of the AES.

The AES is based on a number of standards. In order of priority these standards are: POSIX 1003.1 (the kernel interface), ANSII C, and the X/Open Portability Guide (XPG).

Features are cycled into the AES as follows:

> proposed for trial use -> trial use
> -> proposed full use -> full use

Elements of the AES contained in the approved POSIX standard are specified in the AES for full use. Since ANSI C and XPG are not, as yet, fully approved as standards they are currently included on a trial use basis.

To date, those documents provided for the AES and the OSC have been outstanding and represent excellent summaries of the state of standardization within the UNIX community. For the AES, all of the functions are listed along with their level of acceptance within the AES (currently, trial-use for all but those that are part of POSIX 1003.1). The AES documentation clearly states whether a given feature is present in the X/Open Portability Guide (releases 2 or 3), POSIX 1003.1, System V (release 2 or 3), ANSII C, or BSD 4.3. The OSC matrix lists OSC functions and identifies whether a given function is included within POSIX, XPG, System V Release 2, BSD 4.3, and/or the AES.

2.3. Validation

Validation issues are clearly of concern to the OSF. They have indicated a strong desire to validate their

implementations against the underlying specifications. The OSF is currently evaluating existing technologies to provide validation for POSIX, X/Open, System V, BSD, and ANSII C. Validation of user interfaces provide a particular challenge. The OSF hopes to provide a standard scaffolding that can be used to configure, run, and evaluate all test results. (We note that, were it not for the fact that it is available only on VMS, the technology represented by the DEC/Test Manager could address the OSF's needs in this area).

2.4. Documentation

The OSF's documentation team is part of the product development organization, and works very closely with the development staff. Initially, the traditional UNIX *troff/nroff* markup languages will be used to produce documentation. It is recognized that this technology is far from modern and should be replaced as quickly as possible. The OSF is examining issues of standard document interchange formats, import/export toolkits, and more advanced storage/retrieval technologies (e.g., hypertext.)

2.5. Detailed OSC Review

The version of AIX submitted by IBM to OSF includes many features not currently available in the version of AIX shipped from IBM. Of concern to OSF are the many changes necessary to meet the OSF specifications and portability concerns; the largest of which is that of the virtual memory implementation. As originally written, AIX was designed to use specific hardware features of the IBM RT.

The details of the OSF review are covered by the necessity for a UNIX source license and OSF proprietary concerns. The scope of the discussion is represented by the following outline:

2.5.1 Standards

OSF/1 will comply with the POSIX 1003.1 standard for system calls and libraries. Since POSIX 1003.2 is not yet a standard, OSF/1 will be compliant with those aspects of 1003.2 which are stabilized.

OSF/1 will also be compliant with the XPG3 base level and the XTI option of XPG3.

Except when those features conflict with POSIX or XPG3, OSF/1 will also comply with SVID Issue 2; supported extensions will include shared memory and semaphores. 4.3 BSD functionality will also be supported, except where such functionality is hardware dependent or conflicts with POSIX or XPG3.

The kernel itself will be written in ANSI X3J11 C.

2.5.2 File Systems

OSF/1 should suport both System V style filesystems and 4.3 style filesystems. It will also support an enhanced file system type which will provide disk mirroring, journaling and the ability for a single logical volume to span multiple physical volumes.

2.5.3 Networking

OSF/1 will support the 4.3 BSD *socket* interface, *streams* (as defined by SVID Issue 2), *tcp/ip*, *uucp*, and the XPG3 XTI interface (to the extent that it is stable).

2.5.4 Program Management

As indicated above, the kernel itself will be written in ANSI X3J11 C. OSF/1 will include an ANSI C compiler capable of providing XCOFF object format modules and will include support for shared libraries with dynamic linking. Direct support for other languages is not planned for the first release of OSF/1, but, with strong input from DECUS (again), the OSF recognizes the need to take a leadership position in defining a common calling standard between applications languages.

2.5.5 Kernel Features

OSF/1 will include modifications to the kernel to support preemptive scheduling and interrupt handlers. In addition, POSIX 1003.4 timer interfaces will be supported (to the extent that the standard is available). Although there has been

significant discussion of multi- processor support issues, such support will, most likely, not be available in the first release. However, the architecture is intended to create a framework which will support such modifications in the future.

2.5.6 Terminal Handling

Terminal independent screen handling will be supported via the terminfo interface; the termcap data base will also be provided for backwards compatibility.

2.5.7 Native Language Support

Due to the international nature of the OSF membership, the topic of native language support was much discussed. Full 8-bit character support will be provided for European languages, with mechanisms for multi-byte character support (e.g. for Asian languages) still under discussion. OSF/1 is intended to provide a basis for localization (by systems vendors), with national language calls supported as part of XPG3 compliance.

2.5.8 Security

The first release of OSF/1 should be certifiable at the C2 security level. Since the US Department of Defense certifies only complete systems, the OSF will not directly apply for this certification. Access control lists and (system administrator) configurable audit trails will be supported.

2.5.9 System Administration

It is highly desirable that OSF/1 provide more advanced system administration. Discussion was held regarding which mechanisms should be addressed in the first release of OSF/1. It was generally agreed that at a minimum, disk quotas, a generalized queue management facility, and the ability to configure system facilities while the system is running should be provided.

2.5.10 Validation Suites

Each release of OSF/1 will be tested against the following validation suites:

Monte Carlo OSF Meeting Trip
Report *continued...*

PCTS for POSIX 1003.1 Conformance

PCTS for POSIX 1003.2 Conformance

VSX Verification Suite

SVVS Release 2

Perennial BSD Validation Suite

2.5.11 Documentation

There are no plans to produce a Kernel Environment Specification (equivalent to the AES) with the first release of OSF/1 since there are no formal standards for intra-kernel interfaces nor are the kernel interfaces sufficiently stable to merit such a specification at this time.

Documentation for the OSC is scheduled to include the following documents:

    a. Design of the OSF Operating System
    Component

    b. OSF Operating System Component
    Programmer's Reference

    c. OSF Operating System Component
    Kernel Extension Guide

    d. OSF Operating System Component
    Kernel Porting Guide

Preliminary versions of a., b. & c., above should be available with the release of the *Vendor Kit* for OSF. With release of the release of the *Application Developer's* kit, a. & b. should be technically complete, with a more advanced version of c. available, as well as a preliminary version of d. All documents will be technically complete with release of the *University Kit,* with final versions available at FCS.

(continued in next issue...)

LAT-Telnet Problems *by Jim Karsten, ULTRIX csse*

There is an error in the *Guide to Ethernet Communication Servers* in Volume3, System Management.

On page 5-2, at top of page, it shows the following to get the LAT/Telnet gateway up:

    # lcp -v HOSTNAME\
        -V "HOSTNAME lat service"\
        -v telnet:/dev/tty20,/dev/tty21/,/dev/tty22\
        -V "lat/telent gateway service"

This does not work. The manual page for *lcp* gives the following command syntax (under the -v and -V options) which *does* work:

    # /etc/lcp -v microv -v SERV1:dev/tty15,dev/tty16\
        -V "Ultrix LAT Service" -V "service 1"

Advisory: If you issue the following command to set up the LAT-Telnet gateway on a DECstation 3100

    lcp -v hostname -v Telnet:/dev/tty15,/dev/tty16 -V
        "ULTRIX LAT SERVICE"\
        -V "LAT/TELNET GATEWAY"

you get the error message

    lcp -v: non LAT device

This is due to a problem in the lcp code.

SPR's have been submitted on both these problems.

CERT Internet Security Advisory *by Kenneth Van Wyk, Carnegie-Mellon University, Software Engineering Institute*

Many computers connected to the Internet have recently experienced unauthorized system activity. Investigation shows that the activity has occurred for several months and is spreading. Several UNIX computers have had their *telnet* programs illicitly replaced with versions of *telnet* which log outgoing login sessions (*including usernames and passwords to remote systems*). It appears that access has been gained to many of the machines which have appeared in some of these session logs. (As a first step, frequent *telnet* users should change their passwords **immediately**.)

While there is no cause for panic, there are a number of things that system administrators can do to detect whether the security on their machines has been compromised using this approach and to tighten security on their systems where necessary.

At a minimum, all UNIX site administrators should do the following:

- Test *telnet* for unauthorized changes by using the UNIX *strings* command to search for path/filenames of possible log files. Affected sites have noticed that their *telnet* programs were logging information in user accounts under directory names such as ... and **.mail**.

In general, we suggest that site administrators be attentive to configuration management issues. These include the following:

- Test authenticity of critical programs - Any program with access to the network (e.g., the TCP/IP suite) or with access to usernames and passwords should be periodically tested for unauthorized changes. Such a test can be done by comparing checksums of on-line copies of these programs to checksums of original copies. (Checksums can be calculated with the UNIX *sum* command.) Alternatively, these programs can be periodically reloaded from original tapes.

- Privileged programs - Programs that grant

privileges to users (e.g., setuid root programs/shells in UNIX) can be exploited to gain unrestricted access to systems. System administrators should watch for such programs being placed in places such as */tmp* and */usr/tmp* (on UNIX systems). A common malicious practice is to place a setuid shell (*sh* or *csh*) in the */tmp* directory, thus creating a *back door* whereby any user can gain privileged system access.

- Monitor system logs - System access logs should be periodically scanned (e.g., via UNIX *last* command) for suspicious or unlikely system activity.

- Terminal servers - Terminal servers with unrestricted network access (that is, terminal servers which allow users to connect to and from any system on the Internet) are frequently used to camouflage network connections, making it difficult to track unauthorized activity. Most popular terminal servers can be configured to restrict network access to and from local hosts.

- Passwords - Guest accounts and accounts with trivial passwords (e.g., username=password, password=none) are common targets. System administrators should make sure that all accounts are password protected and encourage users to use acceptable passwords as well as to change their passwords periodically, as a general practice. For more information on passwords, see **Federal Information Processing Standard Publication (FIPS PUB) 112,** available from the National Technical Information Service, U.S. Department of Commerce, Springfield, VA 22161.

- Anonymous file transfer - Unrestricted file transfer access to a system can be exploited to obtain sensitive files such as the UNIX */etc/passwd* file. If used, TFTP (Trivial File Transfer Protocol - which requires no username/password authentication) should always be configured to run as

CERT Internet Security Advisory
continued...

> a non-privileged user and *chroot* to a file structure where the remote user cannot transfer the system */etc/passwd* file. Anonymous FTP, too, should not allow the remote user to access this file, or any other critical system file. Configuring these facilities to *chroot* limits file access to a localized directory structure.
>
> • Apply fixes - Many of the old "holes" in UNIX have been closed. Check with your vendor and install all of the latest fixes.

If system administrators do discover any unauthorized system activity, they are urged to contact the Computer Emergency Response Team (CERT). CERT operates a 24-hour hotline at 412/268-7090.
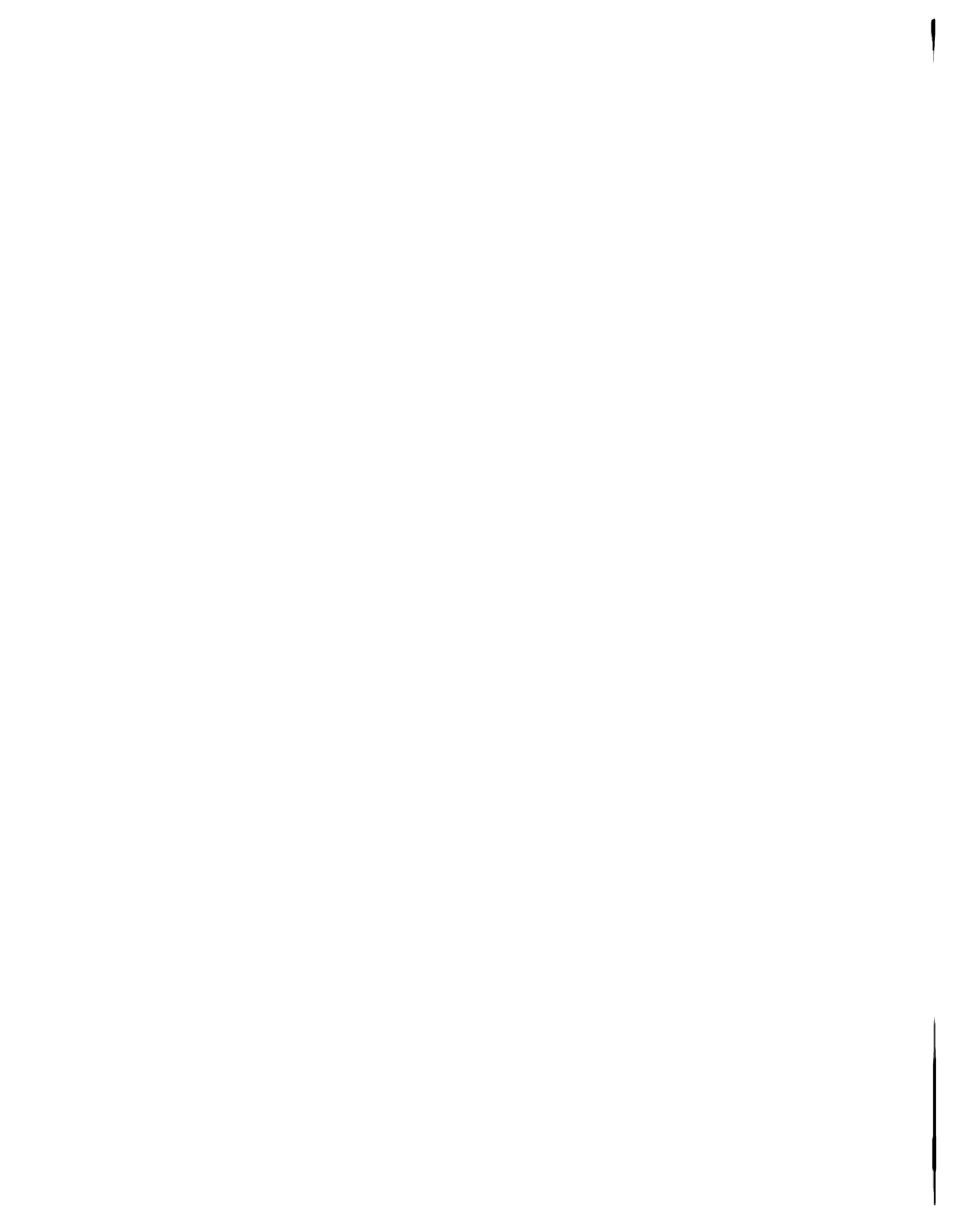
NEWSLETTER OF THE VAX SYSTEMS SIG

Pageswapper

Our Mascot
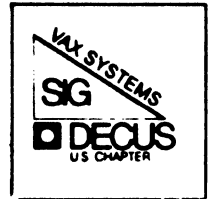
TABLE OF CONTENTS   VOLUME 10 NUMBER 12

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.

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 test is preferred.

Send your contributions to:

David K. Santistevan
Western Data Technologies
5270 Fox Street
P. O. Box 5542
Denver, CO 80217

Submissions may also be made electronically via DCS to KINGS.

DATA STRUCTURES IN DCL

A. Pirnar
ITT-Publitec
Hoekenrode 1 / 1102 BR Amsterdam Z. O. / Netherlands

ABSTRACT

The DCL interpreted programming language can be considered a
general purpose procedural programming language for many purposes.
However unlike most conventional programming languages it does not
have any mechanism for using structures of data. While such classical
languages as Fortran or Basic will cater for at least n-dimensional
arrays either as static or dynamic variables, and more modern versions
of these as well as originally structured languages such as Pascal or
C will allow declaration and usage of user defined data structures,
DCL really has only one 'variable' concept : The symbol.   The
following article outlines a method for systematically implementing
generalized data structures in DCL by extending the interpretation of
a symbol in DCL. The subject is introduced by a problem, leading to
the implementation, evolving through various stages as experienced by
the author.

INTRODUCTION

Many interactive VMS utilities are most easily implemented in DCL thanks to the efficient interpreter and the availability of a wide range of functions in the form of lexicals. Coupled with the short cycle involved between editing and running a DCL procedure, it becomes unnecessary to use a conventional programming language making system run-time calls etc. to write simple VMS utilities, unless developing a large and/or complex enough system or unless performance requirements are overwhelming.

In DCL a variable is called a symbol. The DCL interpreter allocates memory for symbols dynamically as they come into use. The type of a symbol is determined each time an assignment is made to that symbol. This can be either "string" or "integer". A symbol name consists of any combination of alphanumeric characters and the characters "_" and "$" up to a length of 255 characters provided the first character is alphabetic. For most reasonable DCL applications there is no practical limit to the number of symbols one may use.

DCL allows the programmer to store information in other 'variable' spaces as well. For example logical names contain strings and may be used for purposes similar to variables. However the mechanisms involved in translating logical names and operating on them do not naturally and efficiently fit our conventional expectations from a variable mechanism whereas symbols do. In addition, the overheads incurred in terms of performance as well as limitations imposed by a finite dynamic memory space in which all logicals have to exist, make logicals a poor choice as general purpose variables.

An alternative is to write into and read from the users memory space directly. One difficulty with this approach currently is that though it is possible to DEPOSIT into memory, DCL does not provide any direct way to EXAMINE the contents of memory locations into symbols. (If any of the readers of this article know of a single command to do this, please let me know). A considerable amount of overhead will be incurred to access structures "the real way" in raw memory possibly requiring conversions which can be quite inefficient in DCL. Data types will have to be decided upon and enforced by the programmer. In addition, making use of raw memory means that the programmer has to develop a memory management policy. Protecting the application data from other applications as well as the reverse becomes an issue. More importantly, programming data structures onto raw memory is a sensitive job even if we discard the complications introduced by even the simpler memory management schemes. Though this is done routinely by all compilers and interpreters featuring data structures, one is at the point where it is necessary to decide whether the application warrants such basic development and still be in DCL. It seems that a higher level access to structures of data is desirable.

Another alternative is to partition a symbols contents into fields. For example given the symbol A="1234567890" it is possible to view it as a one dimensional array of ten elements each of unity length. Simple substringing operations based on the lexical function F$EXTRACT can be used to read individual elements. Writing an individual element is also possible using offset arguments in assignments though the calculations of the offsets become non-trivial especially with more complicated structures. The most important shortcoming of this approach however is that the data structures are static. The size limit imposed for a symbol by DCL is a primary constraint. Furthermore it becomes quickly cumbersome when more complex or dynamic data structures are to be implemented.

Yet another alternative is to implement all data structures in files. This may seem the best choice if the amount of data is large. However opening files is unacceptably slow if interactive command performance is required. An implementation in memory is really what is wanted.

STRUCTURES OF SYMBOLS

Is there then a way in which data structures can naturally be implemented in DCL? Some time ago, the author was confronted with a DCL problem which seemed to have its simplest solution in the structure of a stack. The problem was that of keeping track of a users 'dives' into a directory structure, allowing the user to 'resurface' with ease. Later on the data structure was modified to a one dimensional array or table to allow more general movements regardless of 'depth' within any directory structure. Still later, the data structure was evolved into one linear table whose fields pointed to a tree structure capable of implicitly containing the relationships between the directories in a directory structure.

A stack is a data structure which can be visualized as a one dimensional array and a pointer called the stack pointer. Generally the stack is bounded at all times, having a finite number of elements of one type, and an index or stack pointer which at any time points to the current entry. The stack pointer may also point to the next available entry position this choice being arbitrary so long as the programmer knows where it points to. The operations on the stack are to PUSH or POP an element onto or from the stack.

Let us assume that we have an initialized stack S. Let us assume that S is actually a one dimensional array addressable in the form S(i) where i ranges from 0 up without any practical bound. Let us also say that the stack pointer called SP is equal to zero when the stack is initialized. SP=0 means that the next available entry position onto the stack is position 0. To push an element, say the string "data0", onto the stack or to pop it off the stack into a symbol DATA0 the following code segments would serve :

```
PUSH:  S(SP) = "data0"        POP:  SP = SP - 1
       SP = SP + 1                  DATA0 = S(SP)
```

```
$!-----------------------------------------------------GO_0.COM---A PIRNAR---
$GO:            vfy = 'f$verify(0)'
$GO_0:          if (f$type(go_flag).nes."") then goto GO_1
$GO_01:         go_sp    == 0
$               go_flag  == 1
$!
$GO_1:          if (go_flag.nes.1) then goto GO_01
$               on error then goto GO_EX_DEF
$               go_from = f$environment("default")
$               go_s'go_sp == go_from              ! FIX TOP OF STACK IN CASE
$!                                                   A NON-GO MOVE WAS MADE
$               if (P1.eqs. "") then goto GO_EX_DEF
$               if (P1.eqs."<") then goto GO_BACK
$               if (P1.eqs."S") then goto GO_SHOW
$               if (P1.eqs."N") then goto GO_NEW
$!
$GO_NEW_PLACE:  go_to = P1
$               set def 'go_to
$               go_to = f$environment("default")
$               if (go_from.eqs.go_to) then goto GO_EX_DEF
$               go_sp == go_sp + 1                              ! PUSH HERE
$               goto GO_EX_SHW
$!
$GO_BACK:       if (go_sp.eq.0) then goto GO_EX_DEF
$               go_sp == go_sp - 1                              ! POP HERE
$               go_to = go_s'go_sp          ! GET DIRECTORY SPEC FROM STACK
$               set def 'go_to              ! MOVE TO THAT DIRECTORY SPEC
$               goto GO_EX_SHW
$!
$GO_NEW:        go_flag == 0
$               goto GO_EX_DEF
$!
$GO_SHOW:       if (go_sp.eq.0) then goto GO_EX_DEF
$               i=0
$GO_SHOW_LOOP:  go_temp = ""
$               if (i.eq.go_sp) then go_temp = ""
$               go_lp = "_ "
$               write sys$output i,go_lp,go_temp,go_s'i,""
$               i=i+1
$               if (i.le.go_sp) then goto GO_SHOW_LOOP
$               goto GO_EX_SYS
$!
$GO_EX_SHW:     write sys$output go_from," ---> ",go_to
$               goto GO_EX_SYS
$GO_EX_DEF:     sh def
$GO_EX_SYS:     if (vfy.eq.1) then set verify
```

If the two operations PUSH and POP were executed consecutively, we would end with the symbol DATA0 containing the string "data0". However the above code segments are not acceptable in DCL because parenthesized array-like expressions are not allowed for symbols to denote variables. But that obstacle can easily be overcome by substituting some other acceptable characters for the parenthesis. Suppose the "_" character is from now on to be used instead of "(". Let us also take the liberty to simplify the syntax used to denote elements of an array by not requiring a closing parenthesis. The only remaining obstacle is now that in the above expressions we have the form S_SP. What we really mean though is, S_0, S_1, etc. depending on the value of SP, and we would like this to be resolved at the time the statement is being executed. This can be easily done by forcing symbol substitution within expressions using the DCL operator for this purpose, the single quote, "'". It is not necessary to enclose the symbol to be forced into substitution when the symbol name is clearly delimited to form a token. So the closing "'" is ignored where possible. Hence the new versions of the PUSH and POP code segments are:

```
PUSH:  S_'SP = "data0"          POP: SP = SP - 1
       SP = SP + 1                   DATA0 = S_'SP
```

Using the above principle it is possible to write a short procedure which will keep track of a users movements into directories and allow the user to retrace the path backward. Various design decisions are : The procedure will accept commands as single letters passed through the parameter P1. Single letter directory specifications (logicals) will not be allowed unless a colon is appended. If P1 is not a command it will be assumed to be a directory specification. If P1 is omitted or in erroneous situations the procedure will just do a "show default". In this example we will assume that the directory specifications typed by the user are always correct. A command to initialize the stack (N for NEW) will be provided. A command to show the stack (S for SHOW) will be provided. Let us call the procedure GO_0.COM (GO version 0). Also please note that no use is made of IF-THEN-ELSE constructs which are new to VMS in version 5.0 and were not available at the time these ideas were conceived.

NOTE TO PAGE MAKEUP

SEE PROGRAM GO_0.COM for placing near this location

Briefly tracing the procedure : The first statement is a standard line to shut off verify mode while saving the verify status to be able to reset it when exiting from the procedure. Next the procedures status flag named "go_flag" is checked for existence. If it doesn't exist, initializations must be done. The initializations being in the segment headed by the label GO_01. Here the stack

pointer "go_sp" is initialized and the procedure status flag is set to indicate that the procedure is up and running. There is then another check for the case where the procedure status flag was found to exist but still needs to be verified. This is needed to provide for the resetting (NEW) facility as well as to approximately verify that the procedures environment is incorrupt. Next the default error condition is set to be an exit from the procedure via a "show default" command.

In the following two lines the procedure determines the current default directory of the user and patches the stack with that specification. This is incorporated so as to avoid a discrepancy between where the procedure 'thinks' the user is by looking at the stacks current entry versus where the user actually might be. Different locations are possible because the user may have used other directory changing commands such as a simple "set def..." in the time between two calls to the procedure.

At this stage the procedure is in control of a stack whose integrity is quite reliable. Next the possible commands are considered in a rather brute force fashion. The reader can suggest elegant methods for implementing an n-way case clause such as this one. (Hint: The production version has all but the no parameter case decoded in one short line of DCL. See page 5-15 of the bibliography for an example)

The rest of the modules in the procedure implement the commands. GO_NEW_PLACE and GO_BACK are the labels at the tops of the segments where the stack is manipulated. The stack consists of the family of symbols having the form "go_si" where i is the index. The stack pointer is the symbol "go_sp". Notice that an even more compact symbolism is used here. What would have conventionally be written as GO_S(i) has been reduced to GO_Si by getting rid of parenthesis completely. Other symbolisms are obviously possible, as they may aid the programmer. However it is a good idea not to have symbol names unnecessarily long as they take up space and could eventually require an increase of the SYSGEN parameter CLISYMTBL or worse a change in the symbol naming convention used thus far. The second part of this article will show a more developed symbolism to represent more complex data structures.

The DCL purist might suggest that the two lines in the GO_BACK segment where the last directory spec is read off the top of the stack and a set def is executed could be combined into a single line :
$ set def &go_s'go_sp

This is true. The "&" and the "'" both force symbol substitutions but they are processed at different times. The DCL interpreter makes two passes over each command line. All "'" substitutions are done in the first pass while all "&" substitutions are done in the second pass. Therefore by combining the two substitutions, an 'indirect' addressing mode is possible. In this procedure however, the inefficient two lines have been used to allow the symbol "go_to" to be defined for later use in the line labelled GO_EX_SHW.

In general it is an advisable method to incorporate as many symbol substitutions into one line as possible as this will lead to compacter and faster procedures. Some of the speed gain will be because less statements are executed and the rest of the speed gain will be because of lesser penalties for GOTO statements as the DCL interpreter is forced to scan fewer lines to locate a targeted label. This is why it is good to move all the comment lines in a procedure to the end. Unfortunately the DCL interpreter does not keep a dynamic list of labels encountered during the execution of a procedure. Though it would add overhead to the DCL interpreter in general, it would also allow much faster execution of loops. The tradeoff seems very worthwhile in many cases. Maybe a mode could be added to let the programmer decide if the label-learning feature is desired. In any case, further discussion of this point is beyond the scope of this article.

Forcing symbol substitutions to form other symbols in expressions is a useful method to implement data structures in DCL. It is like having the final form of the statement formed at execution time. The same idea can be used to implement other types of 'self modifying' code. Admittedly self modifying code is generally undesirable as it easily becomes indecipherable. There are also more fundamental arguments against it in conventional environments and languages such as the difficulty of implementing reentrant code. However it is possible to implement such code in a structured way, avoiding the commonly associated pitfalls but gaining considerable programming power. Strictly speaking, the symbol substitution method does not qualify as self modifying code because the source code is the same each time DCL tries to interpret it. It is maybe better to visualize the symbol substitution method as a macro expansion or a syntactical alternative to the parenthesis containing expressions commonly used to encode data structures.

Continuing with the directory navigation problem, the next step is to give the user access in any sequence to a directory in the table kept by the application. In other words the user should see the table as a table rather than a stack in the original implementation.

Having visualized the stack as a special and operationally restricted case of a one dimensional array, it is quite easy to implement an unrestricted one dimensional array using the method outlined so far. The only addition is that the user should now be allowed to enter a command specifying the index number of a directory in the array (table) of directories. One new symbol is a symbol ("go_sp_max" below) to indicate the size of the table at any time. Although the structure is no longer a stack, the old symbol names are used for compatibility. Thus the table pointer is "go_sp" which is no longer serving as just a stack pointer. A modification must be made to the command decoding section for the new capability. The new capability must check that the user requests a valid table entry, and this is done in the first line of the module "GO_TO_N". The GO_NEW_PLACE module must also be modified to update the table size as it grows. The skeleton code required is as follows :

```
$ if (f$type(P1).eqs."INTEGER") then goto GO_TO_N ! addition to command decoding
.
.
.
$GO_NEW_PLACE:  go_to = P1
$               set def 'go_to
$               go_to = f$environment("default")
$               if (go_from.eqs.go_to) then goto GO_EX_DEF
$               go_sp == go_sp + 1                          ! PUSH HERE
$               if (go_sp.gt.go_sp_max) then go_sp_max == go_sp
$               goto GO_EX_SHW
.
.
.
$GO_TO_N:       if ((P1.lt.0).OR.(P1.gt.go_sp_max).OR.(P1.eq.go_sp)) -
                    then goto GO_EX_DEF
$               go_sp == P1
$               go_to = &go_t_'go_sp
$               set def 'go_to
$               goto GO_EX_SHW
```

A host of new commands become possible, allowing relative movements within the table in backward or forward directions. The functionality of the application is fundamentally changed. It is now desired that the application capture newly visited directories into its table of known directories by either appending inserting or overwriting to the table. This will depend on where in the table the user is positioned and whether append/insert/overwrite modes are

available. In the production version, the design decision to provide
only append/overwrite modes was made. The justification is that
insertion to a table (when implemented as a one dimensional array) is
very slow because one has to shift down all the entries following the
insertion by one position each time an insertion is made. For the
same reason deletions are not handled naturally either.

As is the case with most engineering problems, as soon as a
solution is found, the problem is expanded or restated to be more
general and more challenging. How can we deal with whole directory
structures? Can there be commands to enable seeing whole directory
structures in the table and manipulate these structures? The
insert/delete problem is now unmanageable. What do you do if a
directory in the table is deleted? Should its subdirectories be
deleted? How do we know they were subdirectories anyway? How do we
insert the subdirectories in the first place? Even if we take all the
brute force approaches available, it is clear that the performance
will be unacceptable. But surely there must be better suited data
structures to enable us to do some really nice operations!

TREES OF SYMBOLS

A tree is possibly the most natural data structure known. Many
phenomena exhibit tree relationships between components. That is :
One component is related to several other components. Each of the
other components may in turn be related to still more components.
This is called a recursive definition and lends itself to recursive
programming easily. In our application loops are not allowed. That
is to say a directory may not have its parent as a subdirectory. A
reasonable expectation from all directory schemes currently popular.

The symbol substitution method can be extended to represent tree
structures in DCL. Let us take one node of the tree, or one directory
and define the family of variables (symbols) required as follows :

```
go_s...          symbol containing the full directory specification
go_s..._x        symbol containing a count of the subdirectories
```

The parts indicated as "..." are to be a coding scheme to
implicitly represent the relationship between a directory and its
subdirectories. This can be done as shown below for a parent
directory and its two subdirectories (or children).

```
go_s_1       == "disk9:[is12]"
go_s_1_x     == 2

go_s_1_1     == "disk9:[is12.etc]"
go_s_1_1_x   == 0

go_s_1_2     == "disk9:[is12.mai]"
go_s_1_2_x   == 0
```

As seen the subdirectories are represented in symbols of the same form as the parent directory with the addition of a 'tail' of the form "_n" where n is the ordering sequence of the subdirectories. The parent directory in the example has two subdirectories, therefore the eXpansion symbol with the tail "_x" is equal to 2. The two subdirectories in turn have no subdirectories. Also note that global symbol assignments are made to have the structure saved outside the scope of the procedure.

The symbol substitution method can be used to form these symbol names regularly as the variant parts are just numbers. This looks all very well, but it does not look legible. In the old version of the application a Show command was available with nice numbers down one side which could then be used to select a directory. We clearly can not expect a user to enter a command with parameters such as "1_2_5". It is best if the old format is kept, while internally the application does the bookkeeping with the free advantage of compatibility with the previous version. In other words the application should make a translation. Discarding the administrative symbols (those with tails) the following mapping scheme will do what is wanted.

```
go_t_0 == "go_s_1"          go_s_1    == "disk9:[is12]"
go_t_1 == "go_s_1_1"        go_s_1_1 == "disk9:[is12.etc]"
go_t_2 == "go_s_1_2"        go_s_1_2 == "disk9:[is12.mai]"
```

Now the user only sees and uses the indices "i" in the expressions of the form go_t_i. The actual directory specification is accessed by doing a double translation. The first one to give the symbol containing the specification and the second one to get at the specification itself. Writing a directory specification to the structure now requires two steps as well. This is a justifiable cost however when the gain in representational power is considered.

Insertions and deletions are now easier with the only requirement of renumbering the linear table entries of the form "go_t_i" each time an insertion or deletion is done. For example :

Inserting "disk9:[is12.etc.day]" the main symbols become

```
go_t_0 == "go_s_1"          go_s_1     == "disk9:[is12]"
go_t_1 == "go_s_1_1"        go_s_1_1   == "disk9:[is12.etc]"
go_t_2 == "go_s_1_1_1"      go_s_1_1_1 == "disk9:[is12.etc.day]"
go_t_3 == "go_s_1_2"        go_s_1_2   == "disk9:[is12.mai]"
```

with the auxiliary symbols

```
go_s_1__x       == 2
go_s_1_1__x     == 1
go_s_1_1_1__x   == 0
go_s_1_2__x     == 0
```

The recursive nature of the tree representation allows for
limitless expansion by inserting subdirectories with their
subdirectories in turn to any required depth. The converse
functionality, collapsing the representation of subdirectory
structures is possible as well, simply by setting the expansion
counter at the deepest directory to remain visible, to zero. An
interesting side effect is due to their being an expansion counter at
each level. When the topmost expansion counter is set to zero,
naturally all lower levels become inaccessible to the application.
However when the topmost level is expanded again, all lower levels
will become accessible as well, even if they were not expanded this
time. Lower levels inherit the expansion of the higher levels,
provided they have been expanded themselves at some past time. In the
application the option to expand or collapse to full depth or only one
level is given for consistency reasons.

Deletions are handled by setting a deletion flag which is the
familiar symbol with a tail of "__d". This 'logical' deletion rather
than physical deletion allows for an 'undeletion' option, and more
importantly for related functionalities to act on this information.

Various other functionalities have been provided in the final
application. The correctness of target directories in movements is
checked and single letter commands are checked for valid syntax. The
application can 'capture' the latest directory visited using other
commands such as "set def" if such interference is desired. User
friendly modes for enabling/disabling/directing the applications
outputs has been provided. Where usable, syntactical alternatives
have been incorporated for repeating or concatenating commands to form
command sequences. To give an idea of the latest version of the
application, the help screen is shown here.

```
! GO    P1     P2     P3        IF ya GOnna GO THEN you GO TO GO ....
!-------------------------------------------------------------------------
!        ""                     show def                         GO.COM V2.8
!        ...                    goto any valid directory           VMS V4.4+
!        <..    [n]             go back      (n * <..<) times [n=1]  (c)a pirnar
!        >..    [n]             go forward   (n * >..>) times [n=1]  DECUS VAX-361
!        \..    [n]             go up [-]    (n * \..\) times [n=1]
!        n                      go to nth directory in GO table
!        X                      go to last directory in table chronologically
!        R                      go to sys$login
!        S      [P]    [f]      show table & stati (P=write to f) [f= GO_S$: = GO_S.LIS]
!        N      [R]             reset GO table and defaults       (R=go to sys$login)
!        E      [k,.]  [*]      expand subdirs under entries k,.  [k=current] (*=all)
!        C      [k,.]  [*]      collapse  "       "  entries k,.  [k=current] (*=all)
!        D      [k,.]          delete entries k,. from table      [k=current]
!        U                      undelete latest entry deleted
!        Z      [k]             delete directory k from device     [k=current]
!        I      [0,1]           toggle or set interference mode   [enable]/disable
!        V      [0,1]           toggle or set placing mode        [find+append]/overwrite
!        T      [0,1]           toggle or set talking                 [talk]/silent
!        /..                    multiple command separator        (eg: "n r/e */s" , e/s)
!        H                      H E L P
!-------------------------------------------------------------------------
```

Finally one more useful programming technique involving symbol substitutions in DCL is worth mentioning here. When mode flags are used for selecting blocks of code, the IF-THEN-ELSE structure is commonly used where the language has that option. In DCL the IF-THEN-ELSE construct has not been available until V5, requiring repetition of the IF statement for each statement to be executed for a condition.

An alternative scheme is to have blocks of statements 'comment themselves out' when not to be executed. By choosing the value for the inactive mode of the flag to be "!" which is the comment character in DCL, and the active mode to be null, the following three blocks of code become equivalent.

(1) USING STRUCTURED IF-THEN :

```
$ IF (FLAG .NES. "!") THEN
$   statement_1
$   statement_2
$   statement_3
$ ENDIF
```

(2) NOT HAVING STRUCTURED IF-THEN OPTION :

```
$ IF (FLAG .NES. "!") THEN $   statement_1
$ IF (FLAG .NES. "!") THEN $   statement_2
$ IF (FLAG .NES. "!") THEN $   statement_3
```

(3) USING ACTIVE FLAG TO ENABLE/DISABLE LINES OF CODE :

```
$ 'FLAG statement_1
$ 'FLAG statement_2
$ 'FLAG statement_3
```

In case (3) above, when the value of FLAG is "", each line becomes a statement line for the DCL interpreter after both passes of symbol substitution have taken place and the interpreter is to actually interpret and execute the line. When FLAG is "!", the first pass of the interpreter will cause the line to appear commented out to subsequent interpretation by DCL and the lines will not be executed. The technique is also useful when single statements are to be executed depending on the value of a flag. It is in fact slightly more efficient than the equivalent IF statement.

As can be seen, the symbol substitution mechanisms in DCL can be used to enhance the syntax of DCL to cope with its shortcomings in data structures as well as other minor shortcomings. Although DCL is being enhanced in ways that may render the solutions presented here obsolete, the programming techniques may still be useful for special cases as well as the insight they give into DCL.

BIBLIOGRAPHY

1. "Guide to using VMS command procedures" (DEC order number AA-LA-11A-TE). This document is a quite complete source of information on how to best implement command procedures, showing many commonly used tricks. Well presented with some nice examples.

2. "VMS DCL Dictionary" (DEC order number AA-LA12A-TE). This is the definitive reference text for the DCL programmer.

3. "GO.COM, a flexible directory access utility" by A Pirnar, (DECUS software library VAX-361)

GO.COM: An Advanced "SET DEF" Utility

A. Pirnar
ITT-Publitec
Hoekenrode 1 / 1102 BR Amsterdam Z. O. / Netherlands

ABSTRACT

GO is a DCL utility for flexible directory access using a table
display of directories as the user navigates, and is available in two
versions. GO was developed originally as a stack to keep track of
'dives' into subdirectory structures. It made use of an older utility
called SD.COM to parse directory specifications. In fact the symbolic
command "SD" is still easier to use when invoking GO, because of the
keys being next to each other. Later it was developed as GO_1 into a
table of directories keeping track of the users movements, for future
quick access. More recently GO_2 was written making use of the new
recursive CALL command in VMS V4.4.

Both versions of GO list a command summary when invoked with the
parameter "H". When invoked with no parameters it is the same as "sh
def".

Both versions allow the user to : move to quasi legal directory
specs (e.g. missing square brackets), verify the specified directory
exists, move directly to any entry in the table, step backwards or
forwards in the table, move to the last entry in the table
chronologically, move directly to sys$login, move up to parent
directory, reset the table of directory entries, and show the table
with current and last entries visited.

In addition GO_2 allows the user to : expand all or one level of
subdirectories under table entries, collapse all or one level of
subdirectories under table entries, enable/disable capturing of non-GO
movements, toggle between overwrite and find/append modes when moving
to a directory, toggle talk/silent mode, make multiple moves in the
table in one command, enter multiple commands in one command line,
automatically define logicals "GO_n$" corresponding to table entries,
print the table, delete/undelete table entries, and delete directory
trees from disk.

What follows is a tour through a directory structure using GO. The tour
needn't be confined to one disk as is the case in the examples here. The
symbol "GO" has been globally defined at login to invoke the procedure.
After logging in, invoking GO with no parameters is equivalent to "sh def".

```
$ GO
  DISK9:[IS12]
```

Help can be obtained on the screen as follows :

```
$ GO H
!!  GO    P1      P2      P3                   IF ya GOnna GO THEN you GO TO GO ....
!------------------------------------------------------------------------------
!       " "                          show def                        GO.COM V2.8
!       ...                          goto any valid directory           VMS V4.4+
!       <..    [n]                   go back      (n * <..<) times  [n=1]   (c)a pirnar
!       >..    [n]                   go forward   (n * >..>) times  [n=1]  DECUS VAX-361
!       \..    [n]                   go up [-]    (n * \..\) times  [n=1]
!       n                           go to nth directory in GO table
!       X                           go to last directory in table chronologically
!       R                           go to sys$login
!       S      [P]    [f]           show table & stati (P=write to f) [f= GO_S$: = GO_S.LIS]
!       N      [R]                  reset GO table and defaults          (R=go to sys$login)
!       E      [k,.]  [*]           expand subdirs under entries k,.     [k=current] (*=all)
!       C      [k,.]  [*]           collapse  "       "      entries k,. [k=current] (*=all)
!       D      [k,.]                delete entries k,. from table        [k=current]
!       U                           undelete latest entry deleted
!       Z      [k]                  delete directory k from device       [k=current]
!       I      [0,1]                toggle or set interference mode      [enable]/disable
!       V      [0,1]                toggle or set placing mode      [find+append]/overwrite
!       T      [0,1]                toggle or set talking                   [talk]/silent
!       /..                         multiple command separator      (eg: "n r/e */s" , e/s)
!       H                           H E L P
!------------------------------------------------------------------------------
```

You may expand the subdirectories under the default directory :

```
$ GO E
  DISK9:[IS12]
```

Then to show the table :

```
$ GO S
 0+ DISK9:[IS12]
 1- DISK9:[IS12.BCK]
 2- DISK9:[IS12.DRS]
 3- DISK9:[IS12.ETC]
 4- DISK9:[IS12.LOG]
 5- DISK9:[IS12.MAI]
 6- DISK9:[IS12.PRG]
 7- DISK9:[IS12.SQU]
 8- DISK9:[IS12.UTL]
placing mode is find/append.
```

The user's current location is shown in reverse, (bold here). It is also
possible to direct a copy of the shown information to an ouput file with
the command "GO S P filename". If the filename parameter is not supplied,
GO will output the information to the logical GO_S$ which as a default is a
file named GO_S.LIS.

Moving to the directory defined in the logical MAIL$:

```
$ GO MAIL$
DISK9:[IS12] ---> DISK9:[IS12.MAI]
```

This movement might also have been effected by entering "GO 5" meaning "go

to fifth table entry". Alternatively "GO GO_5$" would do because go defines a standard set of logicals of the form "GO_n$" for directories visited through the table, using the table index as part of the logical name. This is especially useful when a long directory specification has to be referred to later on in other VMS contexts. Yet another way would be "GO > 5" or "GO >>>>>" meaning move down in table by 5 positions. "GO >> 5" would mean move down in table by 10 positions and would land the user at entry 1 as the table is 'wrapped' in such situations. Such multiple or repeated commands are also possible when moving up in the table ("<") or when moving up in the directory structure itself, i.e."set def [-]", with the "\" parameter.

Looking at the table shows that because GO was in the find/append mode the movement was effected to an already existing entry in the table. The previous location is indicated by the "+" next to the table entry index.

```
$ GO S
 0+ DISK9:[IS12]
 1- DISK9:[IS12.BCK]
 2- DISK9:[IS12.DRS]
 3- DISK9:[IS12.ETC]
 4- DISK9:[IS12.LOG]
 5- DISK9:[IS12.MAI]
 6- DISK9:[IS12.PRG]
 7- DISK9:[IS12.SQU]
 8- DISK9:[IS12.UTL]
placing mode is find/append.
```

Moving back to last location chronologically is done using the "X" parameter.

```
$ GO X
DISK9:[IS12.MAI] ---> DISK9:[IS12]
```

You may toggle the table writing mode with "V".

```
$ GO V
%GO-I-V1, placing mode is now overwrite
  DISK9:[IS12]
```

Now trying to move to MAIL$ in the overwrite mode and looking at the table :

```
$ GO MAIL$
DISK9:[IS12] ---> DISK9:[IS12.MAI]
```

```
$ GO S
 0+ DISK9:[IS12]
 1- DISK9:[IS12.MAI]
 2- DISK9:[IS12.DRS]
 3- DISK9:[IS12.ETC]
 4- DISK9:[IS12.LOG]
 5- DISK9:[IS12.MAI]
 6- DISK9:[IS12.PRG]
 7- DISK9:[IS12.SQU]
 8- DISK9:[IS12.UTL]
placing mode is overwrite.
```

The overwrite mode is useful for keeping track of all movements historically. The mode for placing table entries is also applied for captured directories if interference mode is enabled. If the user uses a series of 'foreign' commands such as "set def" to change the default directory, GO will have no way to know these movements and hence they will go unrecorded. However upon the next invocation of GO, it will see that the current default directory has changed. When interference mode is enabled, this most recent 'foreign' directory is captured, using the current placing mode to place it into the table.

Now we will go to the root directory and reset the table,

expand one level of subdirectories, and expand all subdirectories under the
second entry, and look at the resulting table.

```
$ GO R
DISK9:[IS12.MAI] ---> DISK9:[IS12]
$ GO N
  DISK9:[IS12]
$ GO S
  DISK9:[IS12]
$ GO E
  DISK9:[IS12]
$ GO E 2 *
  DISK9:[IS12]
```

```
$ GO S
 0+ DISK9:[IS12]
 1- DISK9:[IS12.BCK]
 2- DISK9:[IS12.DRS]
 3- DISK9:[IS12.DRS.CANVPREP]
 4- DISK9:[IS12.DRS.CANVPREP.GEN]
 5- DISK9:[IS12.DRS.CANVPREP.GGIDS]
 6- DISK9:[IS12.DRS.COBLIB]
 7- DISK9:[IS12.DRS.PROTO]
 8- DISK9:[IS12.DRS.UTL]
 9- DISK9:[IS12.ETC]
10- DISK9:[IS12.LOG]
11- DISK9:[IS12.MAI]
12- DISK9:[IS12.PRG]
13- DISK9:[IS12.SQU]
14- DISK9:[IS12.UTL]
placing mode is find/append.
```

Collapsing the second entry is like this.

```
$ GO C 2
   DISK9:[IS12]
$ GO S
 0+ DISK9:[IS12]
 1- DISK9:[IS12.BCK]
 2- DISK9:[IS12.DRS]
 3- DISK9:[IS12.ETC]
 4- DISK9:[IS12.LOG]
 5- DISK9:[IS12.MAI]
 6- DISK9:[IS12.PRG]
 7- DISK9:[IS12.SQU]
 8- DISK9:[IS12.UTL]
placing mode is find/append.
```

Now if we expand the second entry again, we see that all levels are found
to be expanded. This is because we didn't collapse all levels but just the
top level at entry 2 previously.

```
$ GO E 2
   DISK9:[IS12]
$ GO S
 0+ DISK9:[IS12]
 1- DISK9:[IS12.BCK]
 2- DISK9:[IS12.DRS]
 3- DISK9:[IS12.DRS.CANVPREP]
 4- DISK9:[IS12.DRS.CANVPREP.GEN]
 5- DISK9:[IS12.DRS.CANVPREP.GGIDS]
 6- DISK9:[IS12.DRS.COBLIB]
 7- DISK9:[IS12.DRS.PROTO]
 8- DISK9:[IS12.DRS.UTL]
 9- DISK9:[IS12.ETC]
10- DISK9:[IS12.LOG]
11- DISK9:[IS12.MAI]
12- DISK9:[IS12.PRG]
13- DISK9:[IS12.SQU]
14- DISK9:[IS12.UTL]
placing mode is find/append.
```

To collapse all levels under entry 2 would be as follows :

```
$ GO C 2 *
   DISK9:[IS12]
```

Then deleting entry 6 (DISK9:[IS12.PRG]) from the table and looking at the table reveals :

```
$ GO D 6
   DISK9:[IS12]
$ GO S
 0+ DISK9:[IS12]
 1- DISK9:[IS12.BCK]
 2- DISK9:[IS12.DRS]
 3- DISK9:[IS12.ETC]
 4- DISK9:[IS12.LOG]
 5- DISK9:[IS12.MAI]
 6- DISK9:[IS12.SQU]
 7- DISK9:[IS12.UTL]
placing mode is find/append.
```

It is possible to supply a list of table entries for processing when the Expand, Collapse or Delete commands are given. In these cases the list refers to table indices before the operations have started. For example the command "GO D 2,4" would delete the second and fourth entries from the table. Another more general method to 'submit' more than one command to GO is to use the multiple command separator "/". For example "GO E/S" will expand the current entry and show the table. In the event that one of the commands has a blank in it, it is necessary to enter a leading double quote for the parameter part of the command. Thus for example : GO "E */S . A trailing quote is optional.

Deleting an entry from the table does not delete any files or directories from disk or any other devices. To delete (zap) directories the "Z" parameter is used. Here is an example starting from a given directory structure :

```
$ GO S
 0+ DISK9:[IS12]
 1- DISK9:[IS12.UTL.SYS.GO]
 2- DISK9:[IS12.UTL.SYS.GO.A]
 3- DISK9:[IS12.UTL.SYS.GO.A.AA]
 4- DISK9:[IS12.UTL.SYS.GO.A.AA.AAA]
 5- DISK9:[IS12.UTL.SYS.GO.A.AA.AAB]
 6- DISK9:[IS12.UTL.SYS.GO.A.AA.AAC]
 7- DISK9:[IS12.UTL.SYS.GO.A.AB]
 8- DISK9:[IS12.UTL.SYS.GO.A.AC]
placing mode is find/append.
```

```
$ GO Z 2
%GO-W-ZAP1, EVERYTHING UNDER AND INCLUDING DISK9:[IS12.UTL.SYS.GO.A] WILL BE LOS
%GO-W-ZAP2, ARE YOU SURE                 (yes/[NO]) ? : yes
%GO-W-ZAP3, ARE YOU SURE YOU ARE SURE (yes/[NO]) ? : yes
%GO-W-ZAP4, CONFIRM BEFORE DELETING    ([YES]/no) ? :
%GO-W-ZAP5, DELETION WILL COMMENCE IN 5 SECONDS... (CTRL-C IF IN PANIC)
5
4
3
2
1
0
$1$DUA9:[IS12.UTL.SYS.GO.A.AA.AAA]AAA.A;1, delete? [N]:y
$1$DUA9:[IS12.UTL.SYS.GO.A.AA.AAB]BBB.B;1, delete? [N]:y
$1$DUA9:[IS12.UTL.SYS.GO.A.AA.AAC]CCC.C;1, delete? [N]:y
$1$DUA9:[IS12.UTL.SYS.GO.A.AA]AAA.DIR;1, delete? [N]:y
$1$DUA9:[IS12.UTL.SYS.GO.A.AA]AAB.DIR;1, delete? [N]:y
```

```
$1$DUA9:[IS12.UTL.SYS.GO.A.AA]AAC.DIR;1, delete? [N]:y
$1$DUA9:[IS12.UTL.SYS.GO.A]AA.DIR;1, delete? [N]:y
$1$DUA9:[IS12.UTL.SYS.GO.A]AB.DIR;1, delete? [N]:y
$1$DUA9:[IS12.UTL.SYS.GO.A]AC.DIR;1, delete? [N]:y
$1$DUA9:[IS12.UTL.SYS.GO]A.DIR;1, delete? [N]:y
  DISK9:[IS12.UTL.SYS.GO]

$ GO S
 0+ DISK9:[IS12]
 1- DISK9:[IS12.UTL.SYS.GO]
placing mode is find/append.
```

Though it is possible to undelete the most recent entry deleted from the table, clearly it is not possible to recover zapped files and directories, as VMS does not cater for such a contingency. Like all physical delete operations, zapping should be used with discretion.

GO has proven itself quite useful in some applications where the system functions have been distributed in a complex tree of directories which is too large to grasp manually.
It is also very useful when trying to visualize all directories of a large system which one is exploring.

DUAL-HOST CONFIGURATIONS FOR MicroVAX SYSTEMS

Susan Zaney
DTN: 291-7059
NKS1-2/F2

Noreen Piazza
DTN: 223-6649
MLO5-5/T20

Barbara Towne
DTN: 223-8060
PKO3-1/8C

```
=====================================================================
|                                                                   |
|   o   Dual-host configuration guidelines expanded                 |
|                                                                   |
|   o   Higher availability MicroVAX systems at low cost            |
|                                                                   |
|   o   Increased CPU power for applications that share storage     |
|                                                                   |
|   o   Investment protection for MicroVAX II systems               |
|                                                                   |
=====================================================================
```

WHAT IS DUAL-HOST?

Dual-host configurations offer customers higher availability and multiCPU performance without the premium price often associated with these features. This capability is made possible by VAXcluster software and the DSSI storage subsystem.

A dual-host configuration consists of two MicroVAX systems, each with RF-series Integrated Storage Elements (ISEs), a Digital Storage Systems Interconnect (DSSI) cable, VMS, DECnet, Ethernet and VAXcluster software.

Two common examples of dual-host usage are:

o As servers for a MicroVAX-based VAXcluster (NI-based)--Satellite nodes connected to the Ethernet and participating in the cluster are able to access any ISE through either member of the dual-host configuration. If one of these systems fails, the VAXcluster software will transparently switch the satellite nodes over to the other system. The satellite nodes continue to have complete access to the ISEs in either system. With a dual-host system, users can take advantage of the performance of two systems, as opposed to one simply being a hot standby system.

o As multiuser systems with terminal servers--When timesharing terminals are attached to dual-host systems by terminal servers, users logged in to a system that fails can simply log on to the other system to continue operation.

Dual-host support was announced for the MicroVAX 3300/3400 systems in the October 17, 1988 issue of Sales Update Vol. 20 No. 8, and for the MicroVAX 3500 and 3800 systems in Vol. 20 No. 19 dated April 3, 1989.
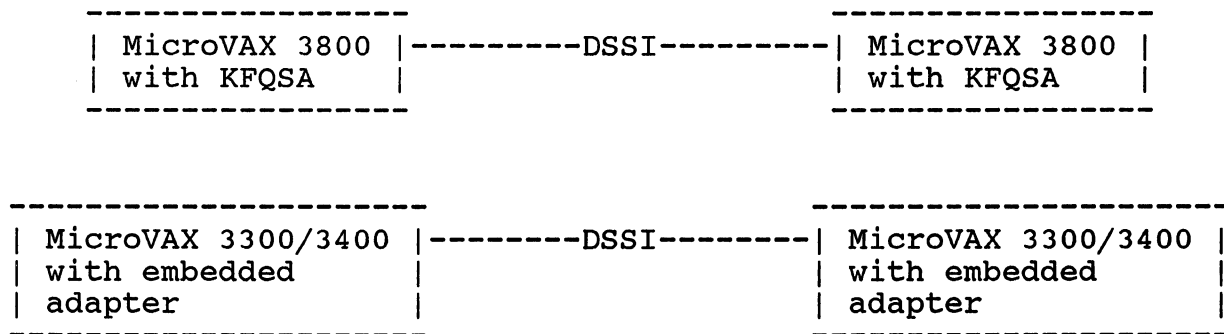
Until now, we have supported configurations limited only to two system enclosures. This announcement extends support to a maximum of two storage expansion boxes between two systems and to the installed base of MicroVAX II systems.

CONFIGURATION EXAMPLES AND GUIDELINES

The following examples depict the typical dual-host configurations that are now supported:

Configuration Example #1

```
          DSSI-based MicroVAX-to-DSSI-based MicroVAX


   ------------------                      ------------------
  | MicroVAX 3800    |---------DSSI--------| MicroVAX 3800    |
  | with KFQSA       |                     | with KFQSA       |
   ------------------                      ------------------


   ---------------------                   ---------------------
  | MicroVAX 3300/3400 |--------DSSI-------| MicroVAX 3300/3400 |
  | with embedded      |                   | with embedded      |
  | adapter            |                   | adapter            |
   ---------------------                   ---------------------
```
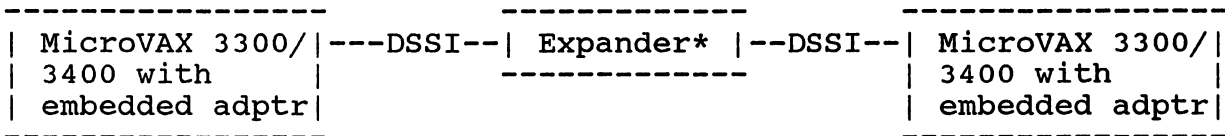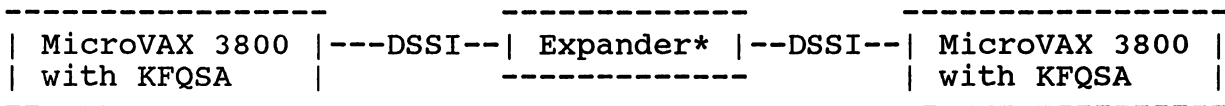
These configurations represent dual-hosted MicroVAX systems each with identical DSSI adapters and DSSI ISEs embedded in the system enclosures. If one of the systems fail, the second system continues to operate and users can access all of the ISEs in both systems. This is true as long as the first system is not powered off (i.e., to service the failure), of does not have a failure in the power supply.

The preceding configurations are available as standard systems that are pre-configured in a dual-host arrangement. By ordering these pre-configured standard systems, all of the pieces that are required to set up a dual-host come as one complete package.

For previously installed MicroVAX systems, simply order any of the standard DSSI-based MicroVAX 3300, 3400, 3500, or 3800 systems, and then add VAXcluster software and a DSSI cable (BC21M-09).

Configuration Example #2

DSSI-based MicroVAX-to-DSSI-based MicroVAX

```
-------------------          --------------          -----------------
| MicroVAX 3800   |---DSSI--| Expander* |--DSSI--| MicroVAX 3800   |
| with KFQSA      |          --------------       | with KFQSA      |
-------------------                                 -----------------


-------------------          --------------          -----------------
| MicroVAX 3300/  |---DSSI--| Expander* |--DSSI--| MicroVAX 3300/  |
| 3400 with       |          --------------       | 3400 with       |
| embedded adptr  |                                | embedded adptr  |
-------------------                                 -----------------
```

* In this application, it is recommended that the system disks reside in the expansion box so that they can continue to operate if one of the systems is powered off.

This configuration represents dual-hosted MicroVAX systems that are sharing one expansion box. This application of dual-host goes one step further by providing automatic fail-over should one of the systems power off, but it also protects the systems and data disks that are configured into the expansion box. That way, if one system is powered off, the system files and the database are still available.

To configure this system, simply order two MicroVAX systems, one storage expansion option (RF30C-DA or RF71B-DA; one DSSI cable is included with each), one additional DSSI cable (BC21M-09), and VAXcluster software, if it is not included with the systems.

Configuration Example #3

Non-DSSI-based MicroVAX-to-Non-DSSI-based MicroVAX

```
----------------          --------          --------          --------------
|MicroVAX II*|-DSSI-|Expndr|-DSSI-|Expndr|-DSSI-|MicroVAX II*|
----------------          --------          --------          --------------
```

* The hosts in this configuration can also be any other non-DSSI-
  based MicroVAX systems such as the RD- or RA-based MicroVAX 3500,
  3600, and 3900, in any combination.

This configuration represents dual-hosted MicroVAX systems that are
sharing two expansion enclosures, where all of the ISEs are
configured in the expansion boxes. This is the maximum allowable
configuration for dual-hosted non-DSSI-based MicroVAX systems.

To configure a dual-host with previously installed non-DSSI-based
MicroVAX systems, simply order two storage expansion options
(RF30C-*A, RF71B-*A; one DSSI cable (BC21M-09), DECnet, Ethernet,
and VAXcluster software, if it is not included with the system.

* Variant depends on system type. Refer to Ordering Information.

CONFIGURATION LIMITATIONS

The important configuration restriction to remember is that each
member of a dual-host configuration must use the same DSSI adapter.
MicroVAX 3300/3400 systems use the embedded adapter. MicroVAX II,
3500, 3600, 3800, and 3900 systems use the KFQSA adapter. A dual-
host configuration cannot include a MicroVAX 3300/3400 with an
embedded adapter, and a MicroVAX with a KFQSA.

There are also some restrictions with dual-hosting MicroVAX II
systems. Only one of them may be set to auto re-boot. Please
refer to the KFQSA Installation and User Manual for information.

DUAL-HOST ORDERING INFORMATION

To configure a dual-host system, you must have two Q-bus MicroVAX systems, two DSSI adapters of the same type, the proper cables between the KFQSA and the system enclosure, the appropriate number of DSSI cables, and the proper mix of software. You may mix and match accordingly.

DSSI System		Enclosure	Adapter
MicroVAX 3300		BA215	Embedded
MicroVAX 3400		BA213	Embedded
MicroVAX 3500	(later DSSI Version)	BA213	KFQSA (included)
MicroVAX 3800		BA213	KFQSA (included)

Non-DSSI System		Enclosure	Adapter Required
MicroVAX II		BA23	KFQSA-AA
		H9642	KFQSA-AA
		BA123	KFQSA-BA
MicroVAX 3500	(earlier non-DSSI Ver)	BA213	KFQSA-SG
MicroVAX 3600		Cabinet	KFQSA-SG
MicroVAX 3900		Cabinet	KFQSA-SG

Storage Expansion Boxes

Model No.	Storage Included	Cable Included	Adapter Included	Used On
RF30C-DA	2 RF30 ISEs	BC21M-09	NONE	
RF30C-AA	2 RF30 ISEs	BC21M-09	KFQSA-AA	
RF30C-BA	2 RF30 ISEs	BC21M-09	KFQSA-BA	
RF30C-CA	2 RF30 ISEs	BC21M-09	KFQSA-SG	
RF71B-DA	1 RF71 ISE	BC21M-09	NONE	DSSI Systems
RF71B-AA	1 RF71 ISE	BC21M-09	KFQSA-AA	MicroVAX II/BA23
RF71B-BA	1 RF71 ISE	BC21M-09	KFQSA-BA	MicroVAX II/BA123
RF71B-CA	1 RF71 ISE	BC21M-09	KFQSA-SG	Non-DSSI 3xxx systems

Additional Hardware

KFQSA-AA	KFQSA with cabinet kit to connect to BA23 and H9642*
KFQSA-BA	KFQSA with cabinet kit to connect to BA123*
KFQSA-SG	KFQSA with BA2xx-type I/O handle
BC21M-09	Cable to interconnect enclosure to enclosure

* Cabinet kit uses one size A (1X4-inch) panel insert

SOFTWARE LICENSE REQUIREMENTS

System   License

Server   File and application server license (one per system)
Timeshare   VMS 1-n user license (one per system)

All   LAVc license (one per system)
                DECnet end-node (one per dual-host configuration)
                DECnet full-function (one per dual-host
configuration)

FOR MORE INFORMATION

The following materials can assist you in configuring a dual-host
system:

KFQSA Adapter Installation and User Manual	EK-KFQSA-IN
MicroVAX/VAXserver 3300 System Technical Manual	EK-020AA-IS
MicroVAX/VAXserver 3400 System Technical Manual	EK-163AA-IS
MicroVAX Systems Dual Hosting Manual	EK-338AA-DH
MicroVAX 3800 System Technical Manual	EK-167AA-IS

For information on dual-host systems and dual-host performance,
refer to the MicroVAX Performance Summary (EE-C0191-41).

QUESTIONS AND ANSWERS

CAN MY CUSTOMERS DUAL-HOST A MicroVAX II OR AN RA-BASED MicroVAX
3600 OR 3900 SYSTEM WITH ANY OF THE DSSI-BASED MicroVAX SYSTEMS?

   Yes, the MicroVAX II, 3600 and 3900 systems can be dual-hosted
   with DSSI-based systems (such as the MicroVAX 3500 and 3800) by
   sharing an RF-based storage expansion box. Customers simply need
   to purchase a DSSI storage expansion option (RF30C-*A or RF71B-
   *A) and an additional DSSI cable (BC21M-09) in order to connect
   to any DSSI MicroVAX system.

CAN MY CUSTOMERS DUAL-HOST A MicroVAX 3300 OR 3400 SYSTEM WITH
OTHER MicroVAX SYSTEMS?

   No, dual-host configurations can only be created between MicroVAX
   systems that have identical DSSI adapters. So, MicroVAX
   3300/3400 systems with the embedded DSSI adapter can become a
   dual-host, and the MicroVAX II/3500/3800 systems with the KFQSA
   adapter module can become an dual-host.

CAN MY CUSTOMERS EMBED RF-SERIES ISEs IN THEIR MicroVAX II BA23 OR
BA123 ENCLOSURE?

No, however they can add DSSI functionality to their MicroVAX II
system by using RF-series ISEs in storage expansion enclosures.
These expanders come with the appropriate version of the KFQSA
DSSI adapter and DSSI cable so that customers can easily connect
to their system. The same goes for dual-hosting MicroVAX II
systems. This is achieved by using a DSSI storage expansion
enclosure and sharing it between two MicroVAX II systems.

WHAT VERSIONS OF VMS DO MY CUSTOMERS NEED IN ORDER TO IMPLEMENT
DUAL-HOSTING?

For MicroVAX 3300/3400 systems, VMS V5.0-2A or later is required.
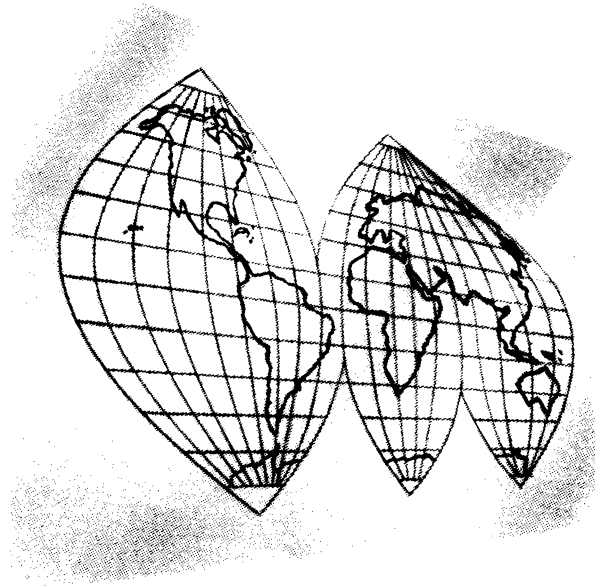For all others systems, VMS V5.1-1 is required.

CAN I CONFIGURE MORE THAN ONE KFQSA ADAPTER IN A SYSTEM?

No, this is not supported at this time.

The
DECUS LIBRARY

Software News

U.S. Chapter Edition

"Solving Your Everyday Problems"

NEW LIBRARY PROGRAMS AVAILABLE FOR THE VAX/VMS FAMILY OF COMPUTERS

DECUS NO: V00430 **TITLE:** Terminal Server/Port Display **Version:** X1-000, January 1989

Submitted by: Jim Duff

Operating System: VAX/VMS V4.7 **Source Language:** MACRO-32 **Keywords:** Networking, Terminal Management

Abstract: The program SHOW__PORT exploits an undocumented extension to a data structure defined in the I/O database for VMS. The extension contains the name of the port and server a user is logged into. The port and server names are those defined on a DECServer or MUXServer product that support the LAT protocol.

Port and server names under LAT can be very useful as the terminal name will not be the same on subsequent logins from the same terminal, whereas the port and server names will.

The program presently prints the information to SYS$OUTPUT. However, the program could readily be modified to be called as a subprogram or implemented as a user written system service.

Notes: Operating System VAX/VMS V4.0 or later is required. CMKRNL privilege is required in order to run this program.

Documentation may or may not be on media.

(Service Charge Code): 600' Magnetic Tape (MA) **Format:** VMS/BACKUP

DECUS NO: V00429 **TITLE:** Encryption Routine **Version:** 01-001, January 1988

Submitted by: Jim Duff

Operating System: VAX/VMS V4.7 **Source Language:** MACRO-32 **Keywords:** File Management, Security

Abstract: This program is a general purpose fast encryption routine that will perform "in-place" encryption on any type of file. Unlike most encryption programs there is no restriction on file type, record length, file size, or text verses data. This program will encrypt anything as long as there is enough virtual memory to load the file. A benefit of encrypting the file in memory is that the encryption is quite fast.

The program is designed to be envoked by DCL, and effectively replaces the ENCRYPT command supplied by Digital Equipment Corporation. However, the encryption algorithm could be easily extracted from the program and used as to perform in line encryption from a user written program.

The program has the same functionality as Digital Equipment Corporation's command, including the ability to encrypt a list of files in the one command.

Notes: Operating System VAX/VMS V4.0 or later is required.

Documentation not available.

Media (Service Charge Code): 600' Magnetic Tape (MA) **Format:** VMS/BACKUP

DECUS NO: V00428 **TITLE:** PQ Printer Utility **Version:** 1.1, April 1988

Submitted by: Mitchell Wolfe, Recording and Research Center, Denver, CO

Operating System: VAX/VMS V4.6 through V5.1 **Source Language:** DCL **Memory Required:** 73KB **Hardware Required:** LA100 printer **Keywords:** Utilities – VMS

Abstract: This package started out as a means to eliminate the fixed fifty line with a ten character left margin page format that all our company research papers and reports were printed by. It also provided a better, more flexible means of printing program generated reports.

The resulting command program combined the various units of the standard DCL print command — printer forms, print command qualifiers, and print queues — with the various units of the LA100 series of printers — eight different pitch sequences, adjustable page length sequences, and the near letter quality sequence, and it brings them all together under one easy-to-use command which prompts the user for each unit (see documentation for a better understanding). From the PQ Login Entry, commonly used PQ Utility commands can be tied to a command symbol in the users LOGIN.COM file for even easier, hassle free printing. It includes near-letter quality printing and adjustable page length (for use with mailing labels, rolodex cards, etc) along with an optional set of 15 printer forms.

Release notes are distributed with each order.

Notes: Requires access to sys$common: 000000.sysmgr to install.

Documentation available in hardcopy only.

Media (Service Charge Code): User's Manual (EA), 600' Magnetic Tape (MA) **Format:** VMS/BACKUP

DECUS NO: V00427 **TITLE:** TIMESHEET.COM **Version:** 1.0, June 1989

Submitted by: Thomas E. Chenault, U.S. Government, WSMR, NM

Operating System: VAX/VMS V4.4, V4.6, V5.0, V5.1 **Source Language:** DCL **Memory Required:** 512W **Software Required:** EDT Editor **Keywords:** Mail, Utilities – VMS

Abstract: TIMESHEET.COM was written to alleviate need for written timesheets, and for supervisors to constantly track down their employees' time. Employees must still sign for leave taken, and secretaries must still submit written records to the Finance and Accounting Department, however it does not apply to non-leave taking employees and their supervisors. It compiles a weekly timesheet that runs from Sunday through Saturday. It can be easily modified for any other timesheet configuration.

Restrictions: Two files must exist in each user's root directory, they are: STANDARD.TIMESHEET (a copy of your default working schedule) and LAST__SENT.TIMESHEET (a copy of the last timesheet sent).

Media (Service Charge Code): 600' Magnetic Tape (MA) **Format:** VMS/BACKUP

DECUS NO: V00426 TITLE: VAX to PPS Version: 1.5, June 1989

Submitted by: Michael Frederick, University of Houston, Houston, TX **Operating System:** MicroVMS V4.5 – 5.1, VAX/VMS V4.5 – 5.1 **Source Language:** BLISS-32, MACRO-32 **Memory Required:** 200KB **Software Required:** Honeywell PPS-II GCOS inbound processor **Hardware Required:** Honeywell PPS-II with off line input, nine track tape drive **Keywords:** Conversions, Utilities – Tape

Abstract: For large printing needs, Honeywell PPS II (Page Printing System, Model II) is capable of printing ninety pages per minute from rolls of paper stock. Variable form lengths, multiple distributions with separator pages, rotation, variable fonts, and preprinted form usage are all possible with this printing system. There are two programs in this software package. They are PPS and CB. The PPS program copies one or more VAX file(s) onto a PPS II-formatted tape. It also copies any PPS II control blocks as directed. The CB program will create

"custom" PPS II control blocks as separate VAX files.

Notes: MACRO-32 code included for BLISS-32 sections. (It is not necessary to have BLISS-32 files to build this code.)Object and executable files included are for VAX/VMS V5.X.

Assoc. Documentation: Honeywell manuals listed in documentation on media.

Media (Service Charge Code): 600' Magnetic Tape (MA)Format: VMS/BACKUP

DECUS NO: V00425 TITLE: LASERS, QUEUES and Other Fun Things Version: June 1989

Submitted by: Bob Armstrong, Algonquin College, Nepean, Ontario, Canada

Operating System: VAX/VMS V5.1 **Source Language:** C **Memory Required:** 300 Pages **Software Required:** VAX C Compiler **Keywords:** File Management, Utilities – VMS

Abstract: LASERS, QUEUES and Other Fun Things is a collection of useful utilities. Following is a brief summary of these utilities:

FileHeader	- A real UNDELETE utility, if the file can be recovered it will remark the blocks as allocated, fix the header and place the file within the directory.
	- A list header utility, list all headers associated with an fid and shows whether the extents are allocated, or free.
	- A patch header utility, not complete but will allow you to flip flags or patch fid's.
	- A real MOVEFILE utility, something like MV under UNIX, will move files (without deletion) to new directories on the same disk (works with wildcards as well).
Drawtree	- A super fast drawtree utility, uses RMS routines and written in C. Handles roots, and extra long directory names.

Laser	- As well as all other qualifiers on the PRINT command LASER supports/FONT. You define what fonts are available for what printers/lasers. LASER sets up the appropriate text modules from SYSDEVCTL.TLB file when the file is printed.
Operations	- Use SYSDEVCTL.TLB to program your printers, reduces operations setup time.
Symbionts	- A collection of single stream symbionts, to do character translation from Digital Equipment Corporation multinational to specific sequences for third party printers. Easily modifiable for other printers/lasers.
QUEUES across DECNET	- Link queues across a normal DECNET link. This will allow users to print on a remote machine without having NETMBX, everything appears as a local queue.

Media (Service Charge Code): User's Manual (EA), 600' Magnetic Tape (MC) Format: VMS/BACKUP

DECUS NO: V00424 TITLE: FLECS: FORTRAN Language with Extended Control Structures Version: 28, April 1989

Author: Terry Beyer, University of Oregon

Submitted by: G. Buffington, Dept Nat. Defense

Operating System: MS/DOS V3.2 **Source Language:** ASSEMBLER, FLECS **Memory Required:** 128KB **Software Required:** MS-FORTRAN **Keywords:** FORTRAN, Structured Languages/Programming

Abstract: FLECS is an extension of the FORTRAN language which provides control structures necessary to support recent concepts of structured programming. This version of FLECS has been modified to run on PC compatible workstations, running MS-DOS.

Notes: Operating System MS/DOS V3.2 or greater is required. This software package has to be transferred to an IBM PC compatible computer in order to operate it.

Restrictions: Indentation option not fully functional, occasionally inline comments not detected, /W option not valid.

Media (Service Charge Code): 600' Magnetic Tape (MA) Format: VMS/BACKUP

DECUS NO: V00423 TITLE: Kronos Version: 1.0, June 1989

Submitted by: Arthur E. Ragosta, US Army ARTA, Moffett Field, CA

Operating System: VAX/VMS V5.0-2 **Source Language:** DCL, FORTRAN 77, MACRO-32 **Memory Required:** 130KB **Keywords:** Scheduling, Security, System Management – VMS

Abstract: The KRONOS system provides an environment for scheduling the submission of batch jobs that is easier and more functional than the SUBMIT/AFTER command. A detached process is created which wakes up every hour to check a

database for jobs that should be run. Jobs may be scheduled to run at a given time, on a given day of the week, on a given day of the month, on a given weekday of the month, on a given day from the end of the month, every day, or every hour. A user-friendly interface program is provided to maintain the database. A variety of useful example jobs are provided to demonstrate the use of KRONOS; included are System Management, Security, and Performance Monitoring jobs.

Notes: Operating System VAX/VMS V4.0 or later is required.

Media (Service Charge Code): 600' Magnetic Tape (MA) Format: VMS/BACKUP

DECUS NO: VS0100 TITLE: X WINDOWS CLIENTS and DEMOS Version: 11, Release 3

Submitted by: Trevor Taylor, Microcomputer Technology, Queensland, Australia4034

Operating System: VAX/VMS V5.1 Source Language: C Memory Required: 55KB Software Required: DECWINDOWS (Included with VMS V5.1) Keywords: Editors, Tools – Software Development, DECWINDOWS

Abstract: This tape contains a large portion of the official X Window System (Version 11 Release 3) Distribution Kit from the Massachusetts Institute of Technology (M.I.T.).

The primary purpose of this re-mastering is to provide the VMS community with access to some of the more useful tools and demos as examples of X programming and for general use. The code has been converted to run under VMS DECwindows (except for a few clients which are for illustrative purposes only).

Included in the kit are the PostScript versions of the Xlib and X Toolkit Intrinsics manuals from MIT, as well as the C sources for Xlib, the Toolkit and the Athena Widget set.

A sampling of the applications is as follows:
MIT Clients — bitmap, xcalc, xclock, xdpyinfo, xev, xfd, xload, xlsfonts, xlswins, xmag, xmodmap, xprop, xsetroot, xwd, xwininfo, xwud.
Demos and Games — paint, qix, xcolors, xeyes, xfish, xgranite, xhanoi, xmille, xphoon, xsol.

Notes: Operating System VAX/VMS V5.1 for DECWINDOWS support is required.

Restrictions: This software is supplied "AS IS". Some bugs are known to exist.

Media (Service Charge Code): 2400' Magnetic Tape (PC) Format: VMS/BACKUP, TK50 Tape Cartridge (TC) **Format:** VMS/BACKUP

DECUS NO: VS0099 TITLE: TECO Collection Version: August 1989

Submitted by: Pete Siemsen, University of Southern California, Los Angeles, CA

Operating System: IAS, MS-DOS, RSTS/E, RSX-11M, RSX-11M-PLUS, RT-11, ULTRIX, VAX/VMS V4.7, V5.1 Source Language: C, MACRO-11, TECO Keywords: Editors, TECO

Abstract: This VMS directory tree contains a collection of TECO software from DECUS and other sources. Following is a brief summary of this collection:

[.DOC]	The newest manual for "Standard" TECO, May 1985. This manual is more current than what Digital Equipment Corporation distributes. Also, included are Version 39 and Version 40 release notes, describing all kinds of goodies in TECO11 and TECO32, like callable TECO.
[.EMACS11]	EMACS subset for TECO-11, Version 35 or higher. Submitted by Fred Fish.
[.LIDSTER]	MACROS and a documentation file that describes TECO initialization and how to customize. Submitted by Ken Lidster.
[.MACROS]	Best/latest versions of "classic" TECO MACROS from the rest of the collection.
[.RSTS]	TECO stuff from RSTS/E V9.5 contains 1982 sources of VTEDIT, SQU, etc. with some documentation. Submitted by Mark Derrick.
[.RSX...]	Everything relating to TECO found in the RSX SIG tapes.
[.SMITH]	MACROS for munging BASIC under RSTS, documentation for TECO initialization for RSTS and VMS, and VTEDIT with documentation. Submitted by Kelvin Smith.
[.SOFLIB]	TECO entries from the DECUS Library. VTEDIT for VAXTPU, video editors for Hewlett Packard and Tektronix terminals, an EMACS-like package for RSTS/E TECO-11, the distribution of TECO-11 V36, and more.
[.TECO11]	Source code for TECO-11 V36 (mixed mode for VMS).
[.TECO32_FOR_]	Native mode TECO32 released with VMS V5.0, V4 but built under V4 so it will run under V4.
[.TECOC]	TECO in C for VAX/VMS (almost UNIX and MS-DOS). Submitted by Pete Siemsen.
[.VMS...]	TECO software from a VMS SIG CDROM Disk, 1984 – 1987.
[.UTECO]	TECO in C (July 1989) for ULTRIX and SunOS. Submitted by: Matt Fichtenbaum.
[.YMILES]	TECO in C V1.04 (12 June 1988) for MS-DOS. Submitted by Ya'akov Miles.

Notes: Executable and/or object code is included. Sources for VMS TECO32 are not included.

Complete sources not included.

Media (Service Charge Code): 2400' Magnetic Tape (PC) Format: VMS/BACKUP, TK50 Tape Cartridge (TC) **Format:** VMS/BACKUP

NEW LIBRARY PROGRAM AVAILABLE
FOR THE
PDP-11 FAMILY OF COMPUTERS

DECUS NO: 110920 TITLE: RENUM 5.0: RT-11 FORTRAN IV Renumbering Program Version: June 1989

Submitted by: Digital Equipment Corporation

Operating System: RT-11 V5.4 **Source Language:** FORTRAN IV **Memory Required:** 14.5KW **Keywords:** FORTRAN, Tools – Applications Development

Abstract: This package replaces existing statement (label) numbers in a FORTRAN program with a series of sequential numbers. It includes support for OPEN/CLOSE statements, better handling of embedded spaces and tabs, case insensitive support for keywords, support for full five digit label numbers, better handling of continuation lines, more informative error messages. RENUM can renumber multiple subprograms in a single file with numbers unique to the file or numbers unique to each subprogram. RENUM is menu driven and could be run under the VAX/VMS operating system.

Notes: Operating System RT-11 V5.4 is required because file renaming calls require system dependent services.

Documentation not available.

Media (Service Charge Code): One RX01 Diskette (KA) **Format:** RT-11, 600' Magnetic Tape (MA) **Format:** RT-11

REVISIONS TO LIBRARY PROGRAMS

DECUS NO: VS0058 TITLE:TeX Collection Version: August 1989

Author: Various

Submitted by: M. Edward Nieland, Control Data Corporation **Operating System:** MS/DOS V3.1, Macintosh V6.1, UNIX, VAX/VMS V5.1-1 **Source Language:** C, DCL, MODULA, PASCAL, VAX FORTRAN **Hardware Required:** Drivers supplied for various printers and monitors **Keywords:** Text Formatting, LaTeX

Abstract: The TeX Collection is based on the TeX Files stored at SCORE.-STANFORD.EDU and are available via ANONYMOUS FTP, plus additional material collected from sources across the world. It is not necessary to have the compilers, since executable versions of most programs are included on the tape.

The TeX collection includes most source programs as well as object code. This package contains five major directory structures:

[.000__INSTRUCT]	Contains installation instructions.
[.TEX]	Contains all the material needed to get TeX up and running on your system. The material includes TeX, LaTeX, METAFONT, AmS-TeX, TeXsis, BIBTeX, PiC-TeX, DVI Drivers, RNOTOTEX, a spelling checker that understands TeX/LaTeX, and DVITOVDU.
[.TEX__FONTS]	All the fonts for TeX. Designed for a LN03.
[.TEX__SOURCE1]	
[.TEX__SOURCE2]	Contains the source code for everything in .TEX , plus additional material such as: UNIX Material, TeX and Screen Previewers for PCs, TeX and Previewer for Macintosh, GloTeX, previewers for VAX-Stations (Both VWS and DECWindows), LSE Templates for TeX/LaTeX and BIBTeX, Trip and Trap Tests for TeX and METAFONT, and additional DVI Driver material including PostScript LaTeX material for VMS.

The TeX Collection includes the other TeX related individual submissions from the DECUS Library. These submissions are:
. V00135DVI to VDU: A TeX Page Previewer Program
. V00171LaTeX Templates & Help files for LSE
. V00179DEPROC: LaTeX for the DECUS Proceedings
. V00294WEB Pack
. V00301DVIOUT – DVI Output Driver
. V00399DV12PS

The following output devices are supported:
. DEC LN03 (requires a RAM Cartridge) [DVI to LN03 or LNTex].
. DEC LN03 Plus (uses bitmaps) [DVIL3P].
. DEC LA75 [DVI175].
. PostScript (LPS40, Apple LaserWriter, LN03S) [DVIALW or DVI2PS].
. Hewlett Packard Laserjet [DVIJET].
. Hewlett Packard Laserjet Plus [DVIJEP].
. Cannon Engine Laserprinter [DVICAN].
. EPSON Printer [DVIEPS].
. Printronix Printer [DVIPRX].
. Okidata Pacemark 2410 (72 or 144 DPI) [DVIOKI].
. VT terminals, ReGIS Terminals, Tektronix Terminals [DVI to VDU].
. VAXStations running VWS [DVIDIS].
. DECWindows [XDVI].
. Version 3.10 BBN BitGraph Terminal [DVIBIT].
. Golden Dawn Golden Laser 100 printer [DVIGD].
. Imagen imPRESS-language laser printer family [DVIIMP].
. Apple Imagewriter 72 or 144 dpi printers [DVIM72 or DVIMAC].
. MPI Sprinter 72 dpi printer [DVIMPI].
. Toshiba P-1351 180 dpi printer [DVITOS].
. Generic Output [DVI2TTY].

The following items are new with this collection:
. DVI2TTY.
. OZTeX (TeX for the Macintosh).
. XDVI (DECWindows Previewer for VMS).
. DVIOUT.
. WEB Pack.
. DVI2PS for VMS.

Notes: The August 1989 DECUS TeX collection has updated the following items: TeX (now version 2.991), LaTeX & SLITeX (now version 2.09 {24 May 1989}), TeXsis, the LaTeX style collection, DVI to LN03, TeX for MS-DOS, and TeX for UNIX (includes TeXx 2.8.6 and the new SPIDERWEB). Executable and/or object code is included.

Changes and Improvements: Additional and updated material.

Assoc. Documentation: The TeX and LaTeX systems are described in two books, "LaTeX, a Document Preparation System", by Leslie Lamport, (ISBN 0-201-15790-X) and "The TeXbook" , by Donald Knuth, (ISBN-0-201-13448-9) and are available through Addison & Wesley Publishers. This is considered minimal documentation for the system. Donald Knuth's five volume set, "Computers and Typesetting", is highly recommended. The five volume set includes "The TeXBook" and "The METAFONTBook ". "The Joy of TeX" is the documentation for "AmS-TeX". The "PiCTeX Manual" will be needed if you wish to use "PiCTeX". These books should be purchased when you want to use the system properly. These books are not available through DECUS.

Complete sources not included.

Media (Service Charge Code): 2400' Magnetic Tapes (PD) **Format:** VMS/BACKUP, 2400' Magnetic Tape (SD) **Format:** VMS/BACKUP, TK50 Tape Cartridge (TD) **Format:** VMS/BACKUP

DECUS NO: V00378 **TITLE:** XMAIL: VAX/VMS Mail Utility Enhancements **Version:** 2.0, June 1989

Submitted by: Alan Mac Arthur, The Boeing Co., Seattle, WA

Operating System: MicroVMS V4.7, VAX/VMS V5.01, V5.1 **Source Language:** VAX FORTRAN **Software Required:** FORTRAN Compiler **Keywords:** Mail, Utilities – VMS

Abstract: The VAX/VMS Mail Utility has many powerful capabilities, however some desirable functions are not yet available. In particular, it is frequently useful to know if a message that you sent has been read, and to be able to accurately reset your new mail message count to the number of new mail messages that you actually have. System managers may also want to clean up the system mail file, to determine the number of new mail messages pending for a given user or all users, and to send a mail message to all users on the system. This program will perform these functions for both VMS Versions 4 and 5, and because it does not use X-windows or any type of screen management, it can be used in interactive mode from any type of terminal, or even run in batch mode if so desired. X-mail also features comprehensive on line help, complete installation instructions, and checksums of all distribution files to provide verification that your copy has not been corrupted.

Notes: Operating System VAX/VMS V4 and V5 are required. Program must be installed with privilege.

Changes and Improvements: Runs on VMS versions 4 and 5.

Media (Service Charge Code): User's Manual (EB), 600' Magnetic Tape (MA) **Format:** VMS/BACKUP

DECUS NO: V00385 **TITLE:** EDX Editor **Version:** 5.1, June 1989

Submitted by: David Deley, Santa Barbara, CA

Operating System: VAX/VMS V4.6, V4.7, V5.0, 5.1 **Source Language:** MACRO-32, VAXTPU **Keywords:** Editors, Emulators, Word Processing, TPU

Abstract: EDX supports all the major functions of the EDT editor plus many other advanced features not available in EDT or the newer EVE editor. EDT users can easily switch to EDX which is faster and more powerful without having to learn a new editor all over again. EDX uses the EDT-style keypad and also supports a WPS-style keypad.

Some of the advanced features of the EDX editor are:
. Columnar cut and paste mode.
. Wildcard search mode, case sensitive or non-case sensitive, with optional string to exclude as a match.
. Search for all lines containing a specified string and list them along with the corresponding line number.
. Search for and highlight matching parenthesis.
. Sort a buffer, range, or columnar range.
. Translate a buffer from EBCDIC to ASCII, and vice versa.
. Encrypt a buffer using the American National Data Encryption Standard algorithm X3.92-1981.
. Compare two buffers line by line.
. Lock files, preventing others from editing them while you do.
. Obtain a directory listing, with optional /SIZE and /DATE.
    - Read in a selected file from the directory listing, or delete a selected file, or lock a selected file.
. Change your default directory.
. Translate and create DCL symbols and logical names.
Note that all of the above features are performed within the editor, without spawning a subprocess.

EDX is built on the high performance, programmable VAX Text Processing Utility (VAXTPU). Users familiar with VAX-TPU can extend the editor's functions by writing new procedures in VAXTPU.

Notes: Operating System VAX/VMS V4.4 or later is required.

Changes and Improvements: Significant improvements over version 4. New features, new documentation.

Media (Service Charge Code): User's Manual (EB), 600' Magnetic Tape (MA) **Format:** VMS/BACKUP

DECUS NO: V00383 **TITLE:** Flowchart Generator **Version:** 1.1, May 1989

Submitted by: David Cohen, Security Pacific Autom. Co. Global Sys., Los Angeles, CA

Operating System: VAX/VMS V4.5 **Source Language:** DCL, VAX COBOL **Software Required:** COBOL **Keywords:** File Management, Tools –Applications Development, Utilities – VMS

Abstract: Allows you to turn your COM files into pictures. User creates a Flowchart List, containing step names, input/output names, and comments. Step names can be the names of programs that the COM file runs, DCL commands, or other nested COM files. Input/output names are file names. Comments can

be any string in parentheses. The Flowchart generator reads this list and creates a chart with boxes, arrows and text. It tracks the input and output names, so that if the output of one step is the input of a later step, it will be so labelled. There is an instruction manual included.

Changes and Improvements: Bug fix.

Media (Service Charge Code): 600' Magnetic Tape (MA) **Format:** VMS/BACKUP

DECUS NO: V00363 **TITLE:** CPUCHECK **Version:** 2.3, July 1989

Submitted by: F.A. Canali, Gould Inc., Newburyport, MA

Operating System: VAX/VMS V5.1 **Source Language:** DCL, MACRO-32, MSG, VAX BASIC **Keywords:** Security, System Management — VMS, Utilities –VMS

Abstract: CPUCHECK is a program for monitoring CPU usage and killing inactive users. It is designed to run in a memory limited system and attempts to put as little swapping load on a system as possible consistent with detecting inactive users. Sloppiness in timing inactive users is traded for lighter swapping loads on the system. Documentation is in the form of comments at the top of the source code.

Changes and Improvements: Use FORCEX: add protected images.

Media (Service Charge Code): 600' Magnetic Tape (MA)Format: VMS/BACKUP

DECUS NO: V00362 **TITLE:** XDELETE **Version:** 5.2, May 1989

Submitted by: Dr. Gerd Kobschall, Institut fuer Kernphysik, D-6500 Mainz, West Germany

Operating System: VAX/VMS V4.5, V4.7, V5.1 **Source Language:** VAX FORTRAN **Memory Required:** 220KB **Software Required:** SMG-Routines from runtime library **Hardware Required:** VT100 Terminal **Keywords:** File Management, Ulilities – VMS

Abstract: The XDELETE utility gives the user a full screen view of the files in the current directory. The user can mark files for deletion, can type the contents of files in a separate window, rename the files and he can change the current directory. All actions are done in a full screen environment.

Changes and Improvements: Better error checking, type of more possible file types, rename option, checks existence of directories.

Media (Service Charge Code): 600' Magnetic Tape (MA) **Format:** VMS/BACKUP

DECUS NO: V00342 **TITLE:** IIT **Version:** 5.0-2-A, May 1989

Submitted by: C.J. Chapman, Philips Defence Systems MEL, Crawley, Sussex, England RH10 2PZ

Operating System: MicroVMS V5.0-2, VAX/VMS V5.0-2 **Source Language:** MACRO-32 **Memory Required:** 22KB Virtual Allocation **Keywords:** System Management – VMS, Utilities – VMS

Abstract: IIT (Idle Interactive Timeout), Security timeout is a systems management tool that will terminate idle interactive processes in the event of users leaving terminals unattended. Interactive processes are considered idle if they use less than two hundred milliseconds, twenty tic's of CPU time and no buffered or direct I/O's within the default timeout period of ten minutes. These values can be adjusted to suit your site. Parent processes will not be considered idle if any subprocess is active within the chain.

Features include:
. Process notification before termination.
. Dynamic adjustment of timeout period.
. Process priority override.
. Very low CPU usage.

Release notes are distributed with each order.

Changes and Improvements: Complete revision for VMS 5.0.

Media (Service Charge Code): 600' Magnetic Tape (MA) **Format:** VMS/BACKUP, or order VL0009

DECUS NO: V00286 **TITLE:** VIEW **Version:** 5.0-2-B, June 1989

Submitted by: C.J. Chapman, Philips Defence Systems MEL, Crawley, Sussex, England, RH10 2PZ

Operating System: MicroVMS V5.0-2, VAX/VMS V5.0-2 **Source Language:** MACRO-32 **Memory Required:** 30KB Virtual Allocation **Keywords:** System Management – VMS

Abstract: The VIEW utility is a system management tool that enables the Systems Manager to display information on system processes or user processes. VIEW is very useful for taking a snapshot look at your system to establish what images are currently executing. VIEW continuously displays the following process information using manual scroll with dynamic refresh. Use any video terminal that supports the listed keypad functions:

. User Name or Process Name, Image Name, Process Id.
. Login Time, Uic, Process State/Type, CPU Min/Sec.
. Base Priority Current Priority, Working Set Size.
. Image Activation Count, Disk I/O, Buffered I/O.
. Page Faults, VMS Release, CPU's, Processes, Node.
. Idle Time and Uptime since boot time, Date Time.
. Terminal Device, Directory and Image Specification.

Idle time is computed using the arithmetic mean for VAX's using more than one Central Processor Unit.

Terminal Keypad Functions:
. Increase/Decrease Update Interval.	(Up/Down.1)
. Move Process Highlight Bar	(Up/Down.2)
. Increase/Decrease Base Priority	(Left/Right)
. Display Process Page	(Prev/Next)
. Enable/Disable Highlight Bar	(Find.1)
. Clear Alternate Process Buffer	(Find.2)
. Process User or Process Name	(Select.1)
. Alternate Process Buffer	(Select.2)
. Status Flag Display	(Insert.1)
. Move to Directory	(Insert.2)
. Delete Process	(Remove)
. Help Display	(Help)
. Clear Page	(Do)
. Exit	(Ctrl__Y,C)

Multifunction keys are identified using (.1), (.2).

Release Notes are distributed with each order.

Changes and Improvements: Internal field widths have been extended.

Media (Service Charge Code): 600' Magnetic Tape (MA) **Format:** VMS/BACKUP, or order VL0008

DECUS NO: V00279 **TITLE:** WEVE – WONDERFUL EVE EDITOR **Version:** 2.0, December 1988

Submitted by: Messrs. K. Swystun & A. Baillie, Saskatoon Cancer Clinic, Saskatoon, Saskatchewan, Canada S7N OXO

Operating System: VAX/VMS **Source Language:** VAX FORTRAN **Hardware Required:** VT100 or VT200 compatible terminals **Keywords:** Editors, EVE, TPU

Abstract: WEVE (Wonderful EVE Editor) is an editor interface that has been designed to emulate and extend the EDT editor. It is based on the EVE editor which has been enhanced with several user written VAXTPU procedures. This software is intended to give current EDT users an interface emulating EDT, but also incorporating the more powerful features of VAXTPU, such as windowing; multiple buffers intimately related to specific files; spawn; and the ability to run DCL commands from within the editor. Functions have also been written to do things such as: automatic indenting; jump to previous buffer; delete buffer; clear buffer; automatic jump to file that cursor points to; show current line number; join line; begin of line only find; alternate cursor behavior option; show all buffer names; and automatic documentation template insertion. In addition to giving the EDT user immediate added functionality, it also gives him the ability to enhance or customize the editor by writing further procedures.

Notes: Operating system VAX/VMS V.4.4 or higher is required.

Changes and Improvements: Revised to run under VMS 5.0.

Media (Service Charge Code): 600' Magnetic Tape (MA) **Format:** VMS/BACKUP, or order VL0008

DECUS NO: V00193 **TITLE:** VTEDIT: Keypad Text Editor and Corrector for VAXTPU **Version:** 5.0, July 1989

Submitted by: Dr. Gerhard Weck, Infodas GmbH, D-5000 Koeln 71, West Germany

Operating System: MicroVMS V4.7, VAX/VMS V5.1-1 **Source Language:** VAX FORTRAN, VAXTPU **Software Required:** VAXTPU V1.2 or higher (included in VMS V4.4 or higher)**Hardware Required:** VT100, VT200, VT300 series of ANSI terminals or compatible terminal **Keywords:** Editors, TECO, TPU

Abstract: The Video Terminal Editor VTEDIT is an editing interface for the VAX Text Processing Utility VAXTPU, and optionally for VAX LSE. The VTEDIT interface is an efficient, keypad driven editor allowing multi-window editing and providing semi-automatic, context dependent text formatting. VTEDIT implements, among others, the following features:
. Multi file and multi buffer editing.
. Insert and overstrike editing.
. Free and bound cursor movement.
. Recognition of all TECO match control constructs and access to VAXTPU pattern building constructs.
. Journaling the editing session.
. Access to the VAX/VMS operating system and to VAXTPU.
. Operations like:
  - Search and replace.
  - Rectangular cut, paste, and delete.
  - Pattern directed replacement operations.
  - Remember and retrieve buffer positions.
  - Insertion of date, time, file and buffer names.
  - Case and position control for searches.
  - Case conversion and capitalization of words.
  - Center line and fill paragraph.
  - Control of tabulator setting.
  - Replace Tabs with spaces and vice versa.
  - Sorting of buffers and ranges.
  - Wildcard file and buffer selection.
. Optional semi automatic, context dependent text formatting providing the following functions:
  - Case conversion/automatic case control.
  - Automatic indentation.
  - Manual correction of indentation.
  - Automatic word wrap and/or line justification.
  - Automatic insertion of closing parentheses and string delimiters.
  - highlighting of the matching opening parenthesis and string delimiter.
. Command driven line mode editing. . Menu selection of editor commands.
. Use of the mouse as positioning and command input device.
. Extensive on line help.
. Optional access to the Language-Sensitive Editor VAX LSE, providing operations to:
  - Fill and align program comments.
  - Retrieve sources from a CMS library.
  - Move to and/or delete placeholders.
  - Expand tokens, routines, placeholders, and aliases.
  - Compile sources and review errors.
  - Locate errors and retrieve the corresponding source text.
. Access the LSE command interpreter directly.
. Optional access to the Source Code Analyzer VAX SCA, providing operations to:
  - Find declarations of symbols.
  - List positions of variable declarations and/or references.
  - Retrieve corresponding sources.
  - Access the SCA command interpreter directly.

Notes: Operating System VMS V4.4 - V4.7, V5.0 and higher is required. Optional interfaces to VAX LSE (version 2.2) and VAX SCA (version 1.2).

Changes and Improvements: Compatibility with VAXTPU version 2, line mode commands, mouse support, EDITOR command files, better performance.

Media (Service Charge Code): User's Manual (ED), 600' Magnetic Tape (MA) **Format:** VMS/BACKUP

DECUS NO: RB0129 **TITLE:** KRAMDEN Utilities **Version:** July 1989

Submitted by: Bryan Higgins, Salt Creek Research

Operating System: MS-DOS **Source Language:** ASSEMBLER, C **Keywords:** Utilities – MS-DOS

Abstract: The KRAMDEN Utilities are a set of programs for the Digital Equipment Corporation Rainbow 100 running operating system MS-DOS V2.0 or higher. Some of the functions are:

- File utilities, including alternatives to COPY, RENAME, and DEL, which handle multiple files at once.
- A directory listing program (alternative to DIR) which sorts files by name or by date, can restrict matches to files newer than a certain date, shows the weekday of the file date, etc.
- A program which removes all files and directories from a floppy without reformatting.
- A file backup utility.
- A command editor which allows recall, edit and re-execution of previously typed DOS commands.
- A utility which locates files across all drives and directories.
- A utility which searches files for text strings.
- A listing paginator for printers.
- Clock programs.

Notes: Operating System MS-DOS V2.0 or greater is required. Executable and/or object code is included.

Changes and Improvements: New features.

Sources not included.

Media (Service Charge Code): One RX50 Diskette (JA) **Format:** MS-DOS

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
Amdalh CSD
2200 North Greenville Ave.
Richardson, TX 75081
(214) 699-9500 x 272

NEWSLETTER EDITOR
Curt Snyder
Allegan
2525 Dupont Drive
Irvine, CA 92715-1599
(714) 752-4760

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

Terman Frydenlund Applied Tech.
10839 Broadwater Drive
Fairfax, VA 22032

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

COMMUNICATIONS COMMITTEE
John Shephard
El. Lilly Research Labs.
Lilly Corporate Center
MC625, Bldg. 98C/2321
Indianapolis, IN 46285
(317) 276-7947

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
Marcia Roland
Marlboro, 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
Bob Sr. DOC H
Pioneer Standard Elect.
4800 East 131st St.
Data Prfocessing
Cleveland, OH 44105
(216) 587-3600-389

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
Richard Kemp
Softport
99 Madison Avenue
New York, NY 10016
(212) 889-6575

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
Paula Daley
Digital Equipment Corporation
Merrimack, NH 03054-0430

Pam Kukla
Digital Equipment Corporation
Maynard, MA 01754

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. Chris 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

John Babiarz
System Support Services
15 Aircraft Road
Southington, CT 06489
(203) 628-5674

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-1-1
Corning, New York 14831
(607) 974-4496

Herbert G. Reines
Reznick Feddder & Silverman
4520 East West Highway
Suite 300
Bethesda, MD 20814
(301) 652-9100

Richard Copeland
Corning Inc.
Mail Stop HP-CB-06
Corning, NY 14831
(607) 974-8020

J. Edward Crosson
Merrell Dow Pharmaceuticals
2110 East Galbrith Road
Cincinnati, OH 45215-6300
(513) 948-7558

Dottie Jo Elliott
NSI Technology Services Corp.
P.O. Box 12313
Research Triangle Park, NC 27709
(919) 549-0611 x213

VOLUNTEER COORDINATOR
Harry Miller
City of Ontario Police
200 N. Cherry Avenue
Ontario, CA 91754
(714) 988-06481

ASSISTANT VOLUNTEER COORD.
Judy Cutuli
Warner Lambert Company
10 Webster Road
Milford, CT 06460
(203) 783-0247

SEMINARS COMMITTEE REP.
Dana Schwartz
9325 Creekview Drive
Laurel, MD 20708
(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

DIGITAL COUNTERPARTS
John L. Henning
Digital Equipment Corporation
110 Spit Brook Road, ZK02-3/K06
Nashua, NH 03062-2698
(603) 881-2705

John F. Wood
Digital Equipment Corporation
110 Spit Brook Road, ZK02-2/Q21
Nashua, NH 03062-2698
(603) 881-0242

ARTIST & LIBRARY REP.
Bart Z. Lederman
System Resources Corporation
DOT Transportation System Center
Kendall Square DTS-66
Cambridge, MA 02142
(617) 494-2792

WORKING GROUP COORDINATOR
Larry Jasmann
U.S. Coast Guard
10067 Marshal Pond Road
Burke, VA 22015
(202) 267-2624

RALLY WORKING GROUP CHAIR
Steven G. Fredrickson
Fredrickson Consulting Service
2722 37th Avenue SW
Seattle, WA 98126
(206) 938-0482

RALLY WORKING GROUP VICE CHAIR
B. Paul Bushueff
DOT Transportation System Center
Kendall Square DTS-66
Cambridge, MA 02142
(617) 494-2090

POWERHOUSE W/G CHAIR
David Hatfield
Merrimack County Telephone Co.
P.O. Box 337
Contoocook, NH 03229
(603) 746-9911

SMARTSTAR WORKING GROUP CHAIR
Charles B. Gross
Eagle Technology
P.O. Box 1196
Dumfries, VA 22026
(703) 690-2155

ACCENT-R USER GROUP LIAISON
Winston Tellis
Fairfield University
North Benson Road
Fairfield, CT 06430
(203) 254-4000

FOCUS WORKING GROUP CHAIR
Les Hulse
The Gillette Company
Prudential Tower Bldg.
Boston, MA 02199
(617) 421-7910

ORACLE WORKING GROUP CHAIR
Jay-Michael Baslow
Chemical Bank
277 Park Avenue
New York, NY 10172
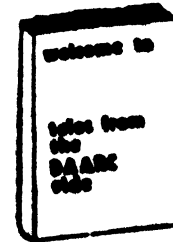(212) 310-5465

ORACLE WORKING GROUP VICE CHAIR
Shaul Ganel
Georgeson & Company Inc.
Wall Street Plaza
New York, NY 10005
(212) 440-9933

INGRES WORKING GROUP CHAIR
Larry W. Hicks
Turn Key Solutions, Inc.
1914 Fox Sterling Drive
Raleigh, NC 27606
(919) 460-9896

CORTEX WORKING GROUP CHAIR
Eric S. Dubiner
duPont IEA
Nemours Building Suite 9418
Wilmington, DE 19898
(302) 773-6780

DAARC SIG

CHAIRMAN
James Deck
Inland Steel Research Lab.
3001 East Columbus Drive
East Chicago, IL 46312
(219) 392-5613

SESSION NOTE EDITOR
Tracy K. Mixon
Science Applications Int'l. Corp.
P.O. Box 2501
800 Oak Ridge Tpke.
Oak Ridge, Tennessee 37831
H/(615) 966-3053
W/(615) 576-2262

COMMUNICATIONS REPRESENTATIVE
NEWSLETTER EDITOR
Dale Hutchison
Cummins Engine
500 Jackson Street
Columbus, Indiana 47201

SYMPOSIUM REPRESENTATIVE
Mike Gallant
Cummins Engine
4720 Baker Street Ext.
Lakewood, NY 14750

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

RS-1
George Winkler
CPC International
Argo, IL

CIM WORKING GROUP
Randall S. Gamby
McDonnell Aircraft Co.
M/S 0801480
P.O. Box 516
St. Louis, MO

DIGITAL COUNTERPART
Bill Forbes
Marlboro, MA

Drew Comstock
Maynard, MA

Laura Startzenbach
Marlboro, MA

DATA MANAGEMENT SYSTEMS SIG
CHAIRMAN
Tim Mueller
Silutions, Inc.
19m E. Central Ave., Suite 223
Paoli, PA 19301
(215) 640-4344

MEMBER AT LARGE
Past SIG Chairman
Doug Dickey
GTE Government Systems
1700 Research Blvd.
Rockville, MD 20805
(301) 294-8462

SESSION NOTES EDITOR
Alan Schultz
Southwestern Bell Yellow Pages
12800 Publications 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
Jodi Austin
Sharpe Microelectronics Tech.
312 SE Stonemill Dr.
Vancouver, WA 98684
(206) 253-3789

MEMBERSHIP COORDINATOR
MEMBER AT LARGE
Rocky Hayden
Userware Intl.
2235 Meyer Avenue
Escondido, CA
(619) 745-6006

SEMINAR REP
Steve Gomez
Signal Technology, Inc.
1750 Montgomery St.
San Francisco, CA 94111
(415) 954-8532

CAMPGROUND COORDINATOR
OLTP WORKING GROUP COORDINATOR
Rosemary O'Mahony
Arthur Anderson & Co.
33 West Monroe St.
Chicago, IL 60603
(312) 507-6510

OLTP WORKING GROUP COORD.
Larry Goodhind
Sharp Microelectronics Tech.
312 SE Stonemill Dr.
Vancouver, WA 98684

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
(203) 807-4077

STORE REPRESENTATIVE
FIMS STANDARDS REP.
Paul W. Plum, Jr
Lukens Steel Company
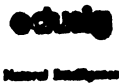Coatesville, PA 19320
(215) 383-2024

SOFTWARE AG WORKING GROUP COORDINATOR
Ron Kaminski
Inland Steel Company
3120 Watling Street, MS 5-000
East Chicago, IN 46312
(219) 853-7668

DIGITAL COUNTERPART
Dan Frantz
Charles Kelley
Nashua, NH

Reed Powell
Marlboro, MA

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
Jim Gerland
Suny at Buffalo
University Computing Services
Computer Center
Buffalo, New York 14260
(716) 636-3557

SESSION NOTES EDITOR
Claude Watson
Lansing Community College
P.O. Box 40010, Code 13
419 North Washington Sq.
Lansing, MI 48901-7210
(517) 374-1750

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
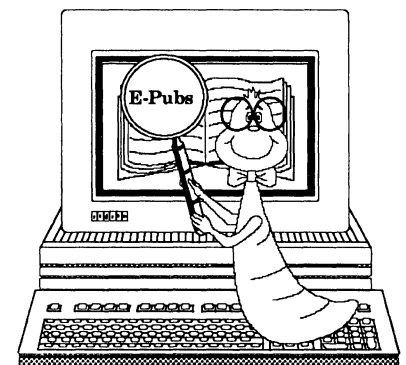
Electronic Publishing UIG

CHAIR (*)
Kevin J. Kindschuh
Northlake Software
812 SW Washington, Suite 1100
Portland, OR 97205
(503) 228-3383
(503) 228-5662 (FAX)

VICE CHAIR (*), WORKING GROUP COORDINATOR
William H. (Bill) Koppelman
Moody's Investors Service
99 Church St.
New York, NY 10007
(212) 553-0474
(212) 553-4737 (FAX)

SIG COUNCIL UIG COORDINATOR
Katherine (Kitt) Trimm
Pivotal, Inc.
6892 Dorado Ct.
Tucson, AZ 85715
(602) 886-5563

SYMPOSIA COMMITTEE REP. (*)
John L. Holland
J C Penney Co.
613 Rawhide Court
Plano, TX 75023
(214) 591-3393

ASST. SYMPOSIA REP.
Mary Margaret McCormick
McDonnell Douglas
P.O. Box 516 MS:1064664
St. Louis, MO 63166-0516
(314) 595-7070

COMM COMM REP (*), SESSION NOTES EDITOR
Kimberly White
Interleaf, Inc.
1646 N. California Blvd. #440
Walnut Creek, CA. 94596
(415) 946-9500
(415) 947-1375 (FAX)

NEWSLETTER EDITOR, UPDATE.DAILY REPORTER
Richard R Wolff
Bonneville Power Administration
Routing SWHP
PO Box 3621
Portland, OR 97208
(503) 230-5069
(503) 230-5316 (FAX)

SIR/WISHLIST COORDINATOR
Patty English
Signal Technology
5951 Encina Road
Goleta, CA 93117
(805) 683-3771
(805) 967-0871 (FAX)

TAPE LIBRARIAN
David Warren
Boeing Aerospace
PO Box 3999
MS: 6J-39
Seattle, WA 98119
(206) 237-8515

VOLUNTEER COORDINATOR
Ann P Jackson
Interleaf, Inc.
10 Canal Park
Cambridge, MA 02141
(617) 577-9813 x 4486

SEMINARS COORDINATOR
(Open)

STORE REP
(Open)

INTERLEAF WG CO-CHAIRS
Janet E Bressan
Boeing Computer Services
PO Box 3999 6J-39
Seattle, WA 98124
(206) 237-8418

Lisa M. Pratt
Boeing Aerospace
PO Box 3999,MS 6J-39
Seattle, WA 98124-2499
(206) 237-8603

TEX/LaTEX/WEB WG CHAIR
Donald E. Ambyh
Delco Electronics Corp.
PO Box 471, MS 1A21
Milwaukie, WI 53201
(414) 768-2682

DECwrite WG CHAIR
(Open)

MEMBER-AT-LARGE
John W. Shepherd
Eli Lilly Research Labs
Lilly Corporate Center
MC625 Bldg. 98C/2321
Indianapolis, IN 46285
(317) 276-7947

DEC COUNTERPARTS (*)
Cathy St Martin
Digital Equipment Corporation
10 Tara Blvd
Nashua, NH 03052

Don Hedman
Digital Equipment Corporation
110 Spit Brook Road ZKO3-4/T61
Nashua, NH 03063

DEC CONTACTS
Marian Weisenfeld
Digital Equipment Corporation
110 Spit Brook Road ZKO3-4/T61
Nashua, NH 03063

Rick Landau
Digital Equipment Corporation
4 Technology Park Drive DSG1-2/C7
Westford, MA 01886

The Graph Paper

GRAPHICS APPLICATIONS SIG

CHAIRPERSON
Bijoy Misra
Harvard Smithsonian Center for Astrophysics
60 Garden Street, MS39
Cambridge, MA 02138
(617) 495-7393
bijoy@cfa3.bitnet

HARDCOPY WORKING GROUP CHAIR
Henry Schneiker
P.O. Box 42767
Tucson, Arizona 85733
(602) 325-5884

SYMPOSIUM COORD.
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

STORE REPRESENTATIVE
CAMPGROUND COORD.
Stephen Schultz
Rochester Institute of Technology
Center for Imaging Science
One Lomb Memorial Drive
Rochester, NY 14623
(716) 475-6907
BITNET: SLS4255@RITVAX
USENET: harvard!rochester!ritcv!ulta!sls4255

SESSION NOTES EDITOR
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
Henry Tenarvitz
Olmsted Engineering
2320 Areo Park Ct.
Traverse City, MI 49684
(616) 946-3174

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 WORKSTATIONS W/G
SEMINARS REP
Daniel Land
John Fluke Mfg. Co., Inc.
Mail Stop 221B
P.O. Box C9090
Everett, WA 98206
(206) 356-5257

VAX SIG LIASON
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

UIS WORKING GROUP
Jeff Johnson
Ford Aerospace
Forbes Blvd.
Seabrook, MD 20706
(301) 464-6800

COMPUTER ANIMATION WORKING GROUP
Steve Hankin
NOAA/PMEL/Bldg. No. 3
7600 Sand Point Way NE
Seattle, WA 98115
(206) 526-6080

GKS/PHIGS WORKING GROUP
Warren S. Yogi
NOAA NOS Ocean Applications Group
NPS FNOC Bldg. 4
Airport Road
Monterry, CA 93943
(408) 646-1649

PUBLIC DOMAIN SOFTWARE WORKING GROUP
Bill Varady
Exxon Research & Engg.
Route 22 East
Annandale, NJ 08801
(201) 730-2793

MEMBERS-AT-LARGE
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

DIGITAL COUNTERPARTS
Jim Flatten
Spit Brook, NH

Rick Landau
Marlboro, MA

Irene McCartney
Maynard, 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
Neil Krandall
University of Cincinnati
Pharmacology & Cell Biophysics
Cincinnati, OH

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

DIGITAL COUNTERPARTS TERMINALS
Gail Jamison-Barnes
William Andrus
Marilyn Fedel
Frank Orlando
Maynard, MA

Art Bigler
Marlboro, 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

LANGUAGES AND TOOLS SIG

CHAIR, Tex/LaTeX/WEB W/G
FOLDER EDITOR
Donald E. Ambyh
Delco Electronics Corp.
P.O. Box 471, MS1A21
Milwaukee, WI 53201
(414)768-2682

VOLUNTEERS COORDINATOR
Shirley Bockstahler-Brandt
Applied Physics Laboratory
Johns Hopkins Road
Laurel, MD 20707
(301)953-6585

SEMINAR COMMITTEE REP.
Barry C. Breen
Sundstrand Data Control, Inc.
15001 N.E. 36th Street
P.O. Box 97001
Redmond, Washington 98073-9701
(206)885-8436

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

VICE CHAIR, UNITS
SYMPOSIUM COORDINATOR
Earl Cory
Contel
31717 La Tienda Drive
Westlake Village, CA 91359
(818) 706-5385

MEMBER ANSI PL/I X3J1 STDS. COMM.
Arthur Coston
Applied Information Systems, Inc.
500 Eastowne Dr.
Chapel Hill, NC 27514
(800)334-5510

DIBOL WORKING GROUP
Mark Derrick
WAAY Television
P.O. Box 2555
Huntsville, AL 35804
(205)535-3131

NEWSLETTER EDITOR
ALT. COMMUNICATIONS COMM. REP.
Alan Folsom, Jr.
Fischer & Porter Co.
E. County Line Rd.
Warminster, PA 18974
(215)674-7154

MEMBER ANSI COBOL X3J4 STDS. COMM.
Bruce Gaarder
Donahue Enterprises, Inc.
2441 26th Avenue. S.
Minneapolis, MN 55406
(612)721-2418

INTERSIG COORDINATOR
Dorothy Geiger
Wollongong Logistics Group
49 Showers Drive #451
Mountain View, CA 94040
(415)962-7160
(415)948-1003

EUROPEAN METHODS, L&T INTERFACE
Bernd Gliss
Max-Planck-Institute
Heisenbergstrasse 1
7000 Stuttgart 80, W. Germany
(711)686-0251

CHAIR, SECURITY WORKING GROUP
Rich Harris
General Research Corp.
5383 Hollister Avenue
P.O. Box 6770
Santa Barbara, CA 93160-6770
(805)964-7724

OBJECT-ORIENTED LANGUAGES WORKING GROUP
Robert Harwood
The Torrington Company
59 Field Street
Torrington, CT 06790
(203)482-9511 x2406

STORE REPRESENTATIVE
CHAIR, TECH. PROD. OF DOC. W/G
Howard Holcombe
RCA
Front & Cooper St.
Camden, NJ 08055
(609)338-4946

PAST SIG CHAIR
PRODUCTIVITY TOOLS COORDINATOR
Kathy Hornbach
Digital Equipment Corporation
ZK03-3/Y25
110 Spit Brook Road
Nashua, NH 03062

CHAIR, TECO WORKING GROUP
Mark J. Hyde
Advanced Computing Services
209 Ardsley Drive
DeWitt, NY 13214
(315)446-7223

CASE & TOOLS INTEGRATION W/G
John Ivler
System Development Engineering
De La Rue Printrak Inc.
1250 North Tustin Avenue
Anaheim, CA 92807
(714)666-2700

ACTING SYMPOSIUM COORDINATOR
MEMBER, ANSI BASIC X2J2 STDS. COMM.
PDP-11 REPRESENTATIVE
CHAIR, PDP-11 LAYERED PRODUCTS W/G
Stephen C. Jackson
SCJ Consulting, Inc.
Suite 105
7260 University Avenue N.E.
Minneapolis, MN 55432
(612)571-8430

SESSION NOTES EDITOR
Mark Katz
GTE Gov't Systems
100 First Avenue
Waltham, MA 02154
(617)466-3437

CHAIR, ADA WORKING GROUP
Lisa Kerby-Rodgers
ESL
495 Java Drive
Sunnyvale, CA 94088
Mailstop: 505
(408)738-2888

SIG SECRETARY
CHAIR, CONFIG. MGMT. WORKING GROUP
Mark Alan Kidwell
Texas Instruments Incorporated
P.O. Box 801 M/S 8006
McKinney, TX 75069
(214)952-2058

MEMB. ANSI FORTRAN X3J8 STDS. COMM.
Dr. Joseph King
Biotechnology Center
University of Wisconsin
1710 University Avenue
Madison, WI 53705
(608)263-8970

DEVEL. COUNTERPART, TECH. LANG.
Leslie J. Klein
Digital Equipment Corporation
ZK02-3/N30
110 Spit Brook Road
Nashua, NH 03062

CHAIR, FORTRAN WORKING GROUP
Scott Krusemark
Systemation, Inc.
8473 Daisywood Ave. NW
North Canton, OH 44720
(216)499-6251

DIGITAL COUNTERPART
Joe Mulvey
Linda Ziman
Nashua, NH 03062

MEMB. ANSI FORTRAN X3J8 STDS. COMM.
Rochelle Lauer
Physics Department
Yale University
P.O. Box 6666
New Haven, CT 06511-8167
(203)432-3366

SESSION QUALITY COORDINATOR
Gary C. Lelvis
IMSL
2500 Park West Tower One
2500 City West Boulevard
Houston, TX 77042-3020
(713)782-6060

CHAIR, LOW LEVEL LANGUAGES W/G
Gerald Lester
Computerized Processes Unlimited
4200 South I-10 Service Rd., Suite #205
Metairie, LA 70001
(504)889-2784

CHAIR, PROJECT MANAGEMENT W/G
Lynn C. Lewis
Lawrence Livermore National Laboratory
University of California
P.O. Box 808
Livermore, CA 94550
(415)422-8949

ASST. CAMPGROUND COORD.
CROSS DEV. & IMBEDDED SYS. W/G
Theresa (Teri) J. McNamara
Data Card Corp.
11111 Bren Road West
Minneapolis, MN 55343
(612)931-1792

ALT. ANSI X3J4 COBOL STDS. COMM.
Dale Marriott
El Paso County Office Bldg.
27 E. Vermijo Street
Colorado Springs, CO 80903
(719)520-6435

ASST. SESSIONS QUALITY COORD.
Raymond E. Marshall
Northern Telecom, Inc.
Network Support Systems Div.
54 Regional Drive
P.O. Box 649
Concord, NH 03301-0649
(603)224-6511

CHAIR, C WORKING GROUP
James Maves
Contel
Box 5009
31717 La Tienda Dr.
Westlake Village, CA 91359
(818)706-5395

ASSOC. W/G COORD. UNSCHEDULED TOPICS
CHAIR, COBOL WORKING GROUP
ALT. SEMINAR COMM. REP.
Bruce Mebust
Burlington Northern Railroad
176 East Fifth Street
P.O. Box 64962
St. Paul, MN 55164
(612)298-2382

STEERING COMM. MEMBER-AT-LARGE
Terry Medlin
Survey Sampling, Inc.
1 Post Road
Fairfield, CT 06432
(203)255-4200

BOF CHAIRS COORDINATOR
SESSION CHAIRS COORDINATOR
Antonino N. Mione
Rutgers University
Center for Computer & Information Services
Hill Center
P.O. Box 879
Piscataway, NJ 08855-0879
(201)932-4784

DEVEL. COUNTERPART, PDP-11 SOFTWARE
Joe Mulvey
Digital Equipment Corporation
ZK01-3/J10
110 Spit Brook Road
Nashua, NH 03062-2642
(603)881-1218

CHAIR, PUBLIC DOMAIN SFTWR. W/G
Edward (Ted) Nieland
System Research Laboratories, Inc.
2800 Indian Ripple Road
Dayton, OH 45440-3696
(513)255-5156
tnieland@wbafb-aamrl.arpa

SIG CHAIR
Joseph Pollizzi, III
Space Telescope Science Institute
3700 San Martin Drive
Homewood Campus
Baltimore, MD 21218
(301)338-4901
pollizzi@stsci.edu

CHAIR, VAXset W/G
ASST. WORKING GROUP COORD.
David J. Powell
The Upjohn Company
7294-25-7
301 Henrietta Street
Kalamazoo, MI 49007
(616)385-7214

VICE CHAIR, TECHNICAL
WORKING GROUPS COORD.
CHAIR, SCAN WORKING GROUP
David K. Ream
Lexi-Comp., Inc.
26173 Tallwood Dr.
N. Olmsted, OH 44070
(216)777-0095
(216)468-0744

LISP/AI COORDINATOR
Don Rosenthal
Space Telescope Science Institute
Homewood Campus
Baltimore, MD 21218
(301)338-4844

SIG TAPE LIBRARIAN
LIBRARY COMMITTEE REP.
Tony Scandora
Argonne National Laboratory
CMT 205
Argonne, IL 60439
(312)972-7541

MEMB. ANSI DIBOL X3J12 STDS. COMM.
Kenneth Schilling
2314 Mira Vista Avenue
Montrose, CA 91020
(818)249-0795

CLINIC DIRECTOR
MASTERS COORDINATOR
SESSION EVALUATION CARDS TABULATOR
George Scott
Computer Sciences Corporation
304 West Route #38, P.O. Box N
Moorestown, NJ 08057
(609)234-1100

STANDARDS COORDINATOR
CHAIR, PASCAL WORKING GROUP
CHAIR, MODULA-2 WORKING GROUP
MEMB. ANSI PASCAL X3J9 STDSD COMM.
E. Wayne Sewell
E-Systems, Garland Division
Box 660023, MS 53700
Dallas, TX 75266-0023
(214)272-0515 x3553

UPDATE.DAILY REPORTER
PUBLIC RELATIONS COORD.
Terry Shannon
Computer Information Systems
165 Bay State Drive
Braintree, MA 02184
(617)848-7515

CHAIR, APL WORKING GROUP
Chet Small
MIT Lincoln Laboratory
244 Wood Street
Lexington, MA 02173
(617)981-4172
(617)863-5500 x4172

CHAIR, PL/I WORKING GROUP
Jack Straub
13102 Borgman
Huntington Woods, MI 48070
(313)358-6338
(313)541-1941

VICE CHAIR, LOGISTICS
CAMPGROUND COORDINATOR
MEMB. ANSI C X3J11 STDS. COMM.
Michael S. Terrazas
LDS Church
50 E. North Temple, 27th Floor
Salt Lake City, UT 84150
(801)240-3246

DEVEL. COUNTERPART, COMMERC. LANG.
Jim Totton
Digital Equipment Corporation
ZK02-3/K06
110 Spit Brook Road
Nashua, NH 03062

CHAIR, BASIC WORKING GROUP
WISHLIST COORDINATOR
Bob Van Keuren
4087 Chamoune Avenue
San Diego, CA 92105
(619)283-5285

SUITE & RECEPTION COORD.
Matt Variot
Contel
Box 5009
31717 La Tienda Drive
Westlake Village, CA 91359
(818)706-5388

PAST SIG CHAIR
Sam Whidden
American Mathematical Society
201 Charles St.
P.O. Box 6248
Providence, RI 02940
(401)272-9500

STEERING COMM. MEMBER-AT-LARGE
Jay Wiley
Bechtel Power Corp.
12400 East Imperial Highway
Norwalk, CA 90650
(213)807-4016

ASST. NEWSLETTER EDITOR
Jim Wilson
Pfizer Inc.
QC Division
P.O. Box 88
Terre Haute, IN 47808
(812)299-2121 x271

CHAIR, TPU/EVE/LSE W/G
John Wilson
Knight Programming Support
724 Oak Brook Blvd.
Battle Creek, MI 49015
(616)961-3515

ALT. ANSI X3J9 PASCAL STDS. COMM
Phil Wirth
E-Systems, Garland Division
Box 660023, MS 53730
Dallas, TX 75266-0023
(214)272-0515 x4319

SOFTWARE METRICS WORKING GROUP
Allan F. Witt
Monsanto Company
Mail Zone O2J
800 N. Lindbergh Blvd.
St. Louis, MO 63167
(314)694-3997

COMMUNICATIONS COMMITTEE REP.
Kerry Wyckoff
1117 E. 1000 Street
Spanish Fork, UT 84660
(801)240-5948

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

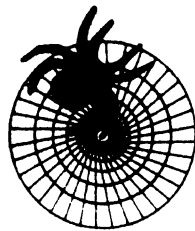**PAST CHAIR**
MUMPS DEV. COMMITTEE REP.
Mark Berryman
Science Applications Int'l. Corp.
10260 Campus Point Drive MS 45
San Diego, CA 92121
(619) 535-7603
Internet: BERRYMAN@FWVC.SAIC.COM

DIGITAL COUNTERPART
Russ White
Marlboro, 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

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

SYMPOSIA REP
John Ceisel
100 S. Wacker #634
Chicago, IL 60606
(312) 507-6410

LIBRARY COMMITTEE REP
Mike West
Network Manager
USAF Avionics Laboratory
WRDC/AATC
WPAFB, OH 45433-6543
(513) 255-1953

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
Steelcase Inc.
(616) 247-2152

NEWSLETTER EDITOR
Judi Mandl
University of Conn. Health Center
263 Farmington Ave.
Farmington, CT 06032
(203) 679-3912

ASSISTANT NEWSLETTER EDITOR
Rick Carter
Systems Programmer/Analyst
Milcare
8500 Byron Road, Loc. 0320
Zeeland, MI 49464
(616) 772-8350

SESSION NOTES EDITOR
Mary Marvel-Nelson
General Motors Research Lab.
Warren, MI 48090
(313) 986-1382

WISH LIST COORDINATOR
Lt.Stuart L. Labovitz
USAF WDRC/ELMT
Bldg 620 Area B
WPAAFB, OH 45433-6523
(513) 255-7680
(513) 255-2062 Sec.
DCS: LABOVITZ
INTERNET: LABOVITZ%ETD1.DNET@WPAFB-ABLAB.ARPA

PAST CHAIRMAN
Bill Brindley
HDQ Naval Security Group Cmd.
(202) 282-0527

TECHNOLOGY COORDINATOR
Bill Hancock
ERI Training
P.O. Box 13557
Arlington, TX 76017
(817) 467-7031
(212) 334-1240
DCS: TOPAZ::HANCOCK
DECUServe: EISNER::HANCOCK
CompuServe: 76324,1303
Internet: HANCOCK@AMB2.LARC.NASA.GOV

MEMBER-AT-LARGE
Sandy Traylor
Target Systems
21063 Carlos Rd.
Yorba Linda, CA 92686

DIGITAL COUNTERPART
Monica Bradlee
Digital Equipment Corporation
550 King St. LKG2-1/U2
Littleton, MA 01460-1289
(508) 486-7341

STORE REPRESENTATIVE
Lesley C. Gray
United Airlines Flight Center
Stapleton International Airport
Denver, CO 80207
(303) 398-4035

NETWORKS SEMINARS REPRESENTATIVE
Michael C. Hutton
Eastman Kodak
901 Elmgrove Road
D-645 2-9A-EP
Rochester, NY 14653-5819
(716) 726-1941

Office Automation

OFFICE AUTOMATION SIG

(*) CHAIR
Joseph W. Whatley
Information Service & Technology
Nielsen Plaza
Northbrook, IL 60062-6288
(312) 480-6104

(*) VICE CHAIR
Ralph Bradshaw
Johnson and Johnson
Route 202
Raritan, NJ 08869-1489
(201) 685-3434

SIR PROGRAM ADMINISTRATOR
Edward L. Bowen
Bell South Services
1876 Data Drive, Room B204
Birmingham, AL 35244
(205) 998-6800

LIBRARY REPRESENTATIVE
Bob Hassinger
Liberty Mutual Research Ctr.
71 Frankland Road
Hopkinton, MA 01748
(508) 435-9061

NEWSLETTER EDITOR
COMMUNICATIONS COMMITTEE REP, LTD.
Therese LeBlanc
LeBlanc & Assoc.
275 London Place
Wheeling, IL 60090
(312) 459-1784

ASSISTANT NEWSLETTER EDITOR
Roger Bruner
Foreign Mission Board
3806 Monument Avenue
Richmond, VA 23230
(804) 353-0151

SUITE COORDINATOR
Open

SYMPOSIUM COORDINATOR
Lynda L.Peach
Mustang Fuel Corp.
2000 Classen Center, 800 East
Oklahoma City, OK 73106
(405) 557-9513

ROADMAP/PUBLICATIONS COORDINATOR
Scott McClure
3M/Industrial Tape Division
220-8E-01 3M Center
St. Paul, MN 55144-1000
(612) 736-4297

SESSION CHAIR COORDINATOR (East Coast)
Kae Sobczyk
Cooper Tire & Rubber Co.
P.O. Box 550
Findlay, OH 458040
(419) 424-4283

SESSION NOTE CHAIR (West Coast)
Open

CAMPGROUND COORDINATOR
Tony Iole
OAS
661 W. Germantown Pike
Plymouth Meeting, PA 19462
(215) 834-1010

TAPE COORDINATOR
Ray Kaplan
P.O. Box 32647
Tucson, AZ 85751
(602) 323-4606

STORE COORDINATOR
Scott McClure
3M/Industrial Tape Div.
220-8E-01 3M Center
St. Paul, MN 55144-1000
(612) 736-4297

SESSION NOTES EDITOR
George Bone
Mare Island Naval Shipyard
Vallejo, CA 94590
(707) 646-2531 (Work)

(*) SPECIAL PROJECTS COORDINATOR
Katherine Trimm
PIVOTAL, Inc.
6892 Dorado Ct.
Tucson, AZ 85715
(602) 886-5563

SECURITY WORKING GROUP CHAIR
Ray Kaplan
P.O. Box 32647
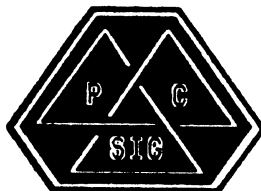Tucson, AZ 85751
(602) 323-4606

DIGITAL COUNTERPARTS
Judy Jurgens
Bob Malay
Digital Equipment Corporation
Nashua, NH

(*) 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
Adolf Coors Co.
Mail Stop BC380
Golden, CO 80401-1295
(303) 277-2855

WORK SYSTEMS WORKING GROUP CHAIR
Thomas R. Hintz
University of Florida
IFAS Computer Network
Bldg. 120
Gainesville, FL 32611
(904) 392-5180

WORKING GROUP CHAIR
Fran Garrett
San Diego Union-Tribune Pub. Co.
350 Camino de la Reina
San Diego, CA 92108
(619) 293-1676

MACINTOSH WORKING GROUP CHAIR
Kent A. Behrends
Tracor Flight Systems, Inc.
1241 East Dyer Road
Santa Ana, CA 92705
(714) 662-0333

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

ART COORDINATOR
Ken Stricker
Martin Marietta Aerospace
P.O. Box 5837 MP 1270
Orlando, FL 32855
(407) 356-1794/7725

STORE REP
George Dover
Tektronix Inc.
MS 73-646
P.O. Box 500
Beaverton, OR 97077
(503) 627-4592

VOLUNTEER COORDINATOR
Scott Warren
920 W. Cantwell
Stillwater, OK 74075
(405) 624-0070

MEMBERS-AT-LARGE
Mark Sebern
Sebern Engineering Inc.
P.O. Box 268
Cedarburg, WI 53012
(414) 375-2200

DEC COUNTERPARTS
Anita Uhler
Digital Equipment Corporation
30 Porter Road LJ02/D4
Littleton, MA 01460
(508) 486-2397

Steve Stebulis
Digital Equipment Corporation
146 Main Street
ML05-2/U13
Maynard, MA 01754
(508) 493-4491

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
Jodi Austin
Sharp Microelectronics Tech. Inc.
312 S.E. Stonemill Drive
Vancouver, WA 98684
(206) 253-3789

LIBRARY REPRESENTATIVE
R.R. Patel (PAT)
Medstone Int'l Inc.
(714) 646-8211

SEMINAR UNIT REP.
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

RSTS PRODUCT PLANNING COORDINATOR
Errol E. Ethier
Information Design and Management, Inc.
23 Hunting Avenue
Shrewsbury, MA 01545
(508) 842-4220

DIGITAL COUNTERPART
Jeanne Davis
Digital Equipment Corporation
Merrimack, NH 03054

MEMBERS-AT-LARGE
Mark Hartman
2404 E. Nutwood E23
Fullerton, CA 92631
(714) 738-8300

Jeff J. Killeen
Information Design & Management, Inc.
31 Hopedale Street
Hopedale, MA 01747

Bruce L. Gaarder
Donahue Enterprises, Inc.
2441 26th Avenue South
Minneapolis, MN 55406
(612) 721-2418 x105

TAPE COPY COORDINATOR
W. Franklin Mitchell, Jr.
Erskine College
1 Washington Street
Due West, SC 29639
(803) 379-2131

RSX/IAS SIG

CHAIRMAN
Jim Bostwick
Cargill Research
Minneapolis, MN

CHAIRMAN EMERITUS
Dan Eisner

SYMPOSIA COORDINATOR
Rick Sharpe
Toledo Edison
Toledo OH

PRE-SYMPOSIUM SEMINAR COORDINATOR
Jim McGlinchey
Warrenton, PA

MULTI-TASKER EDITOR
James McGlinchey
Software Engineering Consultant
427-3 Amherst St. Suite 303
Nashua, NH 03063
(603) 884-7378

COMMCOMM REPRESENTATIVE
DeVIAS LETTER EDITOR
Frank R. Borger
Michael Reese Medical Center
Chicago, IL

STORE COORDINATOR
Steve L. Coffman
R. R. Donnelley & Sons
Lisle, IL

SESSION NOTE EDITOR
Burt Janz
BHJ Associates
Nashua, NH

TAPE COPY COORDINATOR
Glen Everhart
GE
Glen Mills, PA

LIBRARY REPRESENTATIVE
Ted Smith
The University of PA
Philadelphia, PA

RSX/IAS HISTORIAN
IAS WORKING GROUP CHAIR
Alan Frisbie
Flying Disk Systems
Los Angeles, California

CAMPGROUND COORDINATOR
James E. Berg
Department of Defense
Ft. Meade, MD

DIGITAL COUNTERPART
Pat Cherny
Nashua, NH

WORKING GROUP COORDINATOR
Charlotte Allen
Electronic Data Systems Corp.
Detroit, MI

VICE CHAIRMAN
BUDGET & FINANCE COORDINATOR
Gary Maxwell
U.S. Geological Survey
Menlo Park, CA

SRD WORKING GROUP COORDINATOR
Bob Turkelson
Goddar Space Flight Center
Greenbelt, MD

MENU COORDINATOR
Jerry Ethington
Prolifix Inc.
Frankfort, KY

MEMBERT-AT-LARGE
Bob Curley
The University of PA
Philadelphia, PA

Arnold De Larisch
Florida Atlantic University
Boca Raton, FL

Jim Neeland
Hughes Research Labs.
Malibu, CA

Anthony Scandora Jr.
Argonne National Laboratory
Argonne, IL

Ralph Stamerjohn
Creve Coeur, MO

RT-11 SIG

CHAIRMAN(*)
Milton Campbell
Talisman Systems
1142 Manhattan Avenue #255
Manhattan Beach, CA 90266
(213) 318-2206

SYMPOSIA COORDINATOR (*)
David P. Evans
Division 1152
Sandia National Labs
Albuquerque, NM 87185
(916) 756-3291

COMMCOMM REP (acting)(*)
DECUS STORE REP
Laura DeChellis
MDB Systems Inc.
1110 W. Taft Avenue
Orange, CA 92613
(714) 998-6900

NEWSLETTER EDITOR(*)
PRODUCT PLANNING CONTACT(*)
TECO CONTACT
John M. Crowell
Multiware, Inc.
2121-B Second St. Suite 107
Davis, CA 95616
(916) 756-3291

SEMINARS (*)
STANDARDS COORDINATOR
Robert Roddy
David Taylor Research Center
Code 1564E
Bethesda, MD 20084-5000
(301) 227-1724

TAPE COPY GENERATION
John Bedel
David Taylor Research Center
Code 1564E
Bethesda, MD 20084-5000
(301) 227-1724

LUG CONTACT
Ned Rhodes
Software Systems Group
2001 North Kenilworth St.
Arlington, VA 22205
(703) 534-2297

FORTRAN CONTACT
Robert Walraven
Multiware, Inc.
2121-B 2nd St. Suite 107
Davis, CA 95616

MACRO CONTACT
> Nick Bourgeois
> NAB Software Services, Inc.
> P.O. Box 20009
> Albuquerque, NM 87154
> (505) 821-1453

DECUS LIBRARY CONTACT (*)
NETWORKING CONTACT
> Jim Crapuchettes
> Omnex Corporation
> 2483 Old Middlefield Way
> Mountain View, CA 94043
> (415) 966-8400

RUNOFF CONTACT
> John Davies, III
> David Taylor Research Center
> Code 1450
> Bethesda, MD 20084-5000
> (301) 227-1592

PERSONAL COMPUTERS
> Dennis V. Jensen
> AMES Laboratory ISU/USDO
> 310 Metallurgy
> Ames, Iowa 50011
> (515) 294-4823

NETWORKING CONTACT
> Jim Crapuchettes
> Omnex Corporation
> 2483 Old Middlefield Way
> Mountain View, CA 94043
> (415) 966-8400

OTHER LANGUAGES
> Gary Sallee
> 19912 Fernglen Drive
> Yorba Linda, CA 92686
> (714) 970-2864

TSX CONTACT
C CONTACT
> Jack Peterson
> Horizon Data Systems
> P.O. Box 29028
> Richmond, VA 23229
> (804) 740-9244

WISH LIST CONTACT
UNIX/ULTRIX CONTACT
> Bradford Lubell
> L.A. Heart Lab, UCLA
> 10833 Le Conte Avenue
> Los Angeles, CA 90024
> (213) 206-6119

PRO RT-11 & HARDWARE
> William Walker
> EG&G Mound Applied Tech.
> P.O. Box 3000, A-152
> Miamisburg, OH 45343
> (513) 865-3557

RT-11 SUITE MANAGER
> David R. Billing
> EG&G Mound Applied Tech.
> P.O. Box 3000
> Miamisburg, OH 45343
> (513) 865-3086

DIGITAL COUNTERPART
> Connie Pawelczak
> Maynard, MA

SITE SIG

NEWSLETTER EDITOR
> Gregory N. Brooks
> Washington University
> School of Medicine
> Department of Pediatrics
> 400 South Kingshighway
> St. Louis, MO 63110
> (314) 454-2237

SITE COMMUNICATIONS COMMITTEE REP.
> Terry C. Shannon
> Int'l Data Corporation
> Five Speen Street
> Farmingham, MA
> (508) 872-8200

LIBRARY COORDINATOR
> Larry W. Hicks
> 1914 Fox Sterling Drive
> Raleigh, NC 27606
> (919) 859-2599

SITE SIG CHAIR
> Timothy S. Frazer
> 1247 Woodridge Avenue
> Naples, FL 33940
> (813) 263-1669

SITE SIG SYMPOSIA COORDINATOR
> Susan M. Abercrombie
> Fitch Enterprises
> 48 Malilly Road
> Portland, ME 04103

SITE SIG VICE CHAIR
> Adam Zavitski
> 1001 Harvest Mill Court
> Raleigh, NC 27610
> (919) 266-5086

ASSISTANT SITE SYMPOSIA COORDINATOR
> Marc Lippmann
> Jamesbury Corporation
> P.O. Box 15004
> 640 Lincoln Street
> Worcester, MA 01615
> (617) 852-0200 x2804

WORKING GROUPS COORD
VAX/VMS LIASON
> Peter E. Cregger
> SAS Institute Inc.
> Box 8000 SAS Circle
> Cary, NC 27512-8000
> (919) 467-8000

MEMBER-AT-LARGE
> Gary Siftar
> 9006 So. 199th E. Avenue
> Broken Arrow, OK 74014
> (918) 491-3178

> Dave Hunt
> Lawrence Livermore Nat'l Lab
> MS L-54 P.O. Box 808
> Livermore, CA 94550
> (415) 422-0434

> Emily Kitchen
> A.H. Robins Co.
> 1211 Sherwood Avenue R2SY
> Richmond, VA 23220
> (804) 257-2925

DIGITAL COUNTERPART
> Joe Allan
> Digital Equipment Corporation
> 150 Flanders Road
> WFR1-2/G10
> Westboro, MA 01581
> (617) 870-3284

> Rosemary Good
> Digital Equipment Corporation
> BUO/E55
> Bedford, MA 01730
> (617) 249-4877

SITE SEMINARS COORDINATOR
> Philip D. Ventura
> Hughes Aircraft Co.
> Space & Comm Group
> 16800 E. Centre Tech.
> Aurora, CO 80011
> (303) 341-3394

SESSION NOTE EDITOR
> Gary Bremer
> Emerson Electric Co.
> 8100 W. Florissant Avenue
> St. Louis, MO 63136
> Attn: Gary Bremer/MS 4448
> (314) 553-4448

SECURITY WORKING GROUP
> Stephen Tihor
> 251 Mercer Street
> New York, NY 10012
> tihor@acfcluster.nyu.edu
> tihor@nyuacf.bitnet
> (212) 998-3052/228-1321

SITE/SYS. MGRS. HANDBOOK W/G
> Michael Solms
> Bancohio National Bank
> 770 W. Broad Street
> MS: 0330
> Columbus, OH 43251
> (614) 463-8722

BUSINESS PRACTICES W/G
> Dominick G. Darkangelo
> General Electric Co.
> Bldg. KW RM D160
> P.O. Box 8
> Schenectady, NY 12301
> (518) 387-5478

UNISIG

CHAIRMAN
> Kurt L. Reisler
> Hadron Incorporated
> 9990 Lee Highway
> Fairfax, VA 22030
> (703) 359-6100
> ...uunet!hadron!klr

SYMPOSIA COORDINATOR
> Michael Angelo
> Compac Computer Corporation
> 20555 SH 249
> Houston, TX 77070
> 713/374-8141
> fax: 713/374-7305
> ...cpqhua!michaela@uunet.uu.net

SESSION NOTES EDITOR
> William Cheswick
> AT&T Bell Labs
> 600 Mountain Avenue
> Murray Hill, NJ 07974
> ...research!ches

NEWSLETTER EDITOR
> Sharon Gates-Fishman
> NDC Corporation
> 730 E Cypress Avenue
> Monrovia, CA 91016
> (818) 358-1871
> !amdahl!cit-vax!ndc!sgf

VICE-CHAIR
Dorothy A. Geiger
225 St. Pauls Avenue
#12-T
Jersey City, NJ 07306
201/792-0263
...decwrl!dgeiger

TAPE LIBRARIAN
Carl Lowenstein
Marine Physical Laboratory
Scripps Institute of Oc'graphy, P-004
La Jolla, CA 92093
619/534-1805
...{decvax,ucbvax}!ucsd!mplvax!cdl

STANDARDS COORDINATOR & USENET LIAISON
Ed Gould
Mt. Xinu
2910 7th St.
Suite 120
Berkley, CA 94710
(415) 644-0146
ucbvax!mtxinu!ed

GHOST IN THE SIG
Norman Wilson
Bell Laboratories, 2C514
600 Mountain Avenue
Murray Hill, NJ 07974
(201) 582-2842
{decvax,ihnp4}!research!norman

COMMUNICATIONS COMMITTEE REP
Ron Jarrell
Computing Center, Virginia Tech
1700 Pratt Drive
Blacksburg, VA 24061-0214
(703) 231-9513
...JarrellRA@vtcc1,cc.vt.edu

SEMINARS COORDINATOR
Steven Stepanek
Computer Science Dept.
School of Eng. & Computer Science
California State University at Northridge
18111 Nordhoff St.
Northridge, CA 91330
(818) 885-2799 or 3398
...sgs@mx.csun.edu

OSF DELEGATE
Stephen M. Lazarus
Ford Aerospace
MS X-20
P.O. Box 49041
San Jose, CA 95161
408/473-4203
...sgi!sun!sd1!sml

CAMPGROUND COORDINATOR
Sophie Strauss-Duckett
NASA-Ames
MS 233-7
Moffett Field, CA 94035
415/654-4787

DIGITAL COUNTERPART
Sharon MacDonald
Ted Prindle
Digital Equipment Corporation
110 Spit Brook Road
Nahsua, NH 03062-2698

VAX SYSTEMS SIG
CHAIRMAN
Susan T. Rehse
Lockheed Missiles & Space Co.
0/19-50, B/531,
P.O. Box 3504
Sunnyvale, CA 94088-3504

VICE CHAIRMAN
David Wyse
Projects Unlimited
3680 Wyse Road
Dayton, OH 45414-2539

EXECUTIVE COMMITTEE
Margaret Drake
Univ. of Texas
Health Science Center
7703 Floyd Curl Drive
San Antonio, TX 78284

Jeffrey Jalbert
J C C
P.O. Box 381
Granville, OH 43023

Lowell LeFebvre
Sytek, Inc.
19 Church St.
P.O. Box 128
Berea, OH 44017

Robert McQueen
Knoll Pharmaceuticals
MIS Department
30 North Jefferson Road
Whippany, NJ 07981

Betsy Ramsey
Catholic University of America
Computer Center
Washington, DC 20064

David Schmidt
Management Science Associates
6565 Penn Avenue
Pittsburgh PA 15206-4490

E.F. Berkley Shands
Washington University
Dept. of Computer Science
Campus Box 1045, Bryan 509
St. Louis, MO 63130-4899

SYMPOSIA COORDINATOR
Betsy Ramsey
Catholic University of America
Computer Center
Washington, DC 20064

SYMPOSIA COORDINATOR, ASST.
Michael Carullo
Westinghouse Electric Corp.
P.O. Box 746
M/S 432
Baltimore, MD 21203

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
Glenn Everhart
25 Sleigh Ride Road
Glen Mills, PA 19342

COMMUNICATIONS COORDINATOR
G Beau Williamson
Rockwell International
1200 N. Alma Road
M/S 406-280
Richardson, TX 75081

NEWSLETTER EDITOR
David Santisteven
Western Technologies
P.O. Box 5542 TA
Denver, CO 80217

VAX NOTES SYSTEM MANAGER
Lawrence J. Kilgallen
Box 81, MIT Station
Cambridge, MA 02139-0901

SESSION NOTES EDITOR
Ken Johnson
Meridien Technology
P.O. Box 2006
St. Louis, MO 63011

BOOTBLOCK EDITOR
John L. Prather
Technicon Instruments Corp.
511 Benedict Avenue
Tarrytown, NY 10591

BOOTBLOCK STAFF
Richard DeJordy
American Mathematical Society
201 Charles Street
Providence, RI 02904

STORE COORDINATOR
Len M. Struttmann
Rockwell International
Collins Govt. Avionics Div.
M/S 153-100
400 Collins Road, N.E.
Cedar Rapids, IA 52498

MASTER'S LIST COORDINATOR
Carl Friedberg
Rocket Science Inc.
165 William Street, 9th Floor
New York, NY 10038-2605

SYSTEM IMPROVEMENT REQUEST
David Schmidt
Management Science Associates
6565 Penn Avenue
Pittsburgh, PA 15206-4490

VOLUNTEER COORD.
Ron Tencati
Jet Propulsion Lab
4800 Oak Grove Drive
MS:602-145
Pasadena, CA 91109

SPRING CAMPGROUND COORD.
Glen S. Johnston
General Dynamics
Mail Zone 5402
P.O. Box 748
Fort Worth, TX 76101

FALL CAMPGROUND COORD.
Thomas Linscumb
Computation Center
University of Texas
Austin, TX 78712

WORKING GROUP COORD.
Lowell LeFebvre
Sytek, Inc.
19 Church St.
P.O. Box 128
Berea, OH 44017

PRODUCTION SYSTEMS W/G CHAIR
E.F. Berkley Shands
Washington University
Department of Computer Science
Campus Box 1045, Bryan 509
St. Louis, MO 63130-4899

DECNET SECURITY W/G CHAIR
Ron Tencati
Jet Propulsion Lab
4800 Oak Grove Drive
MS: 602-145
Pasadena, CA 91109

INTERNALS W/G CHAIR
VMS USER'S NETWORK W/G CHAIR
Jamie Hanrahan
Simpact Associate
9210 Sky Park Court
San Diego, CA 92123

INTERNAL W/G CO-CHAIR
Allen Watson
Watson Consulting Inc.
3 River Street Ext., Apt. 30
Little Ferry, NJ 07643

SMALL SYSTEMS W/G CHAIR
David Mehren
Integra Systems Corporation
P.O. Box 40341
Tucson, AZ 85717-0341

MIGRATION AND HOST DEV.
VAXINTOSH W/G CHAIR
Jim Downward
KMS Fusion Inc.
P.O. Box 156D
Ann Arbor, MI 48106

MULTIPROCESSOR W/G CHAIR
Eugene Pal
U.S. Army
CAORA (ATORCATC)
Fort Leavenworth, KA

PERFORMANCE W/G CHAIR
John T. Peterson
Data Metrics Systems Inc.
5270 Lyngate Court
Burke, VA 22015

REAL TIME/FOREIGN DEVICES W/G CHAIR
Larry Robertson
Bear Computer Systems
56512 Case Avenue
North Hollywood, CA

SECURITY W/G CHAIR
C. Douglas Brown
Sandia National Labs
Division 2644
P.O. Box 5800
Alburquerque, NM 87185-5800

SYSTEM MANAGEMENT W/G CHAIR
Steve Tihor
251 Mercer Street
New York, NY 10012
BITnet: TIHOR@NYUACF
Internet:TIHOR@Acfcluster.NYU.EDU
TIHOR@NYU.EDU
UUCPnet:cmc12!tihor

VAXCLUSTER W/G CHAIR
Thomas Linscomb
Computation Center
University of Texas
Austin, TX 78712

ADVISORS
Joseph Angelico
110 Anthony Drive
Slidell, LA 70458

Ken Al Coar
Digital Equipment Corp.
P.O. Box 27320
721 Emerson Road
St. Louis, MO 63141

Jack Cundiff
Horry-Georgetown Tech. College
P.O. Box 1966
Conway, SC 29526

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

Clyde T. Poole
The University of Texas at Austin
Dept. of Computer Sciences
Taylor Hall 2.124
Austin, TX 78712-1188

Al Siegel
Battelle Memorial Institute
505 King Avenue
Columbus, OH 43201-2693

DTR/4GL SIG Fall 1989
Special RALLY PIR Ballot

DECUS Membership Number: __ __ __ __ __ __ __
(vote not valid unless this is a valid membership number)

CPU Classes (Check all that apply):

      Large cluster__ LAVC__ microVAX__ workstation__

Application Types at your site (Check all that apply):

__ Business EDP/MIS	__ Software Development
__ Education	__ Engineering/Scientific
__ Office Automation	__ Service Bureau
__ Other (specify) _____	

Number years using computers: __   Number of years using 4GLs__

Products Used (Check all that apply):

__DTR-11	__VAX-DTR	__CDD	__CDD/Plus	__TDMS
__FMS	__DBMS(any)	__Rdb	__RALLY	__TEAMDATA
__DECreporter	__Accent-R	__Cortex	__FOCUS	__Ingres
__Oracle	__Powerhouse	__Smartstar	__DECwindows	
__Other (Specify) _____				

See October 1989 Issue of the Wombat Examiner and 4GL Dispatch
for PIR detail and instructions

Total of **50** points
Maximum **10** points *per PIR*
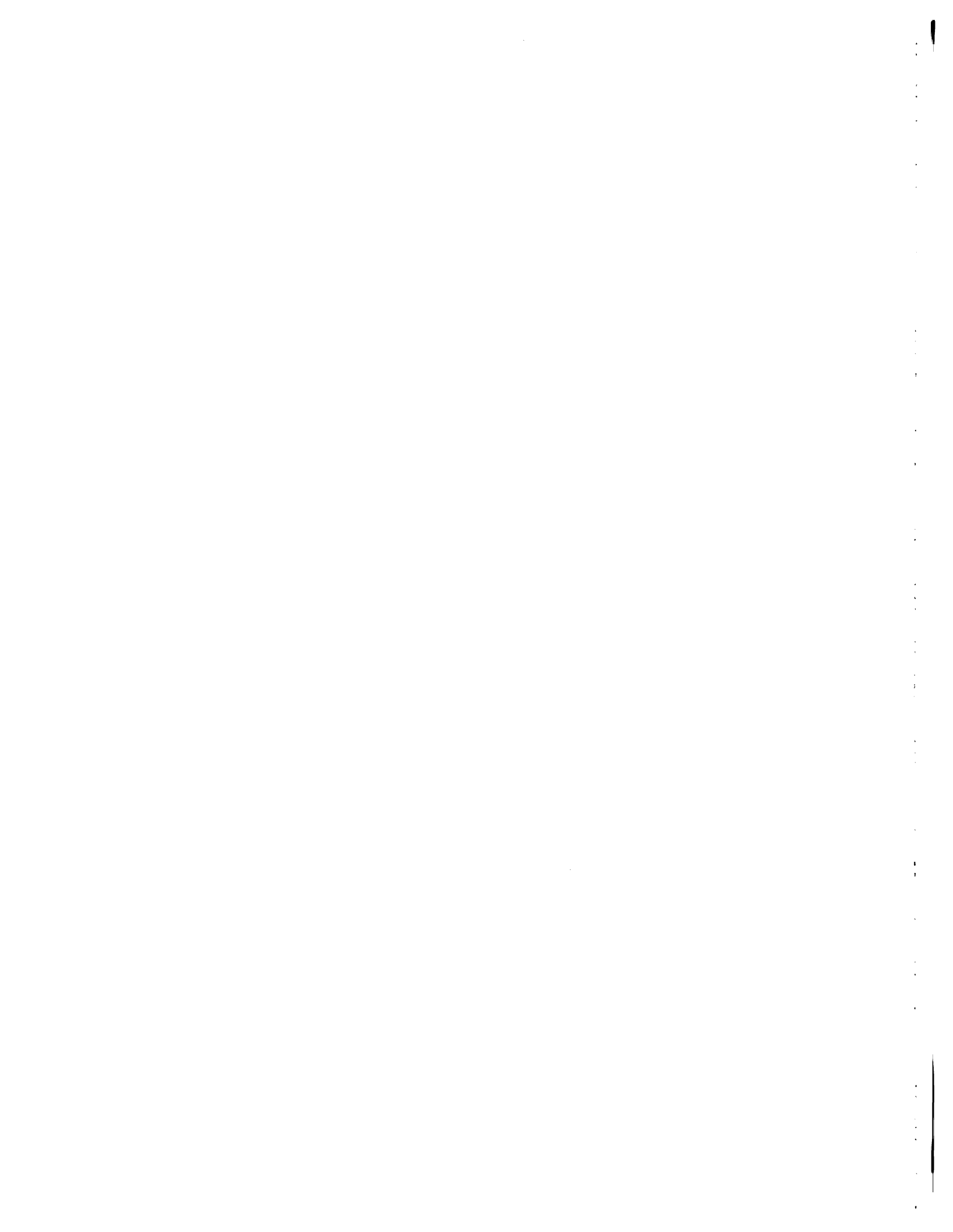
PIR Number	Points	PIR Number	Points
F89- 1	__	F89-16	__
F89- 2	__	F89-17	__
F89- 3	__	F89-18	__
F89- 4	__	F89-19	__
F89- 5	__	F89-20	__
F89- 6	__	F89-21	__
F89- 7	__	F89-22	__
F89- 8	__	F89-23	__
F89- 9	__	F89-24	__
F89-10	__	F89-25	__
F89-11	__	F89-26	__
F89-12	__	F89-27	__
F89-13	__	F89-28	__
F89-14	__	F89-29	__
F89-15	__	F89-30	__

Return your ballot to be received by <u>December 15, 1989</u>, to:

      T. C. Wool
      E. I. duPont
      Engineering Department
      P. O. Box 6090
      Newark, DE 19714-6090

QU

Electronic Publishing (E-Pubs)
Software Improvement Request and Wishlist Form

Name: .... Company: ....

Address: .... Phone: ....

....

The E-Pubs UIG is concerned with Digital and third party hardware/software products in the electronic publishing arena. What product does your request or suggestion concern? Please include the software version number where appropriate. Please reference only one product per form.

....

If your request or suggestion does not relate to a product, please check which of the following E-Pubs UIG topics it does concern:

Newsletter .... ☐   Symposium Sessions ... ☐   UIG Tape Submission .. ☐   Session Notes .... ☐

Information Folder ..... ☐   Working Group .... ☐   Pre-symposium .... ☐   DECUS Store Items ... ☐
                              Activities   Seminars

Other .... ☐   ....

How to write a request:
Please explain your request thoroughly. Don't assume that we know how something is done in "XYZ" product or in another SIG. Justify why the capability would be useful and give examples.

Brief description: ....

....

Complete description with examples: ....

....

....

....

....

....

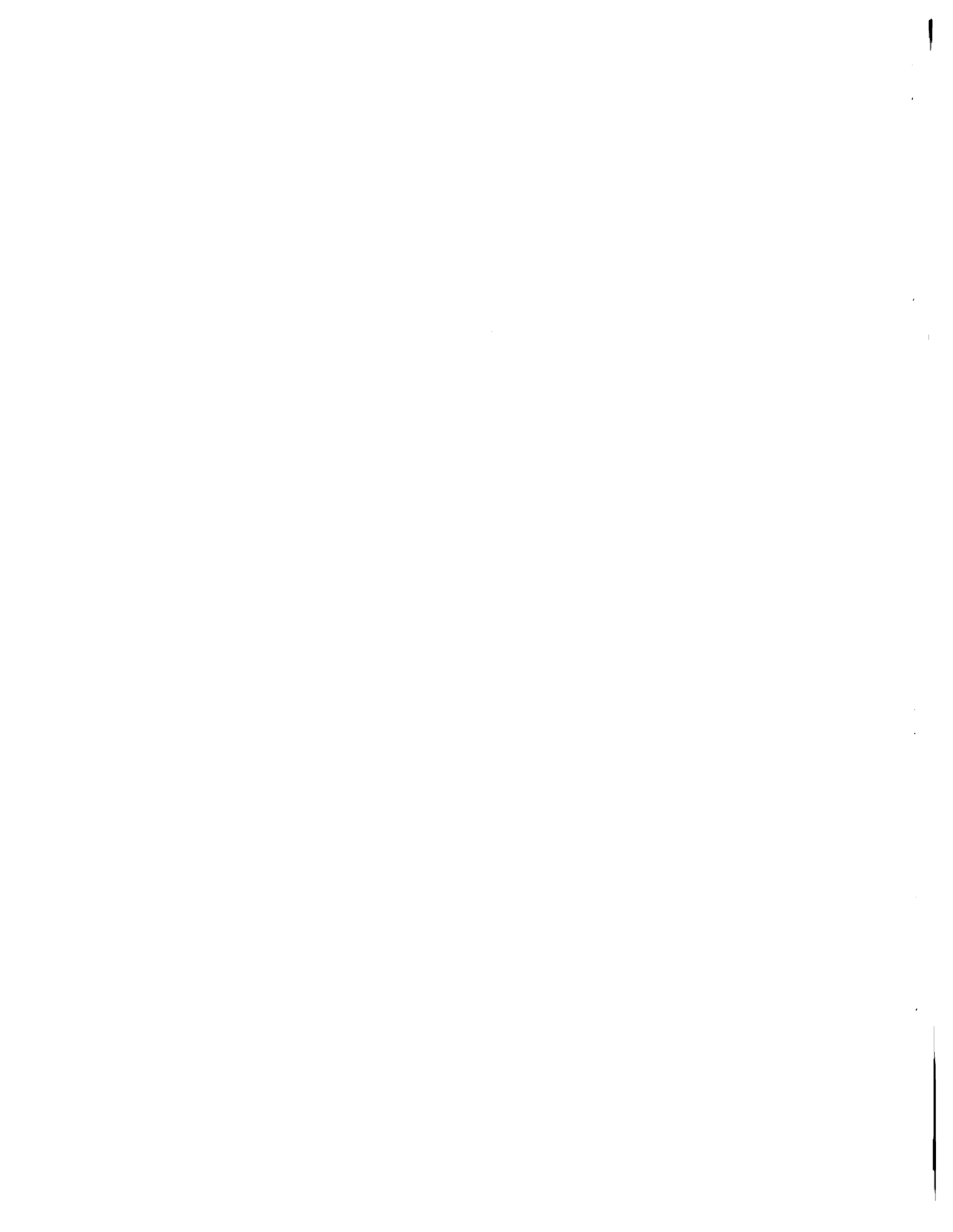..

At Symposia, return this form to the E-Pubs campground or submit at a Wishlist session. To mail, send to:

Patty English-Zemke, 87 Deerhurst Dr., Goleta, CA 93117

H M S   S I G

HARDWARE SUBMISSION FORM -- A SIG INFORMATION INTERCHANGE

Message

Contact
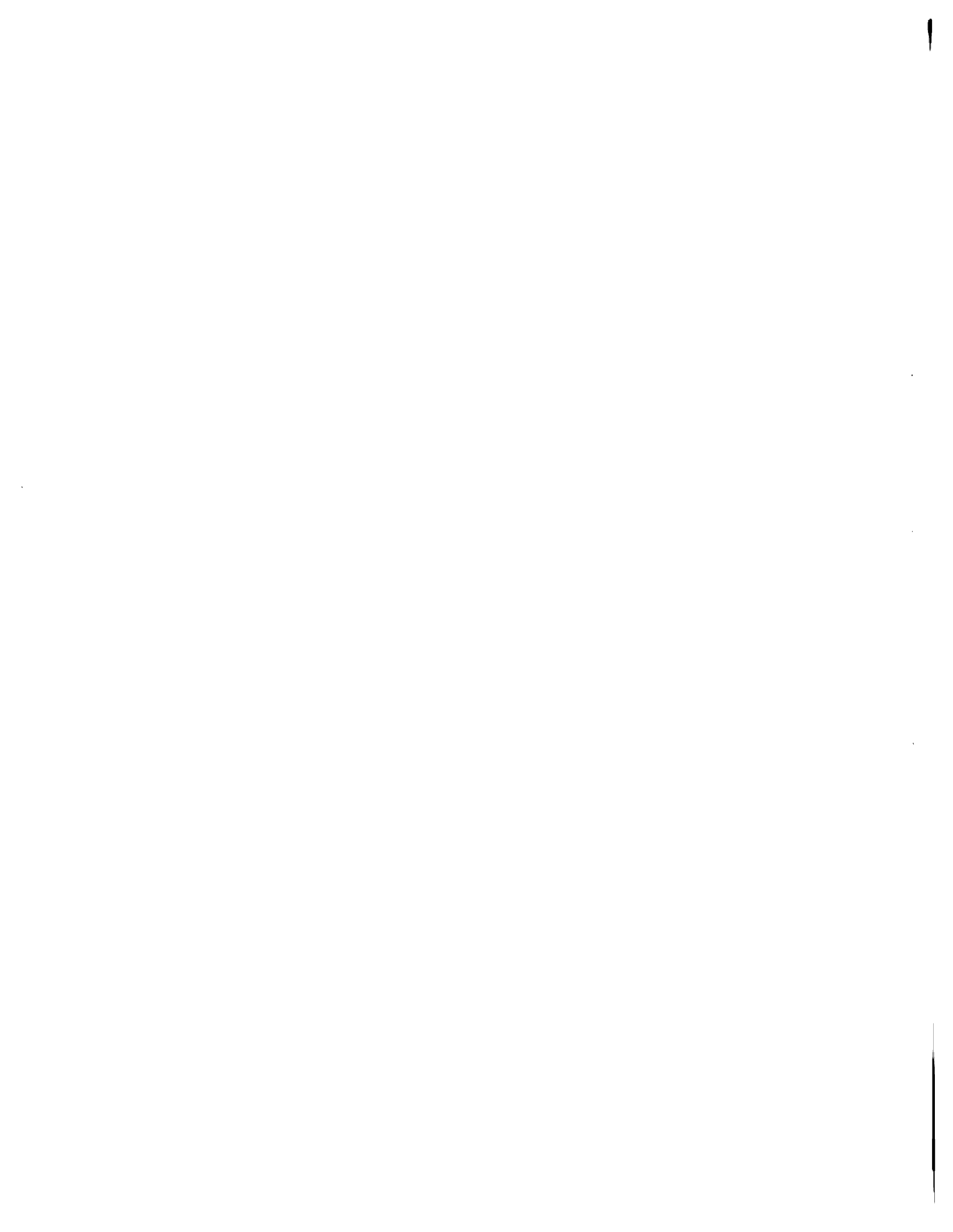Name

Address

Telephone

Type of equipment

SUBMIT ANY TYPE OF HARDWARE PROBLEMS AND/OR FIXES.

SEND TO:

William K. Walker
Monsanto Research Corp.   OR   Neil Krandall
P.O. Box 32 A-152   ==   Univ. of Cincinnati
Miamisburg, OH 45342   Pharamacology & Cell
                                             Biophysics
                                             231 Bethesda Ave MC575
                                             Cincinnati, OH 45267
                                             (513)872-4788

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
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):

Tear out or photocopy reverse to submit an SIR.

Dave Schmidt
Management Science Associates
6565 Penn Avenue
Pittsburgh, PA 15206-4490
USA

_____

DECUS membership number _____ (six digits)

Our site uses/runs VAX cpus with the following (check all that apply)
___ Unix	___ Clusters (any)	___ Dumb Terminals
___ Stand alone	___ Uni-processor	___ VTxx terminals
___ Networked	___ SMP-processor	___ Work stations
___ 1 VUP or less	___ 2 to 3 VUPS	___ PC's (any)
___ 4 to 6 VUPS	___ 7 or more VUPS	___ Other _____

We use VAX's in the following applications (Check all that apply)
___ Business EDP	___ OLTP	___ Application programming
___ Education	___ CAD/CAM/GRAPHICS	___ Scientific/Engineering
___ Business Management	___ Office automation	___ Data Acquisition/Control
___ Systems programming	___ Research	___ Telecommunications
___ Other _____		

I support the following as the most important System Improvement Requests. (List from zero to fifteen SIR's):

I oppose the following SIR's as detrimental. (List from zero to five SIR's):

SIR Number:

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

SIR Number:

_____

_____

_____

_____

_____

MAIL TO:

Dave Schmidt

Vax Systems SIG SIR Coordinator

6565 Penn Avenue at Fifth

Management Science Associates

Pittsburgh, Pennsylvania 15206

(By September 22, 1989 to be counted!)
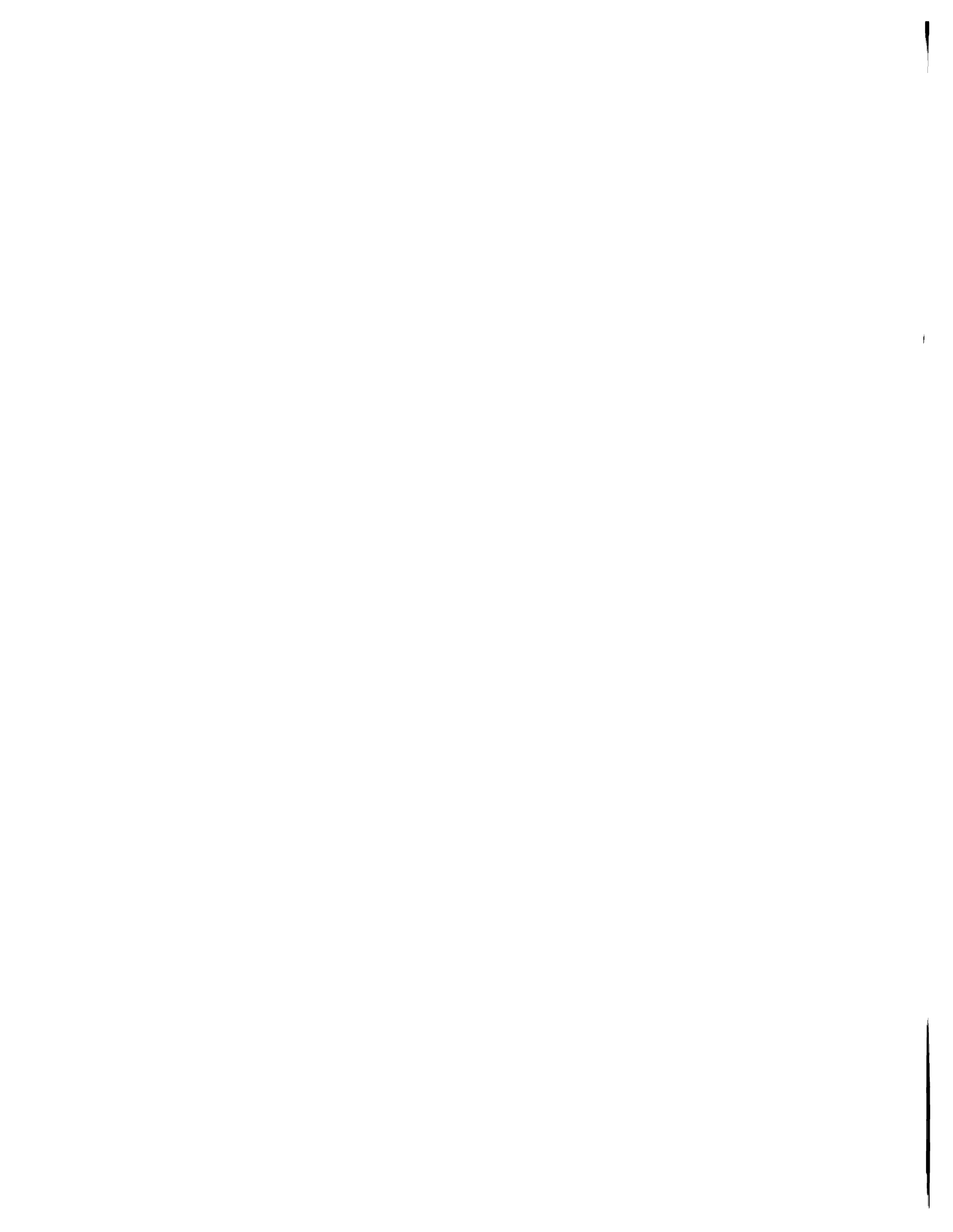
DECUS U.S. CHAPTER
SUBSCRIPTION SERVICE SIGS NEWSLETTERS ORDER FORM
(U.S. Members Only)

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 '89 Proceedings (SP9)	15.00		
Fall '89 Proceedings (FA9)	15.00		
Spring '90 Proceedings (SP0)	15.00		
Fall '90 Proceedings (FA0)	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 01752-4605, (508) 480-3659.

FOR DECUS OFFICE ONLY
Check Number _____ Bank Number _____
Amount $ _____

S&M
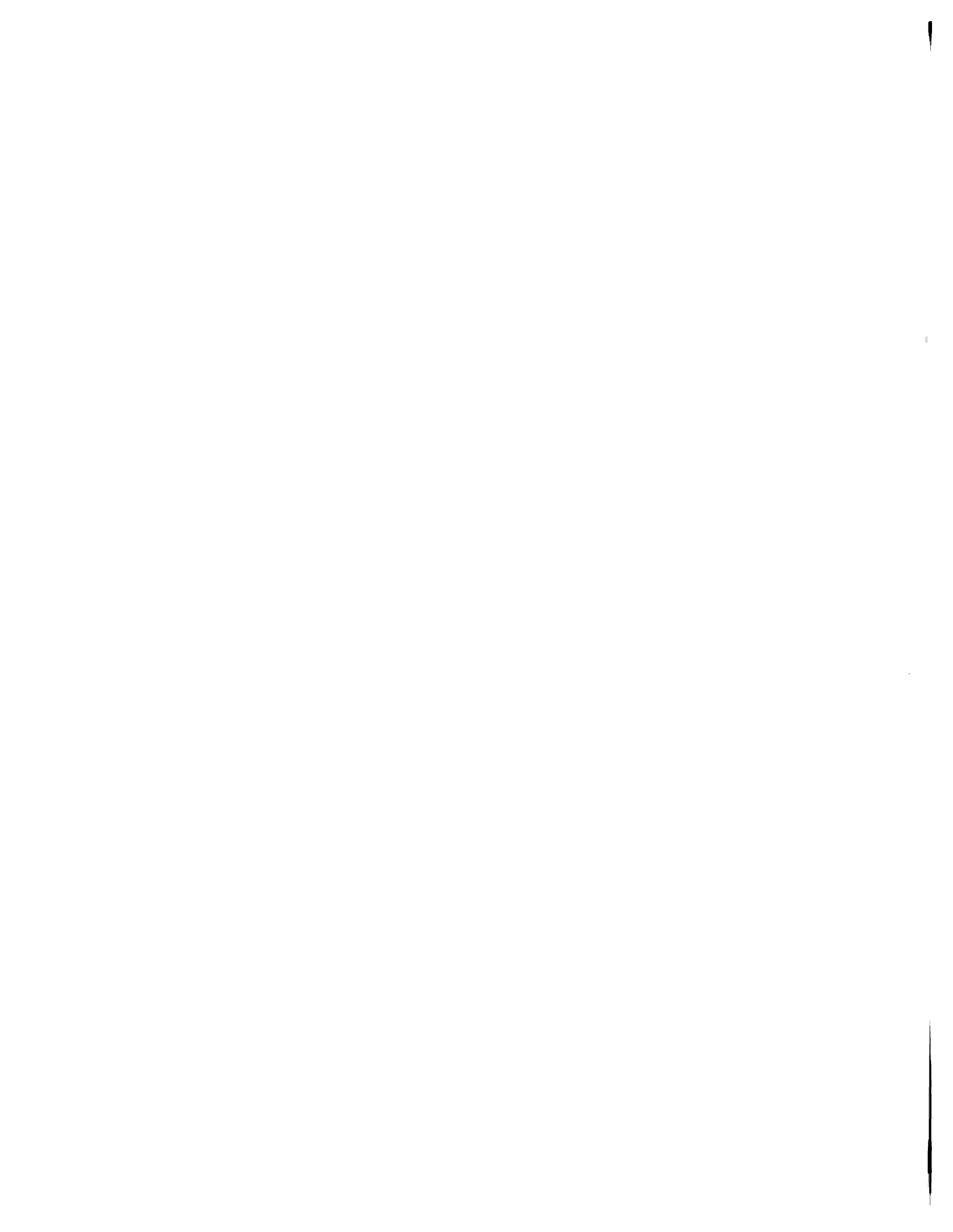
DECUS U. S. Chapter Application For Membership

IMPORTANT! Please provide a complete mailing address, include zip code in accordance with postal regulations for your locality.

☐ New Membership   ☐ Update to Current Membership Profile

Current DECUS Membership Number _____

Do you wish to be included in mailings conducted by Digital (for marketing purposes etc.?) ☐ Permission   ☐ Refusal

Please print clearly or type!

Name
_____

Company
_____

Address
_____

_____

_____

City _____ **State** _____ **Zip** _____

Phone: Home ( ) _____ **Business** ( ) _____

Are you an employee of Digital Equipment Corporation?

☐ Yes   ☐ No

1. How did you learn about DECUS? (check applicable item)

1☐ Another DECUS Member
13☐ Local Users Group
5☐ Hardware Pkg.
8☐ DECUS Chapter Office
12☐ Advertising
7☐ Software Dispatch (Digital Newsletter)
4☐ Digital Sales
2☐ Symposia
14☐ Special Interest Group
6☐ Software Pkg.
10☐ Digital Store

2. 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 Equip.)
26☐ Software Development
27☐ Communications & Networking
28☐ Systems House, VAR/OEM
29☐ Consultant
30☐ Other DP Services

3. I wish to participate in the following DECUS U.S. Chapter Special Interest Group(s):

3☐ Artificial Intelligence
7☐ Business Applications
6☐ Data Management Systems
5☐ DATATRIEVE/4GL
8☐ Education
9☐ Electronic Publishing
10☐ Graphics Applications
11☐ Hardware and Micro
16☐ Languages and Tools
14☐ MUMPS
31☐ DAARC (Data Acquisition, Analysis, Research, and Control)

15☐ Networks
34☐ Office Automation
36☐ Personal Computer
18☐ RSTS
17☐ RSX/IAS
19☐ RT-11
21☐ UNIX
26☐ VAX Systems
32☐ Site, Mgmt. & Training

4. Using the classification numbers from question 3, please indicate which SIG would be the primary focus for your interests?
#_____

5. Using the classification numbers from question 3, please indicate which SIG would be of secondary focus for your interests?
#_____

6. Total employees in entire company/institution/government department: (check one)

2☐ 10,000 or More
3☐ 5,000 to 9,999
4☐ 1,000 to 4,999
5☐ 500 to 999
6☐ 250 to 499
7☐ 100 to 249
8☐ 6 to 99
9☐ Fewer than 6

7. 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 _____

8. Citizen of the United States? ___Yes ___ No

If no: Country _____

Forward to: Digital Equipment Computer Users Society
Membership Group
219 Boston Post Road, BP02
Marlboro, MA, 01752-4605
Phone: (508) 480-3635

Signature
_____

rev: 07/89

S&M-3

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

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 01752-4605

USA