THE HEAP

STRUCTURED LANGUAGES SIG NEWSLETTER

Volume 6, Number 1 **May 1983**

This is the first issue of the Structured Languages SIG Newsletter in quite some time. Our former editor Roy Touzeau could not continue on as editor due to his changing jobs (he is now out in the real-world making money) which has taken him away from DECUS. I should have produced a newsletter last year, but there really haven't been any submissions to speak of let alone print. So I am going to start this newsletter with a plea for help and some ideas for getting the newsletter back on its feet. First of all, if you want to respond to my plea, my address is:

John R. Barr
Department of Computer Science
University of Montana
Missoula, MT 59812

With the advent of paid subscriptions for newsletters, I am going to produce (with the help of the MUMPS SIG newsletter editor) at least four issues of this newsletter next year. We are going to maintain a schedule which will force us to collect newsletter articles and publish the newsletter immediately after each DECUS Symposium so those of you who didn't make it get a chance to read about the meetings and at least a month before each symposium to let you know what you will be missing if you don't go to the next one. In addition, I want to make the newsletter something you want to read. Some of the sections in this newsletter should give you an idea of what will be in future newsletters. Let me know what you would like to see, and I will attempt to get it published. Let me know what you don't like to see and I will attempt to do a better job the next time around.

It is assumed that all articles submitted to the editor of this newsletter are with the authors' permission to publish in any DECUS publication. The articles are the responsibility of the authors and, therefore, DECUS, Digital Equipment Corporation, and the editor assume no responsibility or liability for articles or information appearing in the document. The views herein expressed are those of the authors and do not necessarily express the views of DECUS or Digital Equipment Corporation.

UNIX is a trademark of Bell Laboratories.

Our production schedule for the next four newsletters is as follows:

August 1, 1983 - Deadline for article submissions.
August 10, 1983 - Finished newsletter submitted to DECUS.
September 1, 1983 - Newsletter should be mailed.

November 12, 1983 - Deadline for article submissions.
November 20, 1983 - Finished newsletter submitted to DECUS.
December 15, 1983 - Newsletter should be mailed.

February 1, 1984 - Deadline for article submissions.
February 10, 1984 - Finished newsletter submitted to DECUS.
March 1, 1984 - Newsletter should be mailed.

April 15, 1984 - Deadline for article submissions.
April 25, 1984 - Finished Newsletter submitted to DECUS.
May 15, 1984 - Newsletter should be mailed.

Note that is usually takes 2-4 weeks for the newsletter to make it through the mail to you. The earlier you get submissions to me the easier it will be for me to make the deadlines.

With this newsletter, it has now become "The Heap", and the next newsletter will be published with "The Tree" which is the MUMPS SIG newsletter. I also have a new slogan "Pascal programmers do it in 'The Heap'", and I will be making up some buttons which say "I did it in 'The Heap'" for those of you who submit articles to "The Heap" in the future. It sure beats doing it in the pool.

Also at the end of this newsletter is the menu for the Structured Languages SIG. Alan Folsom, our menu coordinator, has produced the first menu for the SIG. We would like to get your responce so we can present it in Las Vegas in October. Please get your menus back to Alan as soon as possible. He will be sending the results off to DIGITAL as soon as he gets them talleyed. DIGITAL and the SIG leadership will then review the results and DIGITAL comments for presentation in Las Vegas.

John Barr, Editor

FROM THE CHAIR

As the new Chair of the Structured Languages Special Interest Group, I would like to fill all of the interested members in on the events that have transpired since last you heard from the SIG steering committee. We have had a period of uncertainty about our purpose in life, from which I hope we are beginning finally to emerge. I'd like to take this opportunity to report my observations to you on where the SIG has been, where we stand now, where we might be going, and how it has been proposed that we get there.

It must be acknowledged that we've had paper sessions aplenty in the last few DECUS Symposia, together with considerable participation in BOF and rap sessions. This high level of activity, however, has not been productive of the general orientation we must have in order to undertake long-range planning on behalf of our membership. We have now an infusion of help from Digital in the form of new and long-awaited products designed to serve the ends of modern software development. A noteworthy example of these are the DEC/CMS, MMS software management packages.

This change on Digital's part gives us, the Structured Languages SIG, a much more sharply-defined mission in life, namely, to keep Digital pointed along the path of desirable new product development and existing product maintenance. In fact, it is of great importance that each of you who comes to St. Louis comes prepared to comment on the menu which is included in this newsletter. We may not be able to get the menu out soon enough to permit us to get your votes by mail, and thus to get formal responses on the menu items from Digital, but you may be sure Digital will be there to hear our input on those items. Of course, the leadership of the SIG also will be there, and eager to get your input on the things that you think the SIG can do that it's not now doing, or things that you wish we'd finally stop doing.

I have proposed the view, most recently in the rap session held at the symposium in Anaheim, that the members of our SIG are really interested in all those things which contribute to the efficient development of software; this more general category includes not only structured languages, but all the modern development tools, some examples of which have become available under the aegis of the VAX/VMS VNX program recently announced by Digital. As a part of that program, Digital has elected to offer UNIX (UNIX is a trademark of American Bell these days, I think) as a supported product. This has given the SS & OS SIG a chance to more clearly identify the primary interest of its members, which they will do by changing their SIG's name to the UNIX SIG (properly identifying the ownership of UNIX, of course). We can likewise clarify the area we believe to be of major interest to us with an appropriate name change; I have proposed that we henceforth be known as the "Languages and Tools SIG" (LTSIG), with that goal in mind.

As is the case with most SIGs, LTSIG has a steering committee, composed of members who serve as long as they're not drummed out of office. These are (with their current occupants)

- Chair - James W. Livingston, Jr., Measurex Corporation
- Newsletter Editor - John R. Barr, University of Montana
- Symposium Coordinator - Vacant
- Software Librarian - James Triplett, Intermetrics
- Menu Coordinator - Alan L. Folsom Jr.,

I would propose that the organization of our SIG, further, should reflect the realities of Digital's marketing and product development groups, to the extent that this does no violence to our own priorities and interests. With that objective in mind, I suggest that we use interest area coordinators to keep us involved in all the areas that are legitimately within our purview. The interest areas that seem appropriate fall into three general categories: Operating Systems, Languages, and Development Environments. Clearly, there is overlap amongst the areas shown below; that might well be a good thing, since members may wish to serve in multiple capacities. If you see one you like, please volunteer for it!

Interest Areas

1. Operating Systems (Alphabetically)
   - Large Systems
   - RSTS
   - RSX-11M/IAS
   - RT-11/TSX
   - UNIX
   - VAX/VMS

2. Languages (Also alphabetically)
   - Ada
   - BLISS
   - C
   - FORTRAN
     * FLECS

        - RATFOR

- Modula
- Pascal
- PL/I

3. Environments

- Ada Development Environments (Nothing from Digital yet, but we might want to put in our requests early)
- Software Tools VOS (as maintained by Software Tools Users Group) - Note that I'm enrolling the SIG in STUG.
- VNX

Coordinators for each interest area under these major headings would be responsible for keeping the Languages and Tools SIG updated on developments in their areas. They would do this by maintaining contacts within their interest areas, writing newsletter articles, summarizing symposium presentations, and presenting papers at symposia. Ideally, the interest area coordinators for the operating systems will be members of the SIG for that operating system. With a little good fortune, it might be possible to get the operating system SIGs to set up LTSIG coordinator roles, which could be filled by our corresponding LTSIG coordinator. Other organizational developments which can serve our needs will be tried as they are proposed, so please feel obliged to suggest some that you think will be of help to us all. In the meantime, please volunteer to serve as one or more of the coordinators listed above; we need member participation, desperately! You can volunteer by accosting one of the steering committee members at the St. Louis Symposium, where each of them attending will be wearing a banner on his badge, or by writing or calling me at work:

James W. Livingston, Jr.
Measurex Corporation
One Results Way
Cupertino, CA 95014
(408) 255-1500 x4468

I hope that this ramble has communicated something of the interest, enthusiasm, and committment with which I come to the SIG chairmanship. There are many more topics on which I think discussion is both appropriate and desirable. I hope that we can discuss them fully, both at St. Louis and in the pages of this newsletter's future issues. I believe that our SIG has offered much to its members in the past. The fact that we're the sixth largest SIG in DECUS (of 22), with 8,226 members (as of January 4, 1983), must mean that the members of our SIG believe that something called "Structured Languages SIG" has potential for serving their interests. I am strongly committed to finding out exactly what our members want, and helping in every way I can to bring it to them.

Interesting Structured(?) Software

In each issue of the newsletter I will be including some sample programs which may be of interest to you. If you have any short examples of programs which would fit into this section, please send them to me so others can gain.

This month's example program is a result of a problem we had with the conversion of over 75,000 lines of FORTRAN code from an IBM system to Unix (which is of course a trademark of Bell Laboratories). The printed output consisted of reports intended to be FORTRAN print files with the carriage control in the first column. Of course Unix does not give a hoot about FORTRAN carriage control conventions, so we wrote the filter below which will accept the FORTRAN print files as input and output a text file with embedded line-feed, form-feed, and carriage returns which would allow it to printed on our serial printer. The filter is written in C (not one of my favorite languages) and is an example of C control structures and the relative ease with which tools can be implemented on Unix.

```
# include <stdio.h>
main()
{  int c;
   /* Get rid of first form feed if it is there. */
   if ((c = getchar()) != '1') ungetc(c,stdin);
   while ((c = getchar()) != EOF) {
      switch(c) {
        case '1': putchar('\f'); putchar('\n'); break;
        case '0': putchar('\n'); putchar('\n'); break;
        case '+': putchar('\r'); break;
        /* Special case for empty lines. */
        case '\n': putchar(c); ungetc(c,stdin); break;
        default:
           putchar('\n');
           /* Tabs must be printed. */
           if (c == '\t') putchar(c);
           break;
        }
      while ((c = getchar()) != '\n') putchar(c);
      }
   putchar('\n');
}
```

This next program, again a filter on Unix, was used on the same project to automatically extract a file from an ANSI standard magnetic tape. Its input is the text of the header record, and its output is a Unix dd command which will extract the file from the tape, perform the proper record deblocking, and produce a Unix text file.

```
program name(input,output);
  var i:integer; ch:char;
  procedure getnum;
  var j:integer;
  begin
    j:=5;
    while ch='0' do begin read(ch); j:=j-1 end;
    for i:=1 to j do begin
      write(ch);
      read(ch)
      end
  end {getnum};
begin {name}
  read(ch);
  if ch='v' then begin readln; read(ch) end;
  for i:=1 to 4 read(ch);
  write( 'dd if=/dev/rmtl2 of=' );
  repeat
    write(ch);
    read(ch);
  until ch=' ';
  readln;
  for i:=1 to 6 do read(ch);
  write( ' ibs=' );
  getnum;
  write( ' obs=' );
  getnum;
  writeln( ' conv=ascii' )
end.
```

The header records on an ANSI standard magtape contain two or three lines of information. The first record on the tape contains three lines, the first line being the volume label which is always skipped. The normal header records contain two lines which are the same as the second and third lines of the first record. The first line has the file name starting in the fifth column and is blank terminated. The second line contains the block size of the tape records in columns 7-11 with leading zeros, and the record size in columns 12-16 also with leading zeros. The resulting command line:

```
dd if=/dev/rmtl2 of=filename ibs=nnnnn obs=nn
```

is then executed using eval to perform the correct operation. More than one file can be read by repeating the following shell program which reads the next file from the tape. The tapes were in EBCDIC by the way.

```
        dd if=/dev/rmtl2 of=temp ibs=4000 cbs=80 conv=ascii,lcase
        name <temp
        eval 'name <temp'
        mt fsf 1
```

A header record from an ANSI standard magtape follows to give you a better idea what name does.

```
        vollseml3 0                             isdos
        hdrlabtlib.ibmbeta   seml3 00010001      82001 ...
        hdr2f04000008030sedr    /            b   00001
```

I hate to include too much Unix material, but we have found the environment provided by Unix condusive to the implementation of special tools that normally require delving into the operating system of most 'other' DEC-based operating systems. How would you go about extracting text files from an ANSI standard tape produced by an IBM system?

NBS Pascal Progress and Bugs

In this section of the newsletter, I would like to present material relavent to the implementation of the NBS Pascal compiler on DEC operating systems. If you have any bug reports and/or fixes, please send them to me so they can be published for others who use the compiler.

The NBS Pascal compiler is implemented on the RT-11 and RSX-11 operating systems, and can be run on RSTS under either the RT-11 or RSX-11 emulator. The most recent version of the compiler is V1.6i for both RT-11 and RSX-11. You may contact me (John Barr, (406) 243-2883) about the RT-11 version of the compiler. The RSX-11 version should be on the last Structured Languages SIG tape. Both versions are now using the same source files for the first and second passes, and both require the FPP instruction set for floating point operations. You can use the compiler on a machine which only has the EIS instructions as long as no floating point operations are used and you patch or rebuild both passes of the compiler and comment out the SETI and SETD FPP instructions in PASLIB.MAC.

I currently have a graduate student working on a four pass version of the NBS Pascal compiler. We are going to attempt to do some more optimization and handle larger programs at the same time. Portability of the compiler is also part of the design specification. I had hoped to get this version of the compiler ready by June, but it may not be ready until a year from June.

Those of you using V1.6h of the NBS Pascal compiler on RT-11 should appreciate this next item. It seems that my code for the allocation of space on the heap always added an extra word in between each item. The result being much wasted space, and consequently reducing the size of the programs which could be compiled since the size is dependent on the amount of heap space available. The correct code for the 'new' library procedure is included here:

```
$$$004::  ;procedure new( var item:itemptr );
        MOV     @#HGHMEM,R0     ;GET END OF HEAP
        MOV     R0,R1           ;SAVE VALUE
        INC     2(SP)           ;ROUND UP REQUESTED SIZE TO
        BIC     #1,2(SP)        ;... A MULTIPLE OF TWO
        BLE     HERR            ;VALID REQUEST??
        ADD     2(SP),R1        ;FIND NEW END OF HEAP
        CMP     R1,MTOP         ;SEE IF THERE IS ROOM
        BHIS    HERR            ;NO ROOM, ERROR
        MOV     R0,2(SP)        ;RETURN ADDRESS OF HEAP SPACE
        .SETTOP R1              ;GET THE SPACE
        CMP     R0,R1           ;DID WE GET THE SPACE??
        BNE     HERR            ;NO, ERROR
                                ;CONTINUE (ZERO OUT ITEM SPACE)
```

There a several other minor errors with the library routines which I will be correcting and including in the next issues of the newsletter.

NBS Pascal Library Routines

I am sure that many of you who are using the NBS Pascal compiler on RT-11, RSX-11 and RSTS/E have written some interesting library programs that others may be interested in using. If you have any library routines, send them to me so I can present them here in the newsletter. I would also like to generate a library disk/tape for submission to the DECUS library of a group of library programs for use with NBS Pascal. I almost have Kernighan and Plauger's primitives completed for NBS Pascal. Those of you who send me something for the newsletter on a floppy disk will get the primitives in return.

Symposia News

The next DECUS Symposium will be held in St. Louis May 23-27. If this newsletter gets to you in time you are one of the lucky ones. I am including this material just to indicate what type of information will be incuded in future newsletters. This information is not complete because the SIG does not have a regular Symposia Coordinator. We need one badly and you should contact James Livingston if you are interested. There are some benefits to being the Symposia Coordinator. Give Jim a call and he will let you know what is expected of the coordinator and what the coordinator gets in return.

Monday, May 23, 1983:

Structured Languages SIG Roadmap: 10:15am - An outline of SIG presentations and other activities will be presented to help you get through the week.

VAX Languages and Tools Overview: 11:45am - DIGITAL presentation of the VAX Language products and tools product currently available.

RATFOR as a Product Development Tool: 12:45pm - How RATFOR is being used for product development.

Overview of VAX Language Environment: 5:30pm - DIGITAL presentation of the VAX language environment which allows users to mix or match programming languages.

Ada Language System: 6:30pm - DIGITAL presentation about Ada. With little information about this session, I can't tell you any more at this time.

Recursive Programming Using VAX-11 Pascal: 8:00pm - DIGITAL presentation on how to use Pascal to implement recursive programs.

Tuesday, May 24, 1983

Overview of Pascal for RSX Users: 10:15am - DIGITAL presentation on Pascal for RSX-11 systems. This session was canceled last Fall, but will make it this time. DIGITAL will be anouncing their new Pascal compiler for RSX-11. Something we have been waiting for for almost 10 years.

Programming in Pascal for RSX Users: 11:15am - DIGITAL presentation on how to use Pascal in the RSX-11 environment. This should be interesting for RSX-11 systems programmers who have been waiting for a decent pregramming language.

Introduction to Pascal-10/20: 2:15pm - DIGITAL presentation on Pascal for 10s and 20s. This was also canceled last Fall. I think it will make it this time.

VAX Software Development Tools Panel: 3:30pm - DIGITAL panel which will be discussing the tools offered by DIGITAL and how they can be used for the development of software.

Wednesday, May 25, 1983: (You get a day off from languages)

Thursday, May 26, 1983

EDT for Structured Languages: 11:15am - How you can use EDT to help you produce structured programs (better formats).

Structured Languages Business Meeting: 12:45pm - General SIG business meeting to discuss the goals and issues within the SIG.

Systems Programmming in VAX-11 Pascal: 12:45pm - Same time different place. You have to make a choice here of whether you want to learn how DIGITAL uses Pascal for systems programming or help the SIG get things going. This is part of our scheduling problems.

Interfacing I/O Devices Using Pascal: 3:30pm - I assume this is dealing with DIGITAL's Micro Power/Pascal.

Design of a Falcon Based Micro Power/Pascal: 5:00pm - The title is probably a little deceptive. The word application is missing.

Micro Power/Pascal Futures: 6:00pm - What is coming in the future from the little Pascal people.

VAX-11 Pascal I/O Tutorial: 5:30pm - DIGITAL presentation on how to do various types of I/O operations from VAX-11 Pascal. It should include how to access the various types of file structures.

My Language is Better Than Your Language: 8:30pm - This is the SIGs magic session. We are going to attempt to compare languages by posing programming problems solved in one language and see if someone else can do it better using their favorite language.

Friday, May 27, 1983

Writing C Code for VAX/VMX and UNIX Systems: 9:00am - If you are writing code in C which may be used on both VMS and UNIX, this session will help you understand the differences and common features of the two operating systems and C language compilers.

How to Write (Non) Portable C: 10:15am - DIGITAL presentation on how easy it is to write non-portable C code and how you can write portable C code.

Simulation Using Pascal: 11:15am - User paper describing how Pascal language was used to write a simulation program.

Structured Languages Wrap Up: 11:45am - The last SIG meeting of the symposium where you will have a chance to tell the leadership where there were problems and what needs to be done for the next symposia. Be there if you want to help the SIG.

STRUCTURED LANGUAGES SIG WISHLIST

The first wishlist of the Structured Languages Special Interest Group is divided into three sections. The first is a short questionaire which will provide information regarding the nature of the responding installations. Please take the time to fill this out, and return it with the answer sheet.

The wishlist itself is divided into two parts, the first being items directed at DEC, i.e., problems or requests which you feel DEC should respond to. The second part consists of items directed at the SIG. Since the SIG maintains software and addresses issues which are not of direct concern to DEC, this provides a means of informing the SIG leadership of the directions you feel the SIG should pursue.

In each section, you have a maximum of 25 votes, no more than five to be given to any one item. You may give five votes to each of five items, or you may spread your votes thinner, but no item may have more than five votes. In addition, it is possible to "write in" an item, and if several "write in"s seem to address the same issue, they will be combined and treated as a single item when the votes are tallyed.

Please return your answer sheet and questionaire as early as possible, but no later than June 30th, to:

Alan L. Folsom, Jr.
Systems Center - Dept. 436
Fischer & Porter Co.
200 Witmer Road
Horsham, Pa. 19044

DEC DIRECTED MENU ITEMS

0.1 DEC should provide a consistant, standardized structured language, and support it across all operating systems and CPU's.

0.2 Structured languages should be provided for 10/20 machines.

0.3 Structured languages should be provided for the PDP 11 computers.

0.4 A wider range of languages should be supported for RT-11.

0.5 DEC should provide Cross Compilers for the various new 16 bit micros, to facilitate program development.

0.6 Dec should provide more closely coupled compilers and debuggers, to facilitate program development in high level languages.

0.7 A common set of debugging tools should be developed, providing a unified interface across operating systems and languages.

0.8 DEC should announce when software such as loaders are changed, so that modifications can be made to SIG or customer supported software.

0.9 A standard "C" should be provided across all operating systems.

0.10 DEC should provide and support a "C" compiler for the DEC 10/20 machines.

0.11 DEC should provide and support the "C" language for RSX systems.

0.12 The "C" language should be available for the new personal computer lines.

0.13 The "C" language should be available for programming the LSI-11 machines, perhaps as a Cross Compiler package on larger CPU's.

0.14 DEC should put the DECUS "C" compiler on the distribution kits, in the same manner as TECO.

0.15 A standard Pascal should be provided across all operating systems.

0.16 Vax-11 Pascal Global variables should be fixed to allow sharing or non-sharing across seperately compiled modules.

0.17 DEC should support Pascal for RT-11.

0.18 DEC should support Pascal for RSX.

0.19 DEC should provide a Pascal Compiler for the LSI-11, which would run with limited memory. (64k)

0.20 DEC should support Pascal under RSTS, allowing linkage to Macro or Basic+2 object modules.

0.21 PDP 11 Pascal should be provided, allowing linking to RMS 11.

0.22 Fortran 77 should be supported under RT-11.

0.23 DEC should provide customer BLISS courses.

0.24 Modula 2 should be supported under RSX

0.25 Modula 2 should be supported under VMS

0.26 Modula 2 should be supported for the new Personal Computer lines.

0.27 DEC should make an ADA package availble for the VAX.

0.28 DEC should make an ADA package availble fo the PDP 11.

SIG DIRECTED MENU ITEMS

0.1 The SIG should provide an organized method of feedback to DEC on language use and problems.

0.2 The SIG should be involved in formulating language standards.

0.3 The SIG should provide information on the use of structured languages in a time critical commercial environment.

0.4 The SIG should investigate ways of taking advantage of operating system features, while maintaining operating system/implementation independence.

0.5 The SIG should develop and maintain a source code management system.

0.6 The SIG should work on developing a set of compilers for all DEC operating systems, and a unified interface to IBM and HP systems.

0.7 The SIG should provide information and surveys of Third Party Compilers.

0.8 The SIG should develop and maintain Cross Compilers for the various new 16 bit microcomputers.

0.9 The SIG should provide "C" compiler benchmarks.

0.10 The SIG should publish information in the newsletter, and otherwise address the issue, of "C" portability.

0.11 The SIG should develop language translators, such as Fortran to "C".

0.12 The SIG should provide detailed documentation on the input format to the Code Generator phase of the Decus "C" compiler, so that users could write their own code generators for other processors.

0.13 The SIG should develop a Fortran 77 that uses Pass 2 of the DECUS "C" compiler.

0.14 The SIG should support a Structured Fortran preprocessor package.

0.15 The SIG should investigate and develop support for Fortran 77 under RT11.

0.16 The SIG should maintain a Pascal for RSTS.

0.17 The SIG should investigate improvements to the MBS Pascal Compiler and Library.

0.18 The SIG should continue, and enhance, support for Pascal running under RT11 and RSX.

0.19 An RSX Praxis should be made available on the SIG tapes.

0.20 The SIG should make an ADA package available.

0.21 The SIG should address the issue of portable programming in general, especially in the area of standard or multilanguage libraries.

0.22 The SIG should publish articles in the newsletter dealing with information and experiences concerning Praxis and/or Ada like languages.

0.23 The SIG should provide an introductory newsletter, with information about Software Tools, and what is available on the SIG tapes.

0.24 The SIG should provide a comparison of available structured languages, in terms of their capabilities and available compilers.

0.25 The SIG should develop and maintain common debugging tools for all structured languages.

0.26 The SIG should encourage development of Software Tools, to be made available through the SIG tapes.

0.27 The SIG should survey to determine in what areas software tools are most needed.

0.28 The SIG should endeavor to provide faster turnaround for SIG tapes.

0.29 The SIG should provide better documentation of updates on the SIG tapes.

0.30 The SIG should investigate other means of acquiring updates.

0.31 The SIG should develop standardized libraries for various applications, such as file I/O, graphics, and terminal handling.

0.32 The SIG should maintain the largest possible range of languages for RT11.

QUESTIONNAIRE

What Structured Language(s) are you currently using ? ____________

Given a choice what language would you be using ? ____________

What would be your second choice? _____________

What operating system(s) are you using? RSX/IAS ( ), RSTS ( ), RT11 ( ), TOPS 10/20 ( ), VAX ( ), UNIX ( ), other ____________.

What is the primary purpose of your installation? Scientific ( ), Manufacturing ( ), Education ( ), Business ( ), Government ( ), Other ( ) what? _______________.

How large is your user population? _________

How many CPU's are in use at your installation? ________ VAX _____, PDP 11 _____, DEC 10/20 ______, PDP 8, _______, other ____ what? _________

Do you have a suggestion for a DEC directed wishlist item for the next wishlist?

Do you have a suggestion for a SIG directed wishlist item for the next wishlist?

ANSWER SHEET

DEC Directed Wishlist Items

0.1 _____

0.2 _____

0.3 _____

0.4 _____

0.5 _____

0.6 _____

0.7 _____

0.8 _____

0.9 _____

0.10 _____

0.11 _____

0.12 _____

0.13 _____

0.14 _____

0.15 _____

0.16 _____

0.17 _____

0.18 _____

0.19 _____

0.20 _____

0.21 _____

0.22 _____

0.23 _____

0.24 _____

0.25 _____

0.26 _____

0.27 _____

0.28 _____

0.29 _____ (write in)

SIG Directed Wishlist Items

0.1 _____

0.2 _____

0.3 _____

0.4 _____

0.5 _____

0.6 _____

0.7 _____

0.8 _____

0.9 _____

0.10 _____

0.11 _____

0.12 _____

0.13 _____

0.14 _____

0.15 _____

0.16 _____

0.17 _____

0.18 _____

0.19 _____

0.20 _____

0.21 _____

0.22 _____

0.23 _____

0.24 _____

0.25 _____

0.26 _____

0.27 _____

0.28 _____

0.29 _____

0.30 _____

0.31 _____

0.32 _____

0.33 _____ (write in)

DECUS

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

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

MOVING OR REPLACING A DELEGATE?

Please notify us immediately to guarantee continuing receipt of DECUS literature. Allow up to six weeks for change to take effect.

( ) Change of Address
( ) Delegate Replacement

DECUS Membership No.: ____________________

Name: ____________________

Company: ____________________

Address: ____________________

____

State/Country: ____________________

Zip/Postal Code: ____________________

Mail to: DECUS - ATT: Membership
One Iron Way, MR02-1/C11
Marlboro, Massachusetts 01752 USA

Affix mailing label here. If label is not available, print old address here. Include name of installation, company, university, etc.