DISCLAIMER

Software Tools COMMUNICATIONS

NUMBER 10 **MAY 1983**

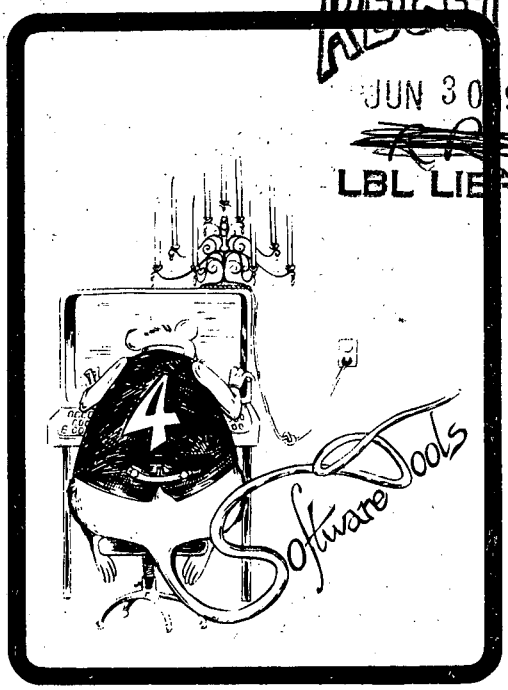
In this Issue

- Details on the next USENIX/Software Tools conference in Toronto.
- Appended to this issue of the newsletter you will find various STUG order forms. In addition to tape request and membership forms (keep STUG growing — give a membership form to a friend!), we are also including:
  - STUG SPR form — now there is a way to report those annoying bugs which keep cropping up everytime you get a new tape.
  - Software Submission Form — Please be sure to read the article on Tape Submission in this issue before making software contributions.

  We suggest that you make duplicates from the newsletter of forms you may need more than once.
- News of a new *STUG Hotline* phone number.
- A list of implementors of the Software Tools, along with their machines.
- News about:
  - Lex and YACC under the Software Tools
  - The SOFTFAIR in Washington, D.C. in July
  - The financial state of STUG
- Two technical papers:
  - "Poetic Programming", taken from the *Software Tools Notes* published in Australia.
  - "The RatsNest Project", describing a STUG standard network in progress.
- A new section: "Letters to the Editor"

As usual, we encourage all members to participate in STUG — give a short talk in Toronto (See "Call for Papers" in this issue), write about your current activities for the next newsletter, and come to the STUG meetings. We also encourage all non-members to join, then do the above...

-- the STUG Newsletter Editors



This work was supported in part by the United States Department of Energy under contract no. DE-AC03-76SF00098.

Behind the Scenes

Incorporation:

STUG is very close to incorporating under the laws of the State of California. Such incorporation will allow STUG to operate under the laws governing any corporation, providing protection to its members and allowing STUG to operate as a "Not-For-Profit" entity. We expect to be "STUG, Inc." by the time we meet in Toronto.

In order to incorporate, STUG is writing a set of STUG by-laws. These by-laws are being written by the STUG board of directors. If you would like to communicate with a member of the STUG board, just drop a line to STUG at the standard address. The current STUG board is made up of five people: Neil Groundwater, Dave Martin, Bill Meine, Dave Stoffel, and Bob Upshaw. The entire board will be in Toronto.

STUG Business:

STUG is about to release two "Requests for Proposals" (RFP's). These RFP's cover two separate office functions necessary for the operation of STUG: A financial officer and a distribution officer. The financial officer will be responsible for interfacing with the STUG treasurer and handling all monetary responsibilities for STUG. The distribution officer will manage the distribution of STUG related materials (tapes, proceedings, etc.), process membership requests, and so on.

If you know of any person(s) interested in responding to the STUG RFP's, please send a note to the STUG address, or leave a message with the STUG Hotline. (Both the address and phone number are contained in this issue.) We plan to mail the RFP's out for consideration by June 1, and select the contractor(s) by August 15.

Call for Papers

1983 Summer USENIX Conference

Presentations are invited for the Software Tools Users Group meeting in Toronto, Canada on July 12, 1983. The Software Tools Users Group meeting will again be held in conjunction with the Usenix meeting, July 12–15, 1983 at the Harbour Castle Hilton. If you have specific registration questions, contact Suzanne MacNary at (617)497–2964. All attendees of the Usenix Conference are invited to attend the STUG technical sessions on Tuesday evening. An implementors' meeting will be scheduled for Thursday evening.

Talks may include descriptions of projects using Ratfor and/or the Tools, newly-created or enhanced tools, software portability and programming environments, thoughts about future directions for the tools, or other areas of interest to the tools community. Suggestions for other types of presentations are also solicted. Abstracts should be submitted to the Software Tools program chairman:

Neil Groundwater
Analytic Disciplines, Inc.
8320 Old Courthouse Road, #300
Vienna, VA 22180
(703)893–6140

They must contain the following information:

Title
Name of author
Name of company
Mailing address
Phone number (and network address, if available)
Audio-visual requirements

The conference committee intends to produce a proceedings consisting of short papers (less than 10 pages) by the authors on the subject of their presentation, as well as all abstracts. Submission of a paper is not required, although it is strongly recommended. Papers will be collected in camera-ready form at the conference.

Financial State of STUG

Barbara Chase

Status Of Tape Requests

Thank you all for patiently waiting for your tapes. Now we are finally caught up with the tape orders, and are even waiting for new orders to arrive. We were even able to make 50 tapes to sell at the Unicom conference in San Diego. We are planning to sell

tapes at the conference in Toronto as well. Thanks to Ben Cranston, we now have a Univac tape we can read and will distribute in late May.

Various Statistics about the tapes:

- 87 tapes were ordered in 1982
  - 15 were delivered in 1982
  - 61 were delivered in 1983
  - 9 are Univac orderes not yet sent out
  - 1 cannot be shipped (no country was specified)
- 29 tapes were sold at the Winter 1983 Unicom conference
- 18 tapes have been ordered in 1983
  - 9 have been delivered as of April 30, 1983

Status Of Membership

As of the last newsletter we had 330 members. This newsletter was sent to both STUG members and previous newsletter recipients. Since that time we have only received an additional 216 members, 46 of which signed up at the Unicom conference. So at this time, we have a total of 546 members. Certainly there is much more interest in STUG than this; the previous newsletter recipients totaled nearly 1500. Where are the other 1000 who are interested in STUG ??

Industrial Members

In our last newletter we introduced two new forms of membership: the Industrial Member for $150 per year, and the Sustaining Member for $1500 per year (or a merchandise and support equivalent). Currently we have three Industrial Members:

Mark McWiggins — *IMSL, Inc.*, Texas

Micheal Liveright — *Systems Control Technology, Inc.*, Calif.

Kei Nakado — *Personal Media Corporation*, Japan

Sustaining Members

Special Thanks to:

Direct Inc.
— for their donation of a Direct 1000 computer

Carousel Microtools
— for their donation of CPM software tools

Lawrence Berkeley Laboratory and *Hughes Aircraft Company*
— for their continued support

Financial Report

The following report is an income statement reflecting STUG's financial status for the 1982 fiscal year. During the first part of the year, STUG distributed tapes and produced a newsletter. The later part of the year, Sept. — Dec., STUG began to set up its operational and financial organization, produced another newletter (thanks to LBL's support), and reorganized its tape distribution process.

Income Statement Dec. 31, 1982		
Sales:		
membership dues		5995.00
tape sales		4341.25
conference registration income		14265.94
		24602.19
Cost Of Goods Sold:		
membership:		
newsletter production (1 issue)	938.95	
newsletter postage (1 issue)	847.39	
database maintenance	1019.50	
	2805.84	
tape production:		
inventory Jan. 1 , 1982	0.00	
purchases (mag tapes & software)	890.38	
cost of production	1285.00	
	2175.38	
less inventory Dec. 31, 1982	-825.38	
	1350.00	
Total Cost of Goods	4155.84	
GROSS PROFIT		20,446.35
Expenses:		
Operating:		
travel	122.00	
phone	31.04	
mail (delivery & box)	148.00	
supplies	28.70	
miscellaneous	43.10	
salaries (clerical)	175.00	
	547.84	
Administative:		
accounting	75.00	
administrative	2137.50	
	2212.50	
Total Expenses	2760.34	
NET INCOME		17,686.01

Below is a income statement reflecting the approximate financial status of the Users Group as of March 31, 1983. (At this point, some of the financial information has not been determined.)

Income Statement March 31, 1983		
BANK BALANCE		23,537.52
Sales:		
membership dues		1055.00
tape sales		1563.00
t-shirt sales		410.00
software catalogue sales		230.00
conference proceeding sales		unknown
Cost of Goods Sold:		
membership:		
database maintenance	217.00	
	217.00	
tape production:		
inventory Jan. 1, 1983	825.38	
purchases (mag tapes & mailers)	1759.11	
cost of production	1082.69	
	3667.18	
less inventory March 31, 1983	-660.30	
	3006.88	
t-shirt production:		
inventory Jan. 1, 1983	0.00	
purchase	269.50	
less inventory March 31, 1983	-90.00	
	179.50	
software catalogue:	unknown	
conference proceedings:	unknown	
Expenses:		
Operating:		
travel	79.63	
mail (delivery & box)	75.00	
supplies	46.04	
miscellaneous	192.50	
implementor's meeting	39.98	
	433.15	
Administrative:		
accounting	705.00	
administrative	1745.00	
trademark	175.00	
	2625.00	

SOFTFAIR Set for D.C. in July

SoftFair--A Conference on Software Development Tools, Techniques and Alternatives, will be held on July 25–28, 1983 at the Hyatt Regency, Crystal City (Washington, D.C.). SoftFair will provide an opportunity for managers and technical staff to see and examine 'first-hand' modern tools and techniques for developing and engineering software. SoftFair invites researchers and developers to present, explain and demonstrate their tools and techniques.

In addition to several outstanding panel discussions, Softfair will include in-depth tutorials on software management techniques, development methodologies, and software engineering. Among the tools being demonstrated are program and application generators, syntax-directed editors, environments and workstations, rapid prototyping, management and design tools.

The conference is being sponsored by the IEEE Computer Society, the National Bureau of Standards, and SIGSOFT ACM. Additional information, including an advance program, are available from SoftFair, c/o IEEE Computer Society, P.O. Box 639, Silver Spring, MD 20901, (301) 589–8142.

STUG Hot-Line

Nancy Deerinck

In order to expedite questions and referrals for STUG members, we have designated a "Hot-Line" for phone queries. The number is located at the Real-Time Systems Group of Lawrence Berkeley Laboratory. If your call is not directly answered by the STUG representative, it will be routed to the RTSG Switchboard. If you connect with the switch-board, ask for the Software Tools Consultant and someone will be paged to answer your call. In the event that no-one is available, the switchboard operator will be happy to take your name and phone-number and relay it to the STUG representative.

The Hot-Line is available for any questions you have regarding STUG activities, conferences, tape availability, software bugs and fixes, newsletter submissions and contacting other implementors. If we can't help you, we can probably steer you to someone who can.

The Number: **(415) 486–4680**
The Times: 7a.m. to 6p.m. Pacific Time
Monday — Friday
(It may be possible to leave
a message at other times)

YACC and Lex Release Schedule

Theresa Breckon

We have been getting a steady stream of inquiries concerning the availability of Yacc and Lex. We hope to have Yacc ready to be distributed with the next STUG distribution tape. It will not be distributed as a standard tool, but as an extension to be evaluated and reviewed by Software Tools users. The plan is to have a new tools distribution tape this summer.

The Real Time Systems Group is in the process of converting all of our tools to conform to the standards that have been set by STUG. This includes converting and testing Yacc. We also have just distributed Yacc to a beta-test site for further testing. If all goes well, it will be included on the next distribution tape.

Lex is still in the development stage. It looks like it will be completed by September of this year. It will then have to be distributed to a beta-test site for further testing before we can submit it to STUG. We will be letting STUG members know of Yacc and Lex's availability through the STUG newsletter.

Call for Tape Submissions

Theresa Breckon
Nancy Deerinck
Van Jacobson

One of the main functions of the Software Tools Users Group is to serve as a focal point for collecting enhancements and extensions to the current Virtual Operating System. We are interested in receiving and screening useful additions to the basic tape and will provide a copy of the next revision of the VOS basic tape *free of charge* to anyone who we feel has made a significant contribution.

The basic tape contains two parts: A) the utilities and libraries that comprise the current, standard

virtual operating system, and B) new and extended utilities or libraries which are being distributed for evaluation and experimentation. The contents of the second part of the tape may be considered for inclusion in the first part of future tapes.

Compiling and editing new revisions of the basic tape is a massive task and can realistically be accomplished only if the software submissions conform to a consistent level of quality and accuracy. The following sections describe an acceptable software submission.

Acceptance Criteria

Your contributions must be written in the current, standard Tools Ratfor and be based on the standard Software Tools library and primitives. We will no longer accept submissions written in Pascal or C, nor can we use material subject to copyright or licensing restrictions. The software submission form in the back of this newsletter contains a release form which is necessary to make your work available to the Software Tools Users Groups members.

The submissions must be portable, i.e., implementable on any Virtual Operating System.

The submissions must be consistent in style with existing utilities, avoiding excessive 'featurism'. Extensions must be upwards compatible with the current version. Tool and routine names should not conflict with existing names.

Usage should be consistent with STUG standards, the current VOS and, where possible, with UNIX[1] usage.

What is Needed

The set of things comprising the VOS is divided into

- *Tools* (utilities) like ed, ratfor, sh, format, etc. All of the distributed tools are intended to be portable. I.e., they contain no machine dependent code and are written in terms of the VOS library.
- *Routines* like scopy, match, type, etc. Routines are the portable part of the VOS library.
- *Primitives* like open, putch, seek, etc. Primitives are the non-portable, machine dependent part of the VOS library.

Much of the success of the VOS rested on its ease of implementation which, in turn, rested on the carefully designed, *minimal* set of primitives. We would like to see the VOS continue to proliferate and want to encourage the community wide growth of the VOS. This desire gives us some definite "likes" and "dislikes" with regard to submissions:

New Tools, provided they are portable and written in terms of the current VOS, are always welcome and will be greeted with enthusiasm.

New implementations of the current VOS will likewise be welcomed. (These will be distributed by STUG to people interested in bringing up the VOS on the same type of machine. This distribution will be separate from and in addition to the "basic tape").

Significant additions to the library routines will be greeted with interest and, occasionally, enthusiasm. (yet another way to do a string copy will not be considered "significant"). Please think carefully about the portability and general utility of the proposed routines.

Significant enhancements to existing tools will be greeted with interest but some suspicion. Enhancements usually make tools larger and large tools are an anathema to small machines and small address space machines (e.g., CP/M, RT-11). Remember that much of the utility of the VOS derives from the Shell's ability to dynamically create enhanced tools out of existing tools via scripts and pipes. Perhaps your enhancement is really pointing the way to some missing, needed, new tool.

Significant additions to the primitives will be considered, but with a great deal of suspicion. Although there are definitely "holes" in the existing primitive set, it is *very* hard to fill those holes in a way that can be implemented across the wide variety of machines and operating systems that run the VOS. Please take the time to think through specifications for new primitives, emphasizing implementability, generality, usefulness, and consistency with existing primitives and UNIX. If you see a clearly valid need for new primitives, the primitives committee will review your suggestions and make a recommendation.

Changes to existing routines or primitives will be greeted with distaste and active dislike. Many hundreds of implementors have built and are building high level, useful tools on a VOS base that they fervently hope is stable (e.g., one doesn't even attempt to write a 5000 line "Yacc" if one thinks that the calling sequence to "getch" will be changing annually). Library changes do a tremendous dis-service to the entire community.

[1]UNIX is a trademark of Bell Laboratories

Submission Guidelines

The optimum format for us is one archive file for each new utility/package of routines. The archive should contain:

- A "man" entry for the utility plus tutorials or other relevant documentation.
- Symbol definitions and common blocks needed for the utility.
- Source code, including any library routines used that are not part of the standard distribution.

These archives generally have the format:

```
main archive
 tool.doc   (documentation)
 cblock1    (common blocks)
 cblock2
 ...
 tool.r     (source code sub-archive)
   rtn1     (each routine is a
   rtn2       sub-archive member)
   ...
```

Problems to Avoid

The following are the most severe problems we have seen in previous submissions. They all cost us time and result in the next tape taking longer to construct or including fewer of the submissions.

Improper data typing. Variables and functions must be correctly typed and there should be no mixing of data types in the same arrays. This is bad programming practice and causes much trouble when we attempt to port code to machines which store characters and integers differently. Much of the time and effort spent on the first tape was in correcting improper usage of data.

Use of machine-specific features or knowledge. Machine-specific system calls were avoided in the first submissions. More subtle use of interchangeable features, such as using mixing quoted strings with ascii character arrays, destroyed the portability of many submissions. Some were salvaged with a good deal of re-write effort, while others had to be discarded.

Missing library routines. Don't forget to include copies of the routines that you added to your local library.

Improper use of existing routines. If expanded capabilities of existing routines are required by your submission, it involves the review of the primitives committee, and requires a very valid need for the capabilities you want provided. In most cases the changes could more correctly be done by building another routine on top of the existing routine. If not, you should submit appropriately convincing documentation to substantiate your functional or syntactical changes.

Use of non-standard Ratfor. All submissions must be able to run through the standard Ratfor compiler, and must be devoid of non-standard Fortran features.

Please don't interpret the preceding sections as an attempt to scare away potential contributors. Software Tools users are developing their software in a unique environment, specifically designed so that the fruits of the development can be shared. Like most environments, this one's ecology is fragile and the developers have to care for it or it will be destroyed.

How and Where

Please send your tapes, accompanied by a tape submission form, to:

Software Tools Users Group
Attn.: Tape Submission
242 El Camino Real #1259
Menlo Park, CA 94025

ALL TAPES MUST BE ACCOMPANIED BY A TAPE SUBMISSION FORM. A form that you can duplicate is included at the back of this newsletter. If you wish to have the tape returned to you, please note that on the form.

Arrangements can also be made to submit software via the ARPANET. Submissions small enough to "mail" can also be sent via UUCP mail. Call the STUG hotline for more information on how to do this (assuming you have ARPANET or UUCP access to your machine).

If you need further information concerning submissions call the STUG hotline: (415) 486-4680.

Poetic Programming

Dr. Desmond FitzGerald
(reprinted courtesy of
The Software Tool Notes)

THE OXFORD DICTIONARY defines the term poetry as the expression of beautiful or elevated thought, imagination or feelings, in appropriate language. Poetry is usually succinct but adequate, sometimes hiding all its meanings until pushed by the intellect. I believe with programming that there should always be a motivation towards the poetic.

Programs and programming methods are generally not the products of any higher guiding philosophy than getting the job done in the *quickest* possible time. There is little reflection on what and how things were done yesterday with a view to improving productivity tommorrow. This limited approach to improving the working environment means that programmers meet constraints that curtail their ability to grapple with real and complex systems.

The concept of poetic programming can extend beyond isolated utility and/or application programs into the entire user environment --- forming a co-operating set of tools. In general, most environments are not of this ideal nature.

Human Beings and Their Potential for Programming

There are many failings or constraints that seem to crop up in programming that limit the amount and scope of work that can effectively be achieved. Among the constraints are

1. A maximum three to five hour concentrated burst of effort is the limit of most people's capability.

2. A distracted environment amounts to little real work being possible. A poor computing environment leads to frustration and distraction. Happy and harmonious surroundings are essential for productive work.

3. Lack of a *meaningful* goal causes a lack of motivation. Even with a worthy goal, lack of recognizable progress towards that goal causes difficulties. This general question of motivation is one of the more difficult areas to manage in a working environment.

4. Humans do not think in a logical sense at a very great speed. One good creative and intuitive idea may take a day or two of working and reworking at the logical level to form a workable framework.

5. The characteristic of the human mind of failing to comprehend increasing volumes of logic can become a major constraint. As an example of this, it is often acknowledged that the sheer volume of code in most operating systems makes it impossible for one person to at any one time be able to *keep on top of the code.* It is suggested that a limit of about five to seven competing factors is all that can be coped with at any one time.

6. Being attached to a way of working can be very counterproductive. Programmers often need to co-operate to achieve a goal and silly personal quirks do not help.

It can be seen that *HUMANS* have some very real problems and shortcomings when the task of programming is being considered.

Artificial Intelligence and Expert Systems

The case for establishing and maintaining high programming (coding) standards is made much stronger by looking to the future of programming. While not suggesting that programming as it is now known will become an obsolete profession, it seems likely that

1. Computers are going to take over more of the hum-drum lower level programming. Programmers will be using more abstract and concise methods.

2. *Expert Systems* will be developed that can accept input to a project from diverse sources and produce an integrated and consistent system. This process can be likened to a person reading a programming text and making very reasonable deductions about the code and the purpose of it without ever formally knowing the rules of the languages.

Good coding techniques *now* are an investment for the future. The quality of present thought ought to be preservable and translatable at quite an abstract level by future generations of expert system software. Poor software will be scrapped.

Making the Most of Personal Creativity

One or two people always give the creative insight and input into a well integrated system. The oft quoted "1% inspiration, 99% perspiration" applies very much to programming. While remaining in inspiration-mode is often an aspiration, we are mostly involved in hack-work.

Having painted this picture of where things stand, are there any elegant ways of coping with the 99% of

time in programming that we are doing fairly mechanical work in perspiration-mode?

Learning from Other Disciplines

It should be generally recognized that some of the time honoured engineering principles of design are equally applicable in programming (hence the term *Software* Engineering).

Some basic engineering design tenets are

1. Modularity of components.

2. Starting from a broad over-view design.

3. Adopt a policy of early prototyping and testing with subsequent refinements.

4. Coming to some agreed standards for the manufacture and use of commonly used components.

On this last point, we can make an analogy between programming computers and designing cars. The car industry by now uses standardized parts, optimized for form, function and robustness. Attention is paid to efficiency --- minimizing energy usage. Care is also put into the car's appearance. Cars have standardized user interfaces. The brake, clutch accelerator and steering wheel are positioned uniformly and respond in a standard manner. Very few of these attributes apply in most computing environments. The *Software Engineering* tenets are a guide towards better programming.

Speed of Thought

The consequence of each programming decision is a thought process which requires translation into code. The real benefits of good tools, programming techniques and library support come into their own to speed up this translation process.

If you can work somewhere near to the speed of your thoughts at a computer terminal, getting good turnaround on each module or group of modules that you write, then you can conceive of and debug a great deal of code in a very short time. This amounts to individual programmer productivity that is a hundred times the "norm". A three to five hour productive session can then achieve a major development. The "system" must not impinge and interfere with the thought process. It ought to remain in the background of consciousness. Working at or near the speed of thought keeps a project moving and motivation high. It encourages revision and refinement of ideas since changes will not take very long to implement. As an example, the Prime Source Level Debugger deserves high praise in being a tool that really helps in this area.

Being Smart vs Being Readable

There is often a tendency to hide in a very obscure one or two line algorithm the all important control flow or other essence of the code. This makes for difficult readability.

It has been found that being overly clever and/or obscure in programming is basically non-productive in the long term. Even the person who wrote the code can be quite mystified as to how it works after six months or more. There is always a simpler way of expressing the algorithm or data construct. If a mutual review of on-going work is followed, the benefits are great and there is an added stimulus towards improving existing code.

Some of the more common problems in this general catagory are

1. Avoid the confusing use of temporary variables.

2. Do not use "tricks" which are dependent on your private knowledge of a particular machine.

3. Consistent indenting around control flow statements, uniform spacing around operators and a common style of comments should be followed.

4. Keep internal documentation up to date. A well written program should be almost self-documenting (even without the use of comments). Additional documentation should not repeat what is obvious.

Development of Tools and Languages

Many promises have been made for new languages such as Ada and Pascal. The fact of experience is that Fortran will remain the bread and butter of technical computing. With Fortran 77, there is little need for non-standard features or extensions. The use of preprocessors (in particular Ratfor) is probably the the most significant trend to emerge in technical programming. A good case can be made to leave Fortran alone and to develop preprocessor techniques to aid in the more succinct and abstract statement of the logic required. There is a certain amount of dispute amongst the practitioners and the academics over whether Fortran 8 *X* should move more towards Ada or just leave out the obsolete features of old Fortran.

The *Software Tools* method with its associated libraries and utilities can make a very big impact on a user environment and personal productivity. Standardized library calls can be made available which give

the application programmer a very dense fabric of programming support. It is possible to liken this support to standard "sub-assemblies" of parts. With a few lines of initialization and the appropriate library calls, the programmer can call on some very powerful workhorses.

The RatsNest Project

Dave Martin

In the last newsletter, we mentioned briefly plans to establish a dialup network of machines running the Software Tools. The goal of this network is to facilitate communication between and exchange of software by STUG members. To this end we have tentatively selected a link-level protocol which seems appropriate for handling the expected electronic mail and file transfer operations. The protocol is called MMDF and is currently used by CSNETs "phonenet" network. The following information is extracted from:

MMDF Dial-up Link Protocol
by Edward S. Szurkowski
CSNET Design Note DN-4
April 1980

The MMDF protocol was designed "to provide a reasonable robust channel for the transport of 7 bit ASCII data between communicating processors over ordinary telephone connections." MMDF is oriented around sequenced, checksummed packets of the following types:

DATA	transmit data
XPATH	characterize transmit path
RPATH	characterize receive path
EXCAPE	send receive escape character
QUIT	cause remote protocol to exit

The above packets are acknowledged by separate acknowledgement packets of matching type; there is NO provision for explicitly NACKing a packet. Packets are NACKed by the lack of an acknowledgement within a "reasonable" amount of time. All packets contain only ASCII characters and are terminated by a carriage return character. Only half-duplex operation is assumed.

Protocol Operation

One of the more interested characteristics of MMDF is the initialization dialogue which takes place between the master (initiator of the connection) and slave. Through exchange of XPATH, RPATH and EXCAPE packets the two machines negotiate:

- The set of characters which may be transmitted without side effects.
- The set of characters which may be received without side effects.
- A suitable ESCAPE character for each end.
- A maximum physical size.

The negotiated maximum packet size (which may be between 40–255 characters) does not restrict "logical" packets in size; the transmitter breaks each logical packet up into as many physical packets as necessary.

Once the negotiations are complete, data is exchanged via DATA packets until the master sends a QUIT packet. The slave is then expected to send a QUITACK and exit.

Discussion

Now that we have a candidate protocol we must ask ourselves whether it is functionally acceptable and if it can be implemented portably. An MMDF host must be able to:

- Timeout on I/O operations
- Accept at least a 40-character packet at telephone (i.e. 120 cps) rates without dropping characters
- Disable terminal echo

The problems of echo suppression and I/O timeout are being addressed by (you guessed it) extensions to the virtual machine. Throttling of the data stream for sluggish hosts is not currently addressed by the MMDF specification; the timeout/retry mechanism is assumed to be adequate.

Although our initial implementation will be "textbook" MMDF, we would very much like to identify any shortcomings or lack of portability in the protocol before we attempt to make it a "STUG standard". Please send your comments on the MMDF protocol or other related networking issues to STUG, Attn. RatsNest Project. If you are unable to locate a copy of the CSNET design note, try asking STUG for a copy; you never know.

List of Implementors

Mark Aaker
NASA Ames Research Center
Mail Stop 241–3
Moffett Field, CA 94035
(415) 965–6410
+VAX 11/780

Eric Arman
The Brookings Institution
1775 Massachusetts Ave, N.W.
Washington, D.C. 20036
202–797–6185
+DEC-10

David A. Baumann
3340–18 Cheviot Drive
Fort Wayne, Indiana 46816
(219) 461–5760
+PDP-11,RSX11-M

Steve Bearman
Scripps Inst. of Oceanography
DSDP A-031
Univ. of California, San Diego
La Jolla, CA 92093
(619) 452–3526
+HP-1000

Nelson Beebe
Department of Physics
University of Utah
Salt Lake City, Ut 84112
(801)581–6901
+DEC TOPS-20

Frank Benjamin
c/o C.W. Holeman
1160 W. Chase Ave.
El Cajon, CA 92020
619–444–0446
+DEC RT-11

Michael Bourke
The Wollongong Group
1135A San Antonio
Palo Alto, CA 94303
Work: 415–962–9224
ARPA: bourke@sri-unix
+Perkin-Elmer, UNIX and OS/32

Frank Bradford
Foundation for Medical Care
952 South Mount Vernon
Colten, CA. 92324
714–825–6053
+PDP 11/34

Theresa Breckon
Real Time Systems Group
Bldg. 46A
Lawrence Berkeley Laboratory
#1 Cyclotron Road
Berkeley, Ca. 94720
+VAX, MODCOMP

Walter E. Brown
Director, Computer Center
Moravian College
Bethlehem, PA 18018
251–861–1300
+PDP 11

Robert Calland
U.S. Navy-Code 62
Naval Ocean Systems Center
San Diego, CA. 92152
619–225–2413
Calldand@ISIC
+VAX-VMS

John Campbell
732 7th Street
San Pedro, CA 90731
213–831–3938
+HP-1000 w/ RTE

Layne Cannon
Pacific Northwest Lab.
P.O. Box 999
Richland, WA 99352
509–375–2822
+VAX 11/780

Max Chandler
1030 S. Winchester Blvd.
Suite 205
San Jose, Ca. 95128
+Microcomputer 8080-Z80's

Michel Cornier
Department d'Informatique
Universite de Montreal
C.P. 6128, Succ "A"
Montreal, Quebec H3C 3J7
CANADA
514–343–7382
+CDC-Cyber 173

John Cowan
Kidder, Peabody and Co., Inc.
20 Exchange Place, 8th Floor
New York, New York 10005
(212)635–5262
+Tandem Non-Stop II

Ben Cranston
Systems Group
Computer Science Center
University of Maryland
College Park, MD 20742
301–454–2946
+Univac

Kent Crispin
Lawrence Livermore Lab
P.O. 808
Livermore, CA
415–422–4273
+CRAY-1

Norman C. Crowfoot
Computer Services
NAU Box 15100
Flagstaff, AZ 86011
602–523–2971
+HONEYWELL DPS8

Jerry J. Deroo
Faculty of Dentistry
University of Toronto
Biometrics Section
124 Edward Street
Toronto, Ontario CANADA M5G1G6
416–978–5396
+UNIX Version 7

Ben Domenico
N.C.A.R.
P.O. Box 3000
Boulder, CO 80307
303–494–5151 x559
+IBM VM/CMS

Walt Donovan
NASA/AMES Research Center
Moffett Field
Mountain View, CA 94035
Walt@bbnc or GAYDOS@BBNB
415–965–6368
+S.E.L. MPX1.4 and 2.0

William J. Donovan, Jr.
12815 S.W. 112 Terr.
Miami, FL. 33186
+IBM 370 ES

Larry Dwyer
Hewlett-Packard
11000 Wolfe Road
Bldg. 430
Cupertino, CA
408–257–7000 x 2095
+HP-1000 w/ RTE

Dr. Philip H. Enslow, Jr.
School of I.C.S.
Georgia Institute of Technology
Atlanta, Georgia 30322
(404) 894–3152
+PRIME 400 and larger CPUs
+PRIME 50-series

Randolph Franklin
Rensselaer Polytechnic Institute
Troy, New York 12181

(518)270–6330
+IBM 3033-MTS
+PRIME (Rev 18)

Joe Gallagher
Director, Scientific Computing
Cleveland Clinic Foundation
9500 Euclid Avenue
Cleveland, Ohio 44106
(216) 444–2551
PDP-15, VAX

Nancy Gow
Los Alamos National Laboratory
P.O. Box 1663
Los Alamos, NM 87545
(505) 667–4028
+VAX 11/780
+Cyber 825
+BSD UNIX 11/70
+7600
+CRAY-1

Neil Groundwater
Analytic Disciplines
8230 Old Courthouse Road
Suite 300
Vienna, VA 22180
703–893–6140
NPG@SDAC-UNIX
+VAX-VMS
+UNIX-Ver.6

Mel Haas
Bell Laboratories
Room HO 2G-431
Homdel, NJ 07733
(201) 949–1562
+UNIX 4 and 5
+IBM MVS/TS0–3081
+IBM Amdahl Ver.8

Dennis Hall
Computer Science and Math Dept.
Lawrence Berkeley Laboratory
Berkeley, CA. 94730
415–486–6053
Hall@LBL-Unix
+VAX

Edward Hall
MIT Lincoln Laboratory
2A Iris Court
Acton, MA 01720
617–863–5500 x3469
nrp5@LL
+Amdahl 470 CMS (370 compatable)
+Data General Eclipse

Richard M. Hambly
Harris Corp
1680 University Ave.
Rochester, New. York 14610
716–244–5830
+PDP11–70/RSX11M plus

John Hanshew
CompuCode
6147 Aspinwatt Road
Oakland, CA 94611
415–339–9463
+Data General RDOS

David R. Hanson
Assistant Professor
Department of Computer Science
University of Arizona
Tucson, AZ 85721
602–626–3617
+DEC-10 and CDC Cyber

Steve Hathaway
P.O. Box 500
Beaverton, OR 97077
503–685–3292
+DEC TOPS-20

C. W. Holeman
1160 W. Chase Ave.
El Cajon, CA 92020
619–444–0446
+DEC RT-11

Jung P. Hong
Los Alamos National Lab.
Mail Stop D455
P.O. Box 1663
Los Alamos, New Mexico 87545
(505) 667–8495
+UNIX

Ray Houghton
National Bur. of Standards
Technology Bldg., Rm A255
Washington, D.C. 20234
(301)921–3545
+VAX-VMS
+PDP-11/780
+Onyx 8002

Jessie C. Howell
6008 Clames Drive
Alexandria, VA 22310
Work: 703–525–6020
+Tandem & DEC IAS

Paul Howson
197 Alma Road
East St. Kilda 3182
Victoria, Austrailia
(03) 527–5881
+PRIME
+PERKIN-ELMER
+HONEYWELL

Van Jacobson
Real Time Systems Group-46A
Lawrence Berkeley Laboratory
#1 Cyclotron Road
Berkeley, Ca 94720
+VAX, MODCOMP

Rob Janes
Mailcode 50181
Cummings Engine Company, Inc.
Columbus, IN 47201
812–372–7211
+MODCOMP IV
+Honeywell DPS8–70

Chris Johnson
Software Resarch Associates
P.O. Box 2432
San Francisco, CA 94126
415–957–1441
+Onyx C8002

Richard Kiessig
Intelligent Decisions, Inc.
440 Cesano Apt. 105
Palo Alto, CA 94306
415–949–2306 (home)
+VAX
+Motorola 6800

Marty Kittower
The Software Toolworks
14478 Gloretta Drive
Sherman Oaks, CA 91423
213–986–4885
+Heath, +Kaypro 2
+Xerox 820
+Osbourne 1 and 2

Bill Lee
University of Texas
Computation Center
Austin, TX 78712
512–471–3242
lee@utexas
+Cyber 177–50

Bob Lewis
CGIS
4231 Norwalk Drive
No. EE312
San Jose, CA. 95129
(415)966–8440 x334
+Apollo

John W. Lewis
General Electric Company
Corporate Research and Dev.
1 River Rd, B-37,Rm 561
Schenectady, New York 12301
518–385–1600
+VAX

Dave Martin
Hughes Aircraft Company
MS C320
Bldg.R1
P.O. Box 92426
Los Angeles, CA. 90009
213–648–9927 after 11:30

+DEC VAX/VMS

Bill Meine
Louisiana Land
3900 S.Wadsworth Blvd.
Suite 660
Lakewood, CO 80235
303–988–8660
+IBM OS/MVS

Webb Miller
Dept. of Computer Science
University of Arizona
Tuscon, Arizona 85721
(602)626–3685
+VAX/UNiX,
+PDP11–70 UNIX

Michael N. Norred
MINESoft
13271 W. 20th Ave.
Golden, CO 80401
(303) 238–8911
+VAX 11/780 and 11/750
+Harris 800

Bill Patton
Armour-Dial, Inc.
P.O. 1427
Fort Madison, Iowa 52627
(319) 463–7111 x330
+General Automation 440, 480

Robert D.Perry, Jr.
Tektronix
Information Display Division
P.O. Box 1000 MS63–296
Wilsonville, OR. 97070
(503)685–3567
+DEC-20

Chris Peterson
M/A-Com Linkabit, Inc.
10453 Roselle St.
San Diego, Ca. 92121
619–453–7007 x454
+DEC TOPS-20
Chrisp@lbl-unix]

Doug Porter
Porter, Carlin and Associates
6030 Unity Drive
Suite M
Norcross, GA 30072
404–447–1341
+Perkin Elmer

Ken Poulton
Terminal Software
3182 Greer Road
Palo Alto, CA 94303
415–856–8659
+HP-3000 w/ MPE IV (or III)

Richard C. Raffenetti
Argonne National Laboratory
9700 South Cass Ave.
Argonne, Ill. 60439
(312) 972–8497
+VAX 11–70, 11/780

Peter Reintjes
Data General Corp.
62 Alexander Drive
Research Triangle Park, NC 27709
919–549–8421
Usenet: dukeuncpbr
+Data General AOS & AOS/VS

John Saxer
Logicon, Inc.
3535 Lomita Road
Bldg. 127 Room 1438
Torrance, CA 90505
213–517–7302
ARPA: saxer@NPRDC
+Burroughs B6900 & B7800

Debbie Scherrer
C.S.A.M.
Lawrence Berkeley Laboratory
#1 Cyclotron Road
Berkeley, CA. 94720
Scherrer@LBL-UNIX
home(415)881–4489

Dr. Phillip Scherrer
Carousel Microtools
30261 Palomares Road
Castro Valley, CA. 94546
415–881–4490
+Digital Research CP/M

Mike Shapiro
NCR Corportation
16550 W. Bernardo Drive
San Diego, Ca. 92127
+NCR V8000 w/ VRX
+CDC Cyber w/ NOS

Sid Shapiro
Wang Institute
Tyng Road
Tyngs Boro, MA 01879
617–649–9731
+Wang VS-80

Dr. Jerome Silbert
Laboratory Service
Veterans Adm. Medical Center
West Haven, CN 06516
203–932–5711 x466
+Data General Eclipse

Dr. James A. Stark
485 — 34th Street
Oakland, CA. 94609
415–658–2566
+CompuPro

David Stoffel
11872 Dunlop Ct.
Reston, VA 22091
(703) 620–4143
+Univac

Joe Sventek
Bldg. 50B
Lawrence Berkeley Laboratory
#1 Cyclotron Road
Berkeley, CA. 94720
415–486–5205
Sventek@LBL-UNIX
+DEC RSX-11M

J. Otto Tennant
Cray Research
1440 Northland Drive
Mendota Heights, MN 55120
+CRAY-1 COS

Bob Upshaw
Real Time Systems Group-46A
Lawrence Berkeley Laboratory
#1 Cyclotron Road
Berkeley, CA 94720
415–486–6411
+MODCOMP w/ MAX-IV
+DEC VAX/VMS

Letters to the Editor

This section of the newsletter is devoted to publishing short letters intended for general dissemination among STUG members. Letters sent to STUG should include a request for publication in order to be considered for the newsletter. Long letters may be abridged by the newsletter editors to fit the available space.

[This letter concerns] data-base software for the virtual machine. I have been using software tools to develop applications programs which require a large and complex data-base. I have developed extensions to the Virtual Machine Library that are sufficient for the design of a multiuser network data-base along CODASYL lines. I would be most interested in communication with anyone who has also worked in this area.

The Virtual Machine Library extensions deal with random access, concurrent processing and a "wait" primitive. Since [STUG members] are among the rare group of people who have actively worked and thought about portability, I would much appreciate [their] point of view, particularly on the technical feasibility.

Jerry Silbert, M.D.
Chief, Clinical Pathology
Veterans Administration
Medical Center
West Haven, CT. 06516

It may interest you to know that C.R.A. (Australia's largest mining company) has recently updated its technical computing policy and the preferred technical programming languages are (in this order):

1. Ratfor
2. Fortran 77
3. Basic

This will no doubt spawn many new users here who will look to the Software Tools system for guidance and standards. As participants in this C.R.A. effort, being responsible for the introduction of the Software Tools system to a number of sites, we are very conscious of the need for consistency, reliability, and portability. These are the very issues that attract engineering users to Software Tools.

The line of communication between STUG and non-U.S.A. users is fragile (due to distance). We don't as yet have the sort of networking that you enjoy in the U.S.A. It is therefore important that there is stability in the Software Tools system.

It is likely that there will be continued pressure to make changes to the Ratfor language. However, we would like to see an agreed standard which will remain stable for some time. The Software Tools distribution tape should provide sources in a form where code utilizing any local Ratfor extensions has been compiled into Ratfor code compatible with this agreed standard.

Having recently re-read the Unix article by Kernighan and Mashey, I am reminded of their statement:

> ...its creators have always favoured taste, restraint, and minimality of construct. There is a steady pressure to reduce the number of system calls, subroutines, and commands by judicious generalization or combination of similar constructs. In some environments, every new construct is hailed as an advance, following the philosophy that more is always better. Unix developers tend to view additional constructs with suspicion, while greeting with pleasure proof that several existing constructs can be combined and simplified, presumably because some insight has been achieved. ...It is especially important to maintain the simplicity of constructs that are truly central to everyone's use.

We feel, in observing some of the new Software Tools work, that perhaps these principles are being lost sight of and too much obscure complexity is creeping in. There seems to be a steady increase of primitives, system-wide definitions and options on tools. In light of this, perhaps there should be a cut-down version of the Software Tools system for newcomers that will introduce them to the ideas gradually while still providing them with something that works reasonably quickly.

This whole effort requires us to tread a fine line, always evaluating what we're doing and how we're doing it. We must always ask ourselves: "Is this extra (complicating) facility really necessary?" If not, then leave it out. I shudder to think that Software Tools users will come to the stage of having to consult the manuals for almost everything they type (as do some users of VMS and other systems that we know).

We feel in somewhat of a dilemma. On the one hand, we would like to adhere to a widely used

standard system. We do not have the resources to maintain and service all the Software Tools ourselves independently of STUG. Yet, on the other hand, if we adopt the STUG distributions, we must be assured of real effort at standardization and portability on your part. If, through constant uncontrolled tinkering, the STUG distribution becomes obscure and over-complicated then we will be forced eventually to scrap it. Please allow us to minimize the buffering and vetting we have to do between STUG and the users with whom we come in contact.

Having said all that, we remain enthusiastic about the new developments you have all so diligently worked at — viz. the TCS package, the mail system, the YACC tool and the graphing tools. We look forward to receiving the new distribution tape.

Paul Howson
Des FitzGerald
Consulting Engineers
197 Alma Road
E. ST. Kilda, 3182
Australia

I have been successfully using the book "Software Tools in PASCAL" in a programming course for students with a previous knowledge of PASCAL. I installed programs from the book in an UPDATE library on a CDC CYBER System under NOS/BE. The students had to get those programs, modify them and run them; the main idea was to make them aware that one need not always start from scratch to implement a big program, and to have them read "good" programs so that by osmosis they will be better programmers. I asked them to implement the FORMAT program to be able to deal with the strange character set of the CDC CYBER and then to be able to deal with a local extension to be used in French texts. All in all, this experience was a success and the students learned quite a lot.

Guy Lapalme
Dept. of Information
and Operational Research
C.P. 6128 Succ. A. Montreal
P.Q. H3C 3J7 CANADA

software tools users group

1259 el camino real #242, menlo park, ca 94025

Application for Membership

Date: _____

Name: _____

Address: _____

_____

State/Country/Zip: _____

Phone: _____

Network Address: _____

Machines and systems on which you use the Software Tools package:

_____

Utilities/library functions you have implemented
_____ The standard package (as distributed by STUG)
_____ The original package (Kernighan and Plauger)
_____ Other: _____

Other systems on which you plan to implement the Tools package:

_____

Special Interests: _____

_____

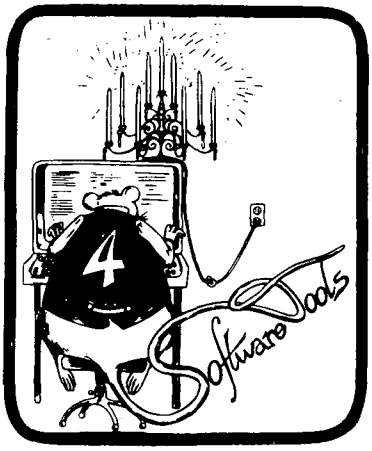
Category of Membership:

_____ Individual Membership $ 15.00
_____ Industrial Membership $ 150.00
_____ Sustaining Membership $1500.00 or more

_____ Overseas Air Mail $ 5.00
_____ Privacy; do NOT want this info. available to STUG members

Amount Enclosed: _____

Make checks payable to Software Tools Users Group, and mail to above address.



software tools users group

1259 el camino real #242, menlo park, ca 94025

Tape Order Form

Date: _____

Name: _____

Address: _____

_____

State/Country/Zip: _____

Phone: _____

Network Address: _____

Target computer(s) for the Tools: _____

Please send the following tapes:

_____ Portable LF Terminated, 2048 cpb ASCII
_____ 800 BPI _____ 1600 BPI

_____ Portable — Blocked Card Image, 3200 cpb ASCII
_____ 800 BPI _____ 1600 BPI

_____ VAX/ Unix 4.1 BSD Tools Tar Format
_____ 800 BPI _____ 1600 BPI

_____ VAX/ VMS Tools (LBL-Hughes) Files-11 Format
_____ 800 BPI _____ 1600 BPI

_____ RSX-11M Tools (LBL) BRU Format
_____ 800 BPI _____ 1600 BPI

_____ UNIVAC 1100 Tools @COPY,G Format

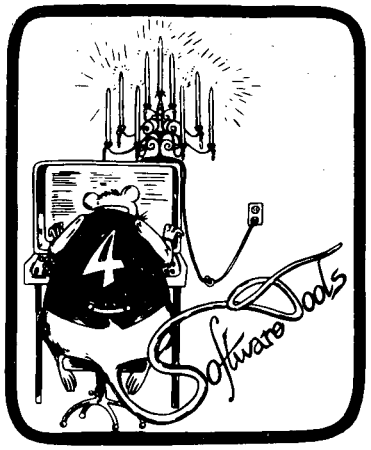
No. of tapes order: Subtotal:

_____ @ $50.00 /tape _____

Overseas Air Mail: @ $10.00 /tape _____

Total Amount Enclosed: _____

Make checks payable to Software Tools Users Group, and mail to above address.



software tools users group

1259 el camino real #242, menlo park, ca 94025

Software Submission Form

Please read the article on Tape Submission included in this newsletter. Your submission should be in archive format and include manual entries, routines, etc. as described in the article.

Machine and System on which you made the tape:

Brief description of tape contents:

Density: _____800 bpi _____1600 bp
(9-track only)

Character Code: _____ EBCDIC _____ ASCII

Blocking Factor:

Software Release
****

I (We) the undersigned give the Software Tools Users Group permission to reproduce and distribute all or any part of the program package material contained on the above tape for the use of STUG members. This material is not subject to copyright.

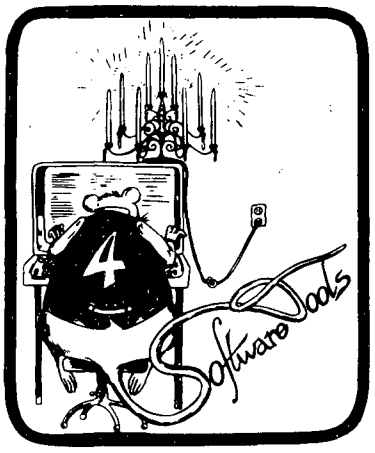
Submitted by: _____

_____

_____

SIGNATURE(s) _____ DATE _____

_____ _____



software tools users group

1259 el camino real #242, menlo park, ca 94025

Software Problem Report

Date:_____

Originator's Name:_____ Address:_____

Phone:_____ _____

Net Address:_____ _____

_____ _____

Name of Tool(s):_____

Machine/Operating System:_____

Date of Distribution Tape:_____

****

Problem Description: (Check all that apply) ____source ____routines ____manual entry

****

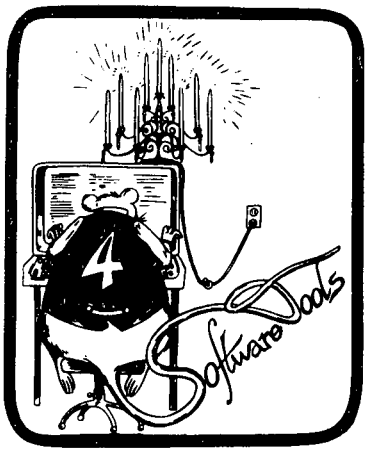
****

Suggested fix, if any:

****

If you have questions, call the
STUG Hotline: (415) 486-4680

****



software tools users group

1259 el camino real #242, menlo park, ca 94025

T-Shirt Order Form

Date: _____

Name: _____

Address: _____

_____

State/Country/Zip: _____

Phone: _____

Please send me the folowing t-shirts:

Size	Color (Blue, Brown)	Number
Small		
Medium		
Large		
X-Large		

No. of t-shirts ordered: _____ Subtotal:

_____ @ $10.00 /t-shirt _____

Total Amount Enclosed: _____

Make checks payable to Software Tools Users Group, and mail to above address.

Lawrence Berkeley Laboratory
RTSG – 46A
University of California
Berkeley, CA 94720

PUB-402/5-83/700