



# News Letter

January 1989

Copyright 1989 USUS, Inc.

All the News that Fits, We Print

William D. Smith, Editor

Volume 3

Number 1

## SAGE/STRIDE News

The following is a short news release on the new Millennium company. Needless to say, with less than a month of existence and only a few people, Millennium is swamped with work to accomplish. We will be able to provide more information at a later date as the business arrangements are solidified.

Through an associate company, Panasatek Manufacturing Ltd., Millennium Computer Corporation has gained the ability to supply Stride computers, boards, and accessories from the inventory that was on hand at the time of the MicroSage entry into bankruptcy.

Millennium has been organized by former employees of MicroSage and is effectively a start-up company, but one with a lot of experienced people and many friends in the Power System and UNIX worlds. Among the initial employees of Millennium are Bill Bonham, Wilber Harvey, Lonnie Cline, and Bob Needham. Ed Chapin, former president of MicroSage is the new president of Millennium. Rhine Meyering is handling Marketing and Sales, while Dave Young is responsible for Operations.

Millennium currently has under development a new computer which will be called the 740 System. The 740 System will initially be offered with a 25 MHz 68030 with expandability to use future Motorola processor technology. The new computer is a Millennium development, but will provide compatibility with other systems such as the Stride 400 and 600 Series, NCR's Tower family, and the Unisys 5000 Series. The 740 System will support both the Power System and UNIX operating systems.

Millennium is located at 125 Catron Drive in Reno, Nevada 89512. Phone: (702) 786-6464; FAX: (702) 786-6525.

## Orphan Software Project (Nov. 29, 1988) by Beverley Henderson

This will be the first formal report on the responses to the Software Information Request Form. May their tribe increase!

First of all, I need to ask all of you to check the address you have for me. I seem, somewhere, somehow, to have gotten my office ZIP code attached to my post office box. This has caused many of the replies to arrive late with "Delayed Due To Wrong ZIP Code" messages. The proper zip code for P. O. Box 1389, El Granada, CA is **94018**. Please accept my apologies.

I am delighted with the diversity of the replies so far. Not only are there many different programs, but a wide range of geographical origin is also apparent. USUS may not be large, but it is widespread. Thank you all for taking the time to help with this project.

In the interests of remaining germane to my title, I will deal separately with the questions. I will send them along to our Editor for inclusion in the letters or questions section so that anyone with answers can respond.

Two of the items already in hand are large -- both in size and potential resources. The first, from **Hays Busch**, is a copy of the July, 1984, USUS News and Report (No.12) that contained the 15 page Vendor Directory by **Eli Willner**. Since I still have my copy that arrived because I was a member, I will be glad to pass this on if there is a need somewhere. However, the information is now on disk, ready to be appended and updated, so this hard copy may be redundant. But if you have a use, please ask.

The second large contribution comes from **Verlene Bonham** (thank you, also) who ~~works~~ *use to work* for MicroSage in Reno. This is the July, 1983, Fourth Edition of the "p-System Applications Catalog" published by SofTech Microsystems of San Diego. My next project is

to read these and/or the current disk print out against each other so that we can add to our list.

I also received an offer of a "p-System Application Catalog" from **Karl Schuneman** in MI. Karl, is yours the same as the one from Verlene? Even if it is a different edition it might contain additional information. Would you check your copy and let me know? Thanks.

We will be adding programs and vendors from the forms we get back. Already I have been able to look up several programs where details were missing from memories (i.e. Hays' Timberline Spreadsheet is in the catalog from Verlene). See, it works!

Another prompt and enthusiastic responder was **David Craig** of Wichita, KS. He seems to own the only LISA in USUS -- or at least the only one we know about so far. How strange that an organization devoted to the p-System has only one of the machines built around that system. At any rate, if there are any other LISA's out there, he would love to know. David Craig, 736 Edgewater, Wichita, KS 67230. Phone: (316) 733-0914. I am very grateful for all his information as I am myself interested in LISA.

From **David Oshiro** of Nichandros Companies in Oakland, CA, I received a list of programs the company is currently running on their Stride 460 under the p-System. These include accounting, word processing, spreadsheet and data base management programs. All in all, a paragon of contemporary, functioning, real word UCSD uses. Among all the "memories", this breath of current vitality was very welcome. Thank you, David.

David also included detailed descriptions of the programs along with some copies of pages from the manuals. If there is interest or need, I will be glad to include more about these in some future article. Or his address is: David Oshiro, Nichandros Companies, 2901 Glascock St., Oakland, CA 94601. Phone: (415) 533-6708.

The most intriguing reply so far comes from **Floyd E. Lauersen** of Spokane, WA. He sent information about two programs, one called Nutri-Calc from PCD Systems and one called QRS (Quick Response System) prepared by Comsis Corporation under contract to the Federal

Highway Administration and supported by the Center for Microcomputers in Transportation at the University of Florida. BUT, he didn't say what he uses it for. I am mighty curious to know if he keeps a VERY large model train system or... Thank you for the fascinating possibilities of this one, Floyd.

From the Mkuze Game Reserve in South Africa, **Peter Goodman** writes in a letter to Frank Lawyer that he has five p-System programs: OMNIS from Blyth Computers in England, Project Planner from Applitec Software in MA, PowerText from Beaman Porter in NY, Statpro from Wadsworth Electronic Publishers in MA and Micro-Dynamo from Pugh-Roberts. He doesn't give an address for them, but in the catalog from Verlene they are also in MA. Thank you for your time, Peter.

These five programs from one person illustrate how much help other replies are in filling in and fleshing out our information. Only Applitec is not in one of the two catalogues I already have, and that program was purchased in 1987 so is far too new for our materials. A good example of why we need help updating our information.

From **Robert Allen** in FL, I have an interesting program, PDBase. This is from an independent company, evidently small as Robert says you can get help from the author, called IOTC in WY. It gives me great pleasure to see some successful competition for the giants. Thank you for your reply, Robert, and I will submit your questions to the letters section.

**Richard S. Murdock** of Worcester, MA, sent Calc7 from Micro Research in CO. He says the program is currently well maintained, with access to the company on CompuServe as well as the usual mail and telephone.

And from the Sunbelt, I received information about LePlot from Grifco in Mesa, AZ. This is the real scoop -- **Bob Griffith**, who sent in the form, is the program author and supports it on CompuServe. This is a "dot matrix plotting program" which seems to mean you could use it to plot without a plotter. Do I have that right, Bob?

Another program that arrived in response to our form does not run under the p-System but does

implement ASE under MS-DOS. The program is called JOE (Jon's Own Editor) and if you are interested in its capabilities, his address is **Jon Bondy**, JLB Enterprises, Box 148, 2926 Mapleshade Rd., Ardmore, PA 19003. Phone: (215) 642-1057 or CIS: 21545,2023.

There are more replies, but I will have to save some of them for next time or I'll have the entire Newsletter devoted to this one topic. Please send in the form if you have any programs under the p-System. This looks as if it could become really useful, as well as serve to put people in touch with sources and information they need.

Thank you again for all your time and **change that zip code!**

#### **Administrator Says** by Hays Busch

Well, I "blew it" for the December issue. My only excuse is that it was budget time for the Genesee Foundation (the homeowners association for the area in which I live). And as Treasurer, I had a lot of work to do for them. By the way, I'm also a member of the Genesee Volunteer Fire Department. Any other "Fire-fighters" among USUS Members?

On his renewal form, Richard S. Murdoch checked "Professional" and added this note,

"To keep the record straight - I am not a "Professional", no work, consulting, etc., status. Just wish to support USUS (am retired)."

By now he's received my letter of thanks, and I thought I would share his enthusiasm for USUS with the rest of you. It's his second year as a Professional Member and that says volumes.

Speaking of members, I will have a full report of our membership make-up in the next issue of the Newsletter. I can tell you now it will be down in total from the numbers we had this past year. But all results are not in at this time, so hold on for a bit. It is pretty obvious USUS is going to have to have some well organized new-member programs next year. In fact, it needs to become our number one priority. So if you have any suggestions, let us have your ideas.

If the Board of Directors approves, renewal notices for next year will go out quarterly in February, May, August and November. Members who do not renew by the end of the quarter (March, June, September, December) will be placed on the "inactive" list at the end of each quarter. This new process will keep Membership lists current quarterly. We just can't continue to expend funds for printing and postage on members who are not supporting USUS. It's not fair to those who do.

We're hoping to make some other changes in the administrative activity that can reduce postage and printing costs and still keep all of you up to date. More about this as it falls into place.

Now, I know that all of you just "can't wait" to read this column and the Board of Director minutes in your NewsLetters! But like William Smith, the Editor, said in the last issue, he can use some help to make the Newsletter more interesting. So don't be shy. Send him programs, questions, comments etc. Many of you are "old timers" or use UCSD Pascal in interesting ways. Why not send William a short write-up on what you are doing. Folks I can think of who would be interesting to hear from include **P. D. Terry, Juerg Scheidegger, Richard McMahon, Mike Ikezawa, Windsor Brown...** to name just a few. This is your Newsletter and the more you contribute, the better it can be. Just send the stuff to William Smith at the La Jolla Post office box. On disk is best, but hardcopy is fine. We can read MAC, SAGE/STRIDE, IBM and APPLE. *(Or send it via EasyPlex on CompuServe. This is best for me. William)*

One more thing. USUS needs more volunteers. We need a person who can take charge of Membership, a person to take charge of New-Member development, a person to become the IBM SIG chair, a person to be Publicity chair and get our name back in print, a Secretary, etc. USUS is your organization. If you don't help, a lot of important things don't get done. And then the value of "member service" falls off. So how about it?

**Treasurer's Report (Nov 1988)**  
by Robert E. Clark, Treasurer

73007,173 William Smith, Assistant SysOp,  
NewsLetter Editor

Income - November 1988

Dues:		(new/renew)
Individual	850.00	3/25
Professional	200.00	0/2
Institutional	0.00	0/0
Other Income:		
Library fees	7.00	
PowerTools	0.00	
CIS	0.00	
Misc.	44.50	
<b>Total Income:</b>	<b>1,101.50</b>	

Expenses - November 1988

Administrator:		
CIS	33.00	
Telephone	16.25	
Photocopies	4.60	
Postage	40.80	
Other:		
Box reimbursement	36.00	
Mail from La Jolla	7.75	
Bank charge	1.00	
<b>Total Expenses</b>	<b>\$139.40</b>	
Bank Balance	\$6,443.25	11-30-88

**Board of Directors Minutes (Dec. 6,1988)**  
by Samuel B. Bassett

MINUTES OF THE SPECIAL MEETING OF THE BOARD OF DIRECTORS OF USUS, INC., HELD IN ROOM 1 OF THE MUSUS FORUM TELECONFERENCING FACILITY ON THE COMPU-SERVE INFORMATION SERVICE, BEGINNING AT 10:09 PM EST DECEMBER 6, 1988.

Present at the meeting were:

User ID	Name
76314,1364	Sam'l Bassett, Board Chaircritter
75226,3643	Bob Spitzer, Board Member
76703,500	Eli Willner, Board Member
72135,1667	Harry Baya, outgoing Board Member
72401,1417	Frank Lawyer, newly-elected Board Member
70260,306	A. Hays Busch, Administrator
72747,3126	Robert Clark, Treasurer

Matters dealt with were:

Bob Spitzer's Board Term

At the November 22nd meeting, the Board determined that Eli Willner and Sam'l Bassett would have 1-year terms, Frank Lawyer and Henry Baumgarten would have 2-year terms, but left the matter of the length of Bob Spitzer's term up in the air. Bob said he didn't care overly much, and figured that at the end of two years, he'd be totally burned out, anyway. William Smith proposed that the Board drop the matter in Bob's lap (i.e. let him choose the length of his term), and when pressed, he opted for 2 years. { Chaircritter note: This means that in the 1989 election, there will be only two Directors up for re-election. }

Beginning & Ending of Directors' Terms of Office

William Smith moved that the By-Laws be amended to make Directors' terms of office begin on January 1st and end on December 31st. After a great deal of recursive discussion, no second to the motion was heard. Harry Baya moved that the Board seat the newly-elected Directors immediately, and their terms end on Dec. 31 of 1989 and 1990, as appropriate. William withdrew his motion in favor of Harry's, and the motion passed unanimously.

Seating New Members & Electing a Chair

The new members (*Frank Lawyer and Henry Baumgarten*) were seated forthwith (Frank Lawyer was the only one present) and the old members (*Harry Baya and Jon Nevins*) were thanked for their efforts on behalf of USUS over the past several years (Harry Baya was the only one present). Eli Willner nominated Sam'l Bassett for the post of Chairman of the Board, and he was duly so elected by unanimous vote (except for his own abstention). The re-elected Chaircritter thanked the Board for their "kindness" in placing him once more in the hot seat.

The Matter of the Presidency

While USUS nominally still has a President -- Weber Baker -- he has not been active in several

months, is not presently a USUS Member, and has not been able to be contacted by anyone, despite repeated attempts by several people. After much discussion of the matter, Bob Spitzer moved that:

- 1) The Board send a letter to Weber, stating that, as of the next Board Meeting (January 10th, 1989), if nothing has been heard from him about resuming his duties, the Board would consider that he had resigned;
- 2) That in the interim the Board consider possible nominees for the position;
- 3) That the Board consider establishing policy and procedures for handling similar situations in the future.

The motion was approved unanimously.

In regard to 1), the Chaircritter undertook to write the letter (the Administrator having firmly declined). In regard to 2), Bob Clark mentioned that Alex Kleider would be willing to assume the position if asked, but is not actively seeking it.

#### Other Officers

The Chaircritter pointed out that USUS has been without a Secretary for well over a year, and that we really need someone to keep track of the paperwork, make a formal paper Book of Minutes of Board Minutes, look through the minutes for policy statements, etc. Bob Clark mentioned Howard Sweet, a USUS member in the Washington, DC, area, who has offered to help with publishing the NewsLetter/Journal. Eli Willner cautioned that the Board (or delegates) should meet with prospective officers before we appoint them to office. The Chaircritter then suggested that Frank Lawyer and Harry Baya, when they are next in the D.C. area for a Washington Apple Pi meeting, try to meet with Howard, sound him out about the job, and get a personal impression of him as a person to work with. This suggestion met with general approval, and the two volunteered to take the matter in hand.

#### The Nominating Process

William Smith proposed that the following become USUS Policy for handling elections:

- 1) That the Board appoint a Nominating Committee in June of each year.
- 2) That the Nominating Committee find and qualify nominees, get their election campaign statements, and deliver the statements to the NewsLetter editor by August 15th.
- 3) That the issue of the NewsLetter reaching the members closest to the end of September carry campaign statements and a ballot.
- 4) That the cutoff date for receiving ballots be November 15th.
- 5) That the Board certify the election at the next Meeting after November 15th.
- 6) That the new Board take office on January 1st of the following year.

Frank lawyer asked if the time could be shorter -- whether it really took 5 months to nominate, elect, and seat a Board of Directors. William replied that, on the basis of this year's experience, it certainly did take 5 months. The Chaircritter asked if there were any objection to the above becoming Policy, and received a chorus of 'no's, and ruled that it become so.

#### Miscellany

Frank Lawyer asked if there was enough money in the Treasury to publish 9 newsletters in 1989, and asked for a Treasurer's report. The Chaircritter asked the Treasurer to prepare a formal Treasurer's report and post it to Section 8 of MUSUS ASAP. Bob Clark said he would, and reported that the current balance in the USUS account is \$5,751.50. Bob Spitzer and Frank Lawyer got into a discussion about various groups in the Washington D.C. area with whom USUS might be able to hold a joint meeting, but got sat on by the Chaircritter, since it was after 11:30 PM EST. The Chaircritter appointed them (and whoever else they could shanghai) to be the Meeting Committee, and asked that they work out the details and report back to the Board at a later date. Frank Lawyer wanted to raise membership concerns, but the Chaircritter closed discussion in view of the lateness of the hour, and opined that if the matter was too important to resolve in discussion in Section 8, that a special meeting

could be called before January 10th. William Smith grumbled that these here on-line Board meetings take entirely too long, that there should be more discussion in either Section 1 or Section 8 of MUSUS, and that what the Board should be doing at meetings is voting, not arguing. He urged that it be made Policy to handle as much of the discussion as possible offline. This proposition got a rousing cheer.

#### Next Meeting & Adjournment

The Board agreed to adjourn and meet again at 7 PM PST / 8 PM MST / 9 PM CST / 10 PM EST January 10th, 1989 in Room 1 of the MUSUS conference facility

The Special Meeting of the Board on CompuServe was adjourned at 11:40 PM EST on December 6, 1988.

Minutes submitted by:  
Samuel B. Bassett

#### **Apple SIG Doings** by Frank Lawyer

Welcome to a brand new year!

#### ELECTIONS...

I would like to thank all who voted for me in the Board of Directors elections. I will do the best that I can to represent your interests and concerns to the Board. However, don't forget that you have to supply your concerns. I will give confidentiality to all who request it, so fire away!

#### WELCOME DISKS...

Not much new here. There will be some changes on the welcome disks next month to reflect the result of the Board of Directors elections. Two new book reviews were added, but nothing major.

#### RENEWALS...

The Apple SIG is doing pretty well on their renewals, but we could do a little better. Remember, when you renew your membership, USUS provides free (and current) welcome disks for your trouble. As I write this, the end of the year is approaching, and those whose renewal was due in 1988 will begin to be dropped early in 1989. There was a renewal form in the November NewsLetter. If you can't find the one

Hays Busch sent you, use the one from the NEWSLETTER, please.

#### LIBRARY DISKS...

Disks APP2I03 and APP2I04 are announced ready as of 11/01/88. Actually, these should have been announced in the last column, but it slipped by me. The directories of the disks have been passed on to the kindly editor, who hopefully will publish them elsewhere in this issue. Copies of all the old Apple SIG columns from this NewsLetter have been moved to APP2I03, so that now they are all on one disk side.

We are still working on the new disk with the graphics routines, but had to set it aside temporarily. Also, still going through the archive disks, converting to Apple specific disks. Still need volunteers to help, so send me a late Christmas card and volunteer to do a disk.

#### SWAP SHEET...

Well, we are still working on this, but haven't received enough input from you out there to make it a reality. Send me your list of "moldie oldies", hardware or software, how much you want for it, what hardware it works on etc. Doesn't have to be p-System oriented, or even Apple oriented. Anybody need a KIM-1, 1K of RAM, no disk, 2K ROM operating system, programmable in 6502 assembly, which you punch in through the hex keyboard after you hand-assemble it? Displays memory as 6 glorious LED hex digits. Nah, I'd never sell my KIM!

#### FASTER THAN...

A product I can warmly recommend is the TransWarp™ accelerator board from Applied Engineering. This will breathe new life into your old Apple2, 2+, 2e, enhanced 2e or Franklin clone. The board contains a 65C02 micro-processor which is driven at a 3.6 MHz (megahertz or million cycles per second), compared to the standard Apple 6502 processor at 1 MHZ. This means that all the CPU operations will run about 3.5 times faster than the standard machine. The TransWarp board also contains 256K of fast RAM. In practice the entire Apple memory space (including ROM) is loaded into this RAM, and everything executed from there, using the 65C02 processor. Therefore, ALL programs that run on the Apple are accelerated.

One of the best things is that this is accomplished without any effort on the part of the user.

In test reports I have seen, this means that an Apple2e with a TransWarp board will run slightly faster than an Apple2GS. Applied Engineering states that this is 40% faster, but I have not verified this claim. Since the price of the TransWarp board is far less than the 2GS, if it's just speed which is your interest, and you don't need all the other goodies that the 2GS provides, this is certainly worth considering.

I don't know if the TransWarp board works in the old Laser128. The new Laser128EX already has a faster processor. The good news for 2GS owners is that Applied Engineering is working on an accelerator board (TransWarp GS) for that machine. From the Applied Engineering ads I read in the magazines, this board is almost ready for shipment. No news on just how fast this will be. Although some Apple2 boards work in the Apple3, Applied Engineering says the TransWarp doesn't. Too bad!

Installation is simple, but a little care has to be taken to set the proper switches on the board so that TransWarp knows which slots have disk drives and other "slow" devices so that it can slow down to 1 MHz to access them.

In use, the speedup is very noticeable. All programs I have used it with show dramatic improvement. The ones which benefit the most are the CPU bound programs like AppleWorks, Visicalc and other spreadsheets, number crunching applications, and my favorite Pascal compare program. It also improves use of the standard Editor. Text scrolling and re-formatting a paragraph using the 'M' option goes a lot quicker. It helps a lot on Pascal and Modula-2 compiles, although these can be very disk intensive also. During operation, it seems to make your disks work more intensively, since it decreases the time between disk access. If you have some big chunks of source code to compile, you will no longer have to make a cup of coffee while they chug along, you can be mesmerized by all those little dots happily dashing across the screen where they used to crawl.

There are a few minor things that bear watching. The board can be set with the on-board switches for the maximum speed used. The selections are

1.0, 1.7 and 3.6 MHz. Normally, 3.6 is the obvious best choice for a Pascal development environment. However, when it gets to games and timing dependent routines, you have to rein in and use the normal Apple speed, which you can do easily by pressing the <Esc> key at bootup time. If you think you are good at some of those shoot-em-up games, try them at warp speed! The only way to access the half speed is to set it with the switches on the board. So far, I have found no reason to do that.

My personal choice as a source for all Applied Engineering products is Preferred Computing in Dallas TX. They honor the major credit cards. They advertise in most of the Apple oriented magazines, including InCider. I have found them to be helpful and courteous when placing orders by telephone. They ship promptly. They carry the full line of Applied Engineering products, which many other mail order organizations do not. Best of all, their prices are usually lower than others too! A recent price for the TransWarp board was \$169.

#### RAMDISK...

Most Apple owners are aware that accessory memory cards can be used as simulated disk drives. This is usually referred to as a RAMDISK, since the disk is simulated by random access memory (RAM). For instance, if you have an Apple2e with the extended memory card which provides an additional 64K, (making a total of 128K) and you boot ProDOS, you will see the simulated disk as /RAM in slot 3 drive 2.

The advantage of RAMDISK is that it is very fast, as fast or faster than a hard disk drive, and therefore far faster than a floppy drive. This can speed up compiles and disk intensive operations by considerable amounts and make your time much more productive. The disadvantage is that RAM is volatile, which means that when you power off your machine, your data in RAM and on RAMDISK no longer exists. When finishing up the day, you need to copy off files to floppies or hard disks to preserve them. If you are like me, and only turn off the computer during thunderstorms, then the only problem is the local power glitch, which can eat your data in RAMDISK in a few milliseconds.

Third party board suppliers often supply software with their boards which works with ProDOS or Pascal (or even DOS3.3) to enable use of their boards as RAMDISK. These boards can hold memory into the megabyte ranges, making for awesome capacity.

Some members who have written have asked for routines to turn their memory cards into a RAMDISK, and to install that RAMDISK as the boot disk on the system. We are looking into this, as there seem to be a few routines floating around. I will report on the Applied Engineering RamDrive™ software, which allows you to use your RamWorks™ or Z-Ram AE cards as RAMDISK. This will be a separate product review. I have found routines for the old Synetix memory card, if anyone is still interested.

APDA...

To all interested in APDA (Apple Programmer's Development Assoc). I have recently been in contact with them, and I have bad news. Effective January 17th, Apple computer is pulling out all their software, documentation, including books and tech-notes, from sale by APDA.

For more than two years now, APDA has been selling Apple products, documentation, books, tech-notes, often at greatly reduced prices. Generally, these products have been products that were difficult or impossible to get through the dealer network, since they were specialized for developers. Also, they were products that Apple often chose not to support, whether by default or by design. In some cases, the products were not current products. In other cases, they were beta versions with sparse documentation. The sales by APDA were through agreement with Apple Computer.

This pullback affects all Apple products for all current hardware including the Apple2 series up

to and including the 2e, the 2c, the 2GS and all the Mac variations. APDA never had any Apple3 related products, and the 2C+ and the Mac 2X are too new to have had products available through APDA.

APDA has mailed a flyer explaining the situation to all APDA members, so I expect more detail on this very soon. In the meantime, you may do well to carefully review those Apple products available from APDA, and order what you want. Specific products of interest for Apple2 Pascal people are listed on the Apple welcome disk, and include Apple Pascal v1.3 for the bargain price of \$75 (list 150), Pascsal tools, etc.

APDA will honor orders up to January 17th, 1989 from current APDA members. Third party products available from APDA are not affected.

If you are not aware of APDA's products, send me a #10 SASE, and I will send photocopies of the listings concerned. Or, contact APDA at 290 SW 43rd Street, Renton WA 98055 (206)-251-6548 but hurry!

LETTERS...

Received a letter from Beverley Henderson and she enclosed copies of a flyer from Pre-Owned Electronics, 30 Clematis Avenue, Waltham, MA 02154. This company has a wide variety of parts available for your Apples, whatever flavor. They have logic boards, specific ICs for boards, cables, print heads, etc. Seems like a good resource to know about. They also buy and sell Apple systems, so you can buy and old 2e or an Apple3 for pretty good prices.

TransWarp, RamWorks and RamDrive are trademarks of Applied Engineering.

Frank Lawyer, 126 Demott Lane, Somerset, NJ 08873; (201) 828-3616

---

### New Library Disk Directories

What follows is a listing of the directory of the newly released Apple disks. These disks were developed by the Apple SIG. See the **Apple SIG Doings** column in the **August 88 NewsLetter** for more information on the naming conventions.

#### APP2I03:

DIR.USAR01.TEXT  
SOR.USAR01.TEXT  
DIR.USAR02.TEXT

Combined directory list of archive volumes 1-16  
Same as above, but sorted by file name  
Combined directory list of archive volumes 17-36

SOR.USAR02.TEXT	Same as above, but sorted by file name
SOR.USARAK.TEXT	Combined listing sorted by file name, starting with A-K
SOR.USARIZ.TEXT	Same as above, starting with L-Z
COMAPP2I03.TEXT	Comments on the above files
DIRAPP2I03.TEXT	You're reading it

**APP2I04:**

APSIG.01.TEXT	Apple SIG column for Newsletter of 07/88
APSIG.02M.TEXT	Apple SIG bulletin sent to members 07/88
APSIG.03.TEXT	Apple SIG column for Newsletter of 08/88
APSIG.MAC1.TEXT	Apple SIG bulletin sent to Mac members 08/88
APSIG.05.TEXT	Apple SIG column for Newsletter of 09/88-10/88
APSIG.04.TEXT	Apple SIG bulletin sent to renewing members
APSIG.06.TEXT	Apple SIG column for Newsletter of 11/88
APSIG.07.TEXT	Apple SIG column for Newsletter of 12/88
APSIG.08.TEXT	Apple SIG bulletin sent to renewing members
COMAPP2I04.TEXT	Comments on the above files
DIRAPP2I04.TEXT	You're reading it

NOTE: There are no program files or code files on an Information Disk. The files all contain text of some sort, and can be printed out on your printer, or scanned with the Editor. In general, all new SIG related articles, bulletins, Newsletter articles etc. will go on these disks. Also, some older files which appear on the Library disks will move here if they are still interesting or germane.

---

**System Level Information About Apple Pascal 1.3**

© 1988 by John C. Miller  
 110 Riverside Drive #14C  
 New York, N.Y., 10024

The various versions of Apple Pascal 1.3.

Apple Pascal 1.3 was released in four versions. Two are on the standard distribution disks, and are known as the 64K and 128K DEVELOPMENT versions. When booted, they check the boot drive for an executable program named "SYSTEM.STARTUP" and, if they fail to find it, they display the familiar system level prompt and await system level commands such as C)ompiler, E)ditor, F)iler, A)ssembler and L)inker.

The other two versions are known as the 64K and 128K RUNTIME versions. When booted, they also attempt to execute "SYSTEM.STARTUP" and, if they fail to find it, display an error message. These versions have no system level prompt. They also provide approximately 2K to 3K more memory for code and data (mostly for code in the 128K version) than do the corresponding development versions. They are, or were, available to software developers only

from Apple Computer by renewable annual license.

Each runtime version includes its own SYSTEM.LIBRARY. Most of the intrinsic units supplied in the runtime libraries are identical to those supplied with the development system, but the units PASCALIO and CHAINSTUFF, which make calls into the operating system, are different. The developer must ensure that the appropriate versions of PASCALIO and CHAINSTUFF are available in SYSTEM.LIBRARY for use by each version of the operating system. The presence of incorrect versions is undetectable by the operating system, and causes unexplained execution errors.

Some Apple Pascal 1.3 System Locations

Table 1 (*on the next page*) lists the location of some important system variables in each of the versions of Apple Pascal 1.3. These may be of interest to programmers for various types of low-level routines. Some additional system variable locations, which are the same for all four versions, are described in chapters III and IV of the Apple Pascal 1.3 documentation.

Table 1 Apple Pascal 1.3 System variable locations

	<u>dev64K</u>	<u>dev128K</u>	<u>rt64K</u>	<u>rt128K</u>	<u>bytes</u>
volume prefix	\$ACEE	\$B84A	\$B81A	\$B73E	8
boot drive name ①	\$ACF6	\$B852	\$B822	\$B746	8
system date ②	\$ACFE	\$B85A	\$B82A	\$B74E	2
on-line volumes ③	\$AD80	\$B8DC	\$B850	\$B774	240
system programs ④	\$AE70	\$B9CC	n/a	n/a	120
chain name (next program name)	\$AF08	\$BA64	\$B960	\$B884	24
chain value (universal string)	\$AF20	\$BA7C	\$B978	\$B89C	82
swap flag (0/1)	\$AF84	\$BAE0	\$B9CE	\$B8F2	2
swapGetPut flag (0/1)	\$AF82	\$BADE	\$B9CC	\$B8F0	2

① It is not possible to change the boot drive merely by changing the boot drive name. Extensive additional changes are required.

② The system date is stored in the following type of record:

**packed record**

```
Month : 0..12; { low 4 bits }
Day   : 0..31; { next 5 bits }
Year  : 0..100; { high 7 bits }
```

**end;**

③ Information about the current volumes on line is stored in the following type of record:

**array [1..20] of record**

```
VolumeName : string [7];
case BlockStrucDevice : boolean of
  true : (VolumeBlocks : integer)
```

**end;**

④ In the development versions, the user executes system programs by typing A, C, E, F or L from the system prompt. The file names of these programs are stored in the following type of record:

**record**

```
AssmVolFile : string [23];
CompVolFile : string [23];
EditVolFile : string [23];
FileVolFile : string [23];
LinkVolFile : string [23];
```

**end;**

The "SYSTEMSTUF" Unit

A unit called SYSTEMSTUF is being submitted for inclusion in the USUS library. SYSTEMSTUF contains all the routines found in the standard CHAINSTUFF unit. However, it runs with all versions of Apple Pascal 1.3.

SYSTEMSTUF also includes the following additional functions and procedures, which work with all standard hardware configurations and all versions of Apple Pascal 1.3.

1. SysType. Returns a value indicating which of the four versions of the operating system is active.
2. GetStatus. Returns an array of boolean variables indicating various characteristics of the available hardware and software.
3. CodeAvail. Returns the approximate number of words of code space available.
4. LineFeed. Enables or disables the transmission of linefeeds following carriage returns to the printer.
5. CtrlDisable. Enables or disables the special functions of the keys ctrl-S, ctrl-F, ctrl-@, ctrl-A, ctrl-Z, ctrl-E and ctrl-W.
6. xCursor and yCursor. These functions return the current screen co-ordinates of the cursor in all versions.
7. charAvail and clearKeyboard. charAvail is equivalent to the function keyPress in APPLESTUFF. clearKeyboard clears the keyboard buffer.
8. BootParams. Initializes a Pascal system disk so that, when booted, it has each of the following options either enabled or disabled (option a. works with all four Pascal systems; the others work only with the runtime systems):
  - a. 40 column text screen only (even if 80 columns available).
  - b. initial scan of all available disk drives.
  - c. swapping mode for file open/close.

- d. swapping mode for open/close and get/put.

BootParams is equivalent to the program rtSetMode which is provided with the Apple Pascal runtime systems.

- 9. InitIIGsTurtle. Provides a way to obtain sharp monochrome graphics with the turtleGraphics unit on the Apple IIGs.
- 10. Peek and poke routines for bits, bytes, words (integers) and strings.

### CommSupport Unit

by Robert Spitzer

Historically, the RemTalk unit was developed as a standard communications interface to be used by p-System software to assure portability and machine independence. There have been several problems with this unit that have been propagated along over time. On the one hand, many of the necessary low level functions of the standard (RS-232) serial communications interface are not defined. On the other hand, certain high level functions are included, that have nothing to do with serial port hardware or UARTs, but are really in the domain of a software "protocol" for dealing with serial communications.

In recognition of the need for a well designed interface to machine serial ports, Jai Gopal initiated a project about two to three years ago to develop such a standard definition. The goal was to separate hardware configuration issues from software protocols and "handshaking", while offering a more complete and flexible (and

portable) support for the standard serial interface (RS-232) which had become fairly universal.

After a series of discussions on MUSUS, in which a number of individuals participated, we reached a relatively stable definition to which no major objections remained. Jai Gopal has further gone on to promote this new standard by donating a series of implementations for machines including (at least) the Stride, IBM PC series, and Atari. These are available on line in the MUSUS libraries.

It is my purpose here to 1) acquaint all members of USUS with this extremely functional and powerful unit, 2) acquaint members of USUS who do not frequent MUSUS with this important work, 3) promote the acceptance of this unit which provides an extremely portable definition of serial communications ports, 4) promote the acceptance of this unit by soliciting contributions of implementations for other machines, and 5) re-open discussions on standard and portable approaches to serial communications, both at the hardware level and at the software protocol level. For these reasons I am presenting the interface section of this unit (only) for the benefit of the membership. Any comments, and especially any code, will be appreciated by USUS membership in general.

I also believe the interface to be a good example of mature, professional programming style, and study of this unit will be of interest to many who are seeking to enhance their own programming skills.

---

The following is a proposed USUS standard unit to be implemented for all Power System machines (Sage/Stride, IBM PC, Atari, Amiga, Pinnacle, etc.). It will allow easy and consistent configuration of asynchronous RS-232 serial ports by UCSD Pascal communication programs including those that may support multiple serial ports. Throughout this document, 'ports' are assumed to be the actual physical connections (DB-25 or RJ11) on the back of the machine, numbered 0..N. In addition to the configuration capabilities, a variety of facilities are provided to ease the task of writing communications software. This 4th draft (and implementation!) attempts to reflect the latest MUSUS discussions on this subject. Please address all comments, suggestions, and bug reports to: Jai Gopal Singh Khalsa, Compuserve ID# 72355,1015

Draft #4

unit CommSupport;

interface

const csVersCompile = 102; { changes when INTERFACE changes }

{--- Possible values of 'csResult': ---}

csOK = 0; { previous routine functioned properly }  
csNilInst = 1; { portInfo parameter was not Opened properly }  
csBadPort = 2; { port specified is not mapped/implemented }  
csBadFeature = 3; { feature requested is not supported here }  
csBadParam = 4; { value parameter passed in is not supported here }  
csBiosErr = 5; { value returned from BIOS was invalid }

**type** csPortInfo = ^integer; { opaque type }  
csTransRcv = (csTransmit, csReceive, csBoth);  
csDataBits = (csData5, csData6, csData7, csData8);  
csStopBits = (csStop1, csStop15, csStop2);  
csParity = (csNone, csOdd, csEven, csMark, csSpace);  
csFlow = (csXOnIn, csXOnOut, csRtsCts, csFlow4, csFlow5);  
csFlows = set of csFlow;  
csPin = (csDCD, csRI, csDSR, csDTR, csRTS, csCTS, csP7, csP8);  
csPins = set of csPin;  
csDteDce = (csDTE, csDCE);  
csConState = (csRaw, csCooked);

**function** csVersRuntime : integer;

{--- will match csVersCompile if interface hasn't changed ---}

**procedure** csMachine (var machineName : string);

{--- returns name of machine, implementation version # ---}

**procedure** csOpenConfig (portNum : integer; var portInfo : csPortInfo; var result : integer);

{--- creates new instance ('portInfo') associated with 'portNum' and reads the current configuration for the specified port. Reports open failures via 'result'. Note that the port configuration is not read again so attempts to control a port with this Unit and other facilities simultaneously may fail ---}

{----- IMPORTANT NOTE -----}

The following 'set' routines (csSetBaud, csSetData, csSetStop, csSetParity, & csSetFlow) do NOT affect the specified port immediately when called; instead, they set values internal to the "instance" variable, 'portInfo'. The procedure 'csSetConfig' MUST then be called to actually alter the serial port. This allows multiple config fields to be changed simultaneously.

----- }

**procedure** csGetBaud (portInfo : csPortInfo; transRcv : csTransRcv; var baudDiv10 : integer);

{--- csBoth is illegal for 'transRcv' ---}

**procedure** csSetBaud (portInfo : csPortInfo; transRcv : csTransRcv; baudDiv10 : integer);

{--- csTransmit/Receive may be illegal for 'transRcv' ---}

**procedure** csGetData (portInfo : csPortInfo; var dataBits : csDataBits);

**procedure** csSetData (portInfo : csPortInfo; dataBits : csDataBits);

**procedure** csGetStop (portInfo : csPortInfo; var stopBits : csStopBits);

**procedure** csSetStop (portInfo : csPortInfo; stopBits : csStopBits);

**procedure** csGetParity (portInfo : csPortInfo; var parity : csParity);

**procedure** csSetParity (portInfo : csPortInfo; parity : csParity);

```

procedure csGetFlow (portInfo : csPortInfo; var flows : csFlows);
procedure csSetFlow (portInfo : csPortInfo; flows : csFlows);
procedure csGetField (portInfo : csPortInfo; fieldName : string; var fieldValue : string);
procedure csSetField ( portInfo : csPortInfo; fieldName, fieldValue : string);
{--- used to support machine-specific config fields; see NOTES: below ---}
procedure csSetConfig (portInfo : csPortInfo);
{----- IMPORTANT NOTE -----
This procedure MUST be called to write 'portInfo' configuration changes to the BIOS and actually
affect the port as specified by calls to the above 'set' routines
-----}

procedure csPortType (portInfo : csPortInfo; var portType : csDteDce; var hasPins : csPins);
{--- returns type of port (DTE or DCE) and set of pins supported ---}
procedure csPinTest (portInfo : csPortInfo; var pins : csPins);
procedure csPinSet (portInfo : csPortInfo; pins : csPins);
procedure csPinClr (portInfo : csPortInfo; pins : csPins);
function csBufSize (portInfo : csPortInfo; transRcv : csTransRcv) : integer;
{--- returns size (in bytes) of buffers on this port; 'csBoth' is illegal for 'transRcv' ---}
procedure csSendBreak (portInfo : csPortInfo);
{--- causes hard BREAK signal to be issued, if possible ---}
procedure csClose (portInfo : csPortInfo; setConfig : boolean);
{--- This procedure MUST BE CALLED TO DISPOSE INSTANCE!!! If 'setConfig' is TRUE,
writes 'portInfo' configuration to BIOS (see 'csSetConfig' above) ---}
function csResult (portInfo : csPortInfo) : integer;
{--- may be called after any operation on 'portInfo'; see CONST list above ---}
function csVolPort (volName : string) : integer;
{--- returns serial port# (0..N) mapped to 'volName'; -1 indicates none ---}
procedure csPortVol (portNum : integer; var volName : string);
{--- returns UCSD volume name mapped to port# (0..N); "" indicates none ---}
function csVolNum (volName : string): integer;
{--- returns UCSD volume (i.e., unit) number; -1 indicates no such volume ---}
procedure csVolName (vUnitNum : integer; var volName : string);
{--- returns UCSD volume name including ':'; "" = no such volume ---}
function csIsBlocked (vUnitNum : integer): boolean;
{--- returns TRUE if the specified UCSD volume (i.e., unit) number is block-structured; FALSE may
indicate a serial volume/unit... ---}
procedure csConsole (state : csConState);
{--- 'csRaw' disables operating system aspects of keyboard filtering not otherwise covered by the
serial config stuff ---}
procedure csSleep (ticks : integer);
{--- pause for 'ticks'/60ths seconds; if possible, don't gobble CPU time! ---}

```

NOTES:

'csResult' values 6..99 are reserved for future use; it is expected that any machine-specific result code values will begin at 100.

The mapping functions 'csVolPort' & 'csPortVol' will be accurate but not necessarily 'fast'... If, for example, the 'portNum' associated with 'volName' is to be used repeatedly, it should be assigned to a local variable rather than have the function called each time thru a loop.

csGetField/csSetField:

This pair of procedures is primarily intended to support machine-specific features. Both parameters ('fieldName' & 'fieldValue') will be treated as case-insensitive. By convention, the following reserved 'fieldNames' have special meaning:

'hasOps1' [hasOps2, hasOps3, etc.]

When passed to 'csGetField', a string will be returned containing all of the 'fieldName' options (separated by commas) that are supported by this implementation. If all such options will fit in a single 80-character string, then 'hasOps1' will suffice and any other 'hasOps' commands will return a NUL string (i.e., they have no meaning).

'portLock'

In addition to an INTERFACE standard, this command implies an important IMPLEMENTATION standard proposed by Richard McMahon... The idea is that on any given machine, a uniform mechanism be employed by all application programs to ensure exclusive access to serial ports, regardless of whether or not this ComSupport unit is being used. Thus, the actual implementation of this feature must be examined and replicated in programs that wish to cooperate but, for whatever reason, can not utilize the ComSupport unit. The 'portLock' fieldName will respond to three possible 'fieldValues':

- 1) 'TRUE' indicates an attempt to lock the port; success := (csResult = csOK).
- 2) 'FALSE' indicates a desire to release the port, provided that it has previously been locked by this user.
- 3) 'THAW' indicates a desire to release the port unconditionally, i.e., regardless of who applied the lock.

Note that the current implementation will release a locked port in 'csClose' if the application forgets to do so; this may not always be desirable...

It should be noted that it is possible and convenient to retain the original configuration of a serial port using this opaque type approach without having to explicitly read and save all of the field values. Instead, two separate instance variables for the same port can be used:

```
csOpenConfig(portNum, origState, result);
```

```
csOpenConfig(portNum, currState, result);
```

During the course of the comm application, the 'currState' instance is used to re-configure the port. When done (i.e., after closing the 'currState' instance), a call to 'csClose(origState,TRUE);' will restore the port's original configuration.

--- IBM PC Implementation Specific Notes ---

- 1) None of the procedures relating to 'csPins' have been implemented.
- 2) "csFlow4" is "Etx/Ack"

# A Review of Apple's Lisa Pascal

by

David Craig

736 Edgewater, Wichita KS 67230

( October 7, 1988 )

## INTRODUCTION

In January 1983 Apple Computer Inc. announced their new Lisa computer. Many computer users saw the Lisa as a revolutionary machine, especially for a microcomputer in 1983. The Lisa based its user interface upon the work that Xerox PARC had done for its SmallTalk language and the Xerox Star computer. The Lisa user interface used windows, pull-down menus, icons, and the mouse. Most Lisa owners were very familiar with the Lisa Office System, the program which manages all the graphical user interface issues, but few knew of the WorkShop development environment and Lisa Pascal which made all this software possible.

The Lisa WorkShop is a command line shell similar in appearance to Apple's older Apple // and /// UCSD-like Pascal shells. The WorkShop provides access to various development tools of which the Pascal Compiler is the center piece. Originally begun in early 1980 as a port from Apple // Pascal, Lisa Pascal evolved into a very powerful compiler. This compiler generated most of the application and system code which made the Lisa work. To make the Lisa software development cycle easier Apple decided to write most of this software in Pascal. This coding effort resulted in nearly one million lines of Pascal code for the Lisa Desktop Libraries, a set of about 100 units which establish the protocols to follow across all the different Lisa applications. Even the Lisa Operating System is written in about 90,000 lines of Pascal with about 10,000 lines in 68000 assembly code for the low-level I/O and time critical portions.

## UNIQUE FEATURES

Lisa Pascal is a superset of UCSD Pascal and initially resembled Apple /// Pascal 2.0. Up to the introduction of the Lisa Apple's Pascal compilers had all generated P-code, but the Lisa Pascal Compiler, in conjunction with the Lisa Code Generator, produces native 68000 object code. When the Macintosh was introduced in January 1984 Lisa Pascal became much more powerful since the Lisa was the only initial development environment for the Macintosh. Overall, Lisa Pascal implements the following unique features (occasional 68000 code is used to illustrate the actual implementation):

### • Direct compilation of Apple // and /// source code

Since the Lisa Pascal Compiler began life as an Apple // compiler it can compile source code from either the // or ///. This source code must not contain any // or /// device specific portions since the Lisa architecture differs greatly from either machine. Once a // or /// program is compiled the Lisa screen resembles the standard 24 line by 80 column screen.

### • Long Integer type (LONGINT)

This type, implemented as a signed four byte integer, allows manipulation of integers in the range -2,147,483,648 to 2,147,483,647. To convert a regular two byte INTEGER to a LONGINT the new ORD4 conversion operator is used. The HIWRD operator extracts the high ordered word (bits 16-31) from a LONGINT, and LOWRD extracts the low ordered word (bits 0-15). These last two operators are implemented by the Compiler as follows (i is an INTEGER and L is a LONGINT):

```
i := LOWRD (L);  MOVE.W  D0,D5  { D0 = L, D5 = i }
i := HIWRD (L);  SWAP    D0
                 MOVE.W  D0,D5
```

• **Detailed compilation and code generation statistics**

The Lisa Pascal Compiler generates I-code (intermediate code) which is actually only standard UCSD Pascal P-code. The Code Generator takes the compiler's I-code file and produces 68000 object code. For a listing of the generated 68000 code the \$ASM compiler directive is used. This produces a listing with the Pascal source statements followed by their object and assembly code. The Compiler and Code Generator display a fair amount of compilation statistics. A sample follows:

```
Lisa Pascal Compiler V3.76 (05-Apr-85)          14:00:51 07-Oct-88
(c)1981 SVS, Inc. (c)1983, 1984 Apple Computer, Inc.
```

```
Input file - [.TEXT] LisaPascalTester
List file - [.TEXT]
Output file - [LisaPascalTester] [.OBJ]
```

```
[242091 words] TEST_INL
[242073 words] TEST_LEA
[242039 words] TEST_BIT
[242000 words] TEST_TYP
[241897 words] ZERO_REC
[241903 words] TEST_LAR
[242067 words] LP_TEST
```

```
Elapsed time: 1.814 seconds.
Compilation complete - no errors found. 113 lines.
```

```
Lisa Pascal MC68000 Code Generator V3.65 (20-Mar-85) 14:00:58 07-Oct-88
(c)1981 SVS, Inc. (c)1983, 1984 Apple Computer, Inc.
```

```
Input file - $I+
Input file - [.I] LisaPascalTester
Output file - [LisaPascalTester] [.OBJ] LisaPascalTester
```

```
TEST_INL - TEST_INL      Code size = 32
TEST_LEA - TEST_LEA      Code size = 178
TEST_BIT - TEST_BIT      Code size = 116
TEST_TYP - TEST_TYP      Code size = 136
ZERO_REC - $2000000      Code size = 68
TEST_LAR - TEST_LAR      Code size = 46
LP_TEST - LP_TEST        Code size = 676
```

```
Elapsed time: 2.550 seconds.
Total code size = 1252
```

• **Bit manipulation**

The bit routines manipulate the 32 bits within LONGINT values. These functions perform bit testing, setting, clearing, rotating, and shifting. The names, operation, sample Pascal code, and generated assembly code follow (b is BOOLEAN, L and L2 are LONGINT):

BCLR	Clear a bit	BCLR(L,0);	BCLR	#\$0000,D0
BSET	Set a bit	BSET(L,0);	BSET	#\$0000,D0
BTST	Test if bit = 1	b := BTST(L,1);	BTST	#\$0001,D0
BITROTR	Rotate right	L2 := BITROTR(L,1);	ROR.L	#\$1,D0
BITROTL	Rotate left	L2 := BITROTL(L,1);	ROL.L	#\$1,D0
BITSR	Shift right	L2 := BITSR(L,1);	LSR.L	#\$1,D0
BITSL	Shift left	L2 := BITSL(L,1);	LSL.L	#\$1,D0
BITNOT	Logical NOT	L2 := BITNOT(L);	NOT.L	D0
BITXOR	Logical XOR	L2 := BITXOR(L,1);	MOVEQ	#\$01,D1 EOR.L D1,D0

```

BITOR      Logical OR      L2 := BITOR(L,1);      OR.L   #$00000001,D0
BITAND     Logical AND     L2 := BITAND(L,1);    AND.L  #$00000001,D0

```

### • Loop exit control

The LEAVE keyword causes an exit of the currently executing loop (FOR-NEXT, WHILE-DO, REPEAT-UNTIL). A sample follows which prints the integers from 0 to 6:

```

PROCEDURE test_LEAVE;
VAR i : INTEGER;
BEGIN
  FOR i := 0 TO 9 DO BEGIN
    WRITELN('i = ',i:1);
    IF i = 6 THEN LEAVE;
  END;
END;

```

### • Floating point calculations based upon SANE

All floating point values are manipulated by the Standard Apple Numeric Environment (SANE), a very fast and accurate floating point engine which is based upon IEEE-standard numerics. Four new number types are available:

SINGLE	4 bytes (same as REAL)	7 place accuracy
DOUBLE	8 bytes	14 place accuracy
EXTENDED	10 bytes	19 place accuracy
COMP	10 bytes (treated as an integer)	

Several SANE constants are available:

PI	Pi (3.141592...)
MINNORMEXTENDED	Minimum EXTENDED value
MINNORMDOUBLE	Minimum DOUBLE value
MINNORMREAL	Minimum REAL/SINGLE value
MAXCOMP	Maximum COMP value
INF	Infinity (E.g., r := 1/0 --> r = INF)

### • Object-oriented programming

The Lisa Pascal Compiler supports another Pascal-like object-oriented language called Clascal. This language was originally developed as the foundation language for Apple's ToolKit libraries. Several keywords are present to define an object and its methods:

```

SUBCLASS  METHODS  CLASS  THISCLASS

```

### • Relaxed order of label, type, constant, and variable declarations

The order of LABEL, TYPE, CONSTANT, and VAR declarations is unimportant as long as no forward-like relationships exist.

### • Date and time constants

These predefined constant strings contain the compilation date and time for program version control purposes as follows:

```

COMPDATE  compilation date (E.g.,: "07-Oct-88")
COMPTIME  compilation time (E.g.,: "14:01:35")

```

### • Type coercion

Types with the same physical size allow direct assignment between their variables as seen below:

```

TYPE t_R = REAL;  t_L = LONGINT;
VAR R : t_R;  L : t_L;

L := t_L(R); { Both t_R and t_L are 4 bytes long }

```

## • Large function results

Functions can return variables whose size is greater than four bytes. An example follows:

```
TYPE t_rect = RECORD { 8 byte record }
               left, right, top, bottom : INTEGER
            END;

FUNCTION zero_rect (the_rect : t_rect) : t_rect;
BEGIN
    WITH the_rect DO
        BEGIN
            left := 0; right := 0;
            top := 0; bottom := 0;
        END;
        zero_rect := the_rect; { return the rectangle }
    END;
```

## • Procedural/Functional parameters

Procedure or function parameters can themselves be procedures or function calls. For example, the following routine from the Lisa's MathLib Unit implements a generic sorting routine where the routines Sorted and Swap are defined by the user:

```
PROCEDURE Math_Sort (first, last: INTEGER;
                    FUNCTION Sorted(i, j: INTEGER): BOOLEAN;
                    PROCEDURE Swap (i, j: INTEGER);
                    VAR error : BOOLEAN);
```

## • Object code inclusion

Assembler object code can be placed directly within the Pascal source code by using the `INLINE` keyword. The format is

```
PROCEDURE/FUNCTION name (parms); INLINE obj_code_words;
```

This stores the `obj_code_words` values directly into the program. Generally this feature is never used in Lisa-specific programs, but is used by Macintosh developers as a quick method of accessing the Macintosh ToolBox through the Trap Dispatcher. The following assembly listing contains an example of `INLINE` and the generated object code:

```
3      3 -- A      PROCEDURE test_INLINE;
4      4 --
5      5 -- B      PROCEDURE xyz (k : INTEGER); INLINE $ABCD, $1234, $5678;
6      6 --
7      7 0- A      BEGIN
8      8 --          xyz(255);
                000008 3F3C 00FF          MOVE.W  #$00FF, -(A7)
                00000C ABCD          TRAP    $ABCD ; Trap call
                00000E 1234          .WORD  $1234
                000010 5678          .WORD  $5678
                000014 4E75          RTS
9      9 -0 A      END;
```

## • Access to QuickDraw

The Lisa and Macintosh user interfaces are made possible by the QuickDraw unit. This unit implements several very powerful and fast bitmap graphic operations. For example, QuickDraw can display 4,000 characters per second, 800 lines per second, and 160 large solid rectangles per second.

## CONCLUSION

In 1985 Apple Computer stopped producing the Lisa and ended all support for it. The Lisa WorkShop and its Pascal Compiler were still supported since many Macintosh developers were using it to produce

Macintosh software. Since the Lisa development environment proved over the years to be quite powerful it was ported over to the Macintosh where it is currently known as the Macintosh Programmer's WorkShop (MPW). MPW Pascal is a direct descendant of Lisa Pascal 3.1.

Overall, I have found the Lisa WorkShop and the Lisa Pascal Compiler to be an excellent implementation of the UCSD Pascal development environment which I will continue to use until my Lisa breaks.

**<<< That's All, Folks ... >>>**

---

```

program LABEL_PRINTER;
{ A simple program written for USUS by Hays
  Busch. It is not printer dependent. It produces
  multiple mailing lables for the same addressee.
  For use by anyone for commercial or private
  purposes. }
uses SCREENOPS;
const STD_SPACE = 2;
type Str_30 : string [30];
var printer : text;
    line_1   : Str_30;
    line_2   : Str_30;
    line_3   : Str_30;
    line_4   : Str_30;
    num_labels : integer;
    ans       : char;
    quit      : boolean;

procedure TELL_USER;
var Flush: boolean;
begin
  Flush := true;
  SC_Clr_Screen;
  repeat
    SC_GOTO_XY (0, 1);
    writeln ( 'This program will print ',
              'multiple copies of up to ',
              'four 30-character wide');
    writeln ( 'lines you enter. It is ',
              'designed to use 1" by ',
              '3 1/2" adhesive labels');
    writeln ( 'and knows how to space ',
              'the labels properly. ');
    writeln;
    writeln ( 'To bypass a line, type the '
              'return key. ');
  until NOT SC_Space_Wait (Flush);
  SC_Clr_Screen;
end { TELL_USER };

procedure GET_INPUT;
begin
  SC_GOTO_XY (0,2);
  write ('Enter line one ');
  readln (line_1);
  write ('Enter line two ');
  readln (line_2);
  write ('Enter line three ');
  readln (line_3);
  write ('Enter line four ');
  readln (line_4);
  writeln;
  write ('How many labels do you want ');
  readln (num_labels);
  SC_Clr_Screen;
end { GET_INPUT };

procedure PRINT_LABELS;
var label_count, index : integer;
begin
  rewrite (printer, 'PRINTER: ');
  for label_count := 1 to num_labels do
    begin
      writeln (printer, line_1);
      writeln (printer, line_2);
      writeln (printer, line_3);
      writeln (printer, line_4);
      for index := 1 to STD_SPACE do
        writeln (printer);
      end { for };
    close (printer);
  end { PRINT_LABELS };

```

```

begin { LABEL_PRINTER }
  TELL_USER;
  quit := false;
  repeat
    GET_INPUT;
    PRINT_LABELS;
    write ( '<return>.for more labels, ',
           'Q(uit.to end program ');
    readln (ans);
  until ans in ['Q','q'];
end { LABEL_PRINTER }.

```

---

Q & / | A? ...

*I received the following reply from Howard J. Sweet (USSAH Box 404, Washington, DC 20317; (202) 726-1277; CIS: 72740,66) in response to Ken Gibbs' inquiry on page 16 of the September / October 1988 NewsLetter.*

My personal machine is a Sage IV, p-System IV.13, May 1984 and like Ken had the original Word7 and the Timberline Spreadsheet. Presently I am operating Multi-User with two users, MASTER and USER2. Both terminals are ANSI. One a VT102 and the other a VT320. Timberline as you probably know is no longer with us.

This last summer I purchased an update on Word7 and installed it on my system. The update was supplied with code files both for our old versions of the p-System and for the newer versions. I have given the new version of Word7 quite a workout and am pleased with its performance. There are several enhancements over the old Word7 when used with the old system and more improvements (untried by me) with the p-System IV.22.

Timberline, while a good product, became frustrating to me because the company was no longer in business and getting an update was impossible. I usually use spreadsheets during the development process on a program and consequently the spreadsheet gets a good workout. Noting that the same people who provide Word7 also had developed Calc7 and hearing good reports from others in the field using it, I decided to try it. Calc7 also comes with code files for the older p-System version and for the newer version. My experience with Calc7 has been enjoyable.

The word processing program, Word7, and the spreadsheet program, Calc7, both come from the same folks. I have found their product support excellent. The name of the company is: Micro Research & Development, Inc., 6135 Willow Creek Ct., Colorado Springs, CO 80919. The Word7 update cost is nominal. The spreadsheet program is in the neighborhood of \$250 or so. Both can be used with your present operating system so you can postpone your system upgrade if need be.

While the older p-System version will handle an ANSI (VT100) terminal, it will not handle the 132 character feature on for example: the VT102 or VT320. This is the only problem I have with using Word7 or Calc7 and it's really not that much of a problem. It is my understanding from Jon Bondy, JLB Enterprises, Inc., Box 148, Ardmore, PA 19003, (215) 642-1057 that the newer version p[ower]-System does have ANSI support in it. Additionally, it requires the p[ower]-System IV.2.2 to run the Power System which, as I understand it, has all the communication goodies in it.

Should you decide to go for the new Word7 and Calc7 I can probably help you out with the VT100 configuration.

*I received a letter from Carey Bloodworth (P.O. Box 17, Swink, OK 74761):*

Do you know of any Pascal compilers available in high level source? I have several executable ones for my PC, but I just bought a 68000 based machine and there is only a C compiler for it.

Also I would like to know if you know of any C to Pascal translators, a chess program in Pascal or any utilities such as YACC and LEX for Pascal.

*I received a letter from Rob Allen (Telephone Systems, 1013 M.L.King Blvd., Tallahassee, FL 32301)*

I have been searching for the following software for a Pinnacle 1A:

- 1) More utilities other then 'EPA' by Jai.
- 2) 32 bit integer routines for Intel machines. (There is a Pascal implementation of a 32 bit unit for the p-System in one of the MUSUS Libraries named A16A32.)

- 3) Word processor that will import text files.
- 4) Spreadsheet.
- 5) Pecan software, absolutely has no support for Pinnacle - Must be hard feelings between the two. Does anyone know - Will the Power System for Stride run on a Pinnacle?

*I also received a letter from Marc De Niel (in Brussel Belgium):*

"Can anyone tell me when AFS (Advanced File System) will be available from Pecan. In June 1987, I ordered and paid for several software packages from Pecan, among which the AFS. They all work fine, except for the AFS, which I never received: it was said to be on back-order, and now, nearly 1.5 years later, it still is.

Pecan does not answer my letters. The answer given on the telephone varies from "real soon now - we just have to finalize the documentation" to "no date set". I understand the AFS is not one of Pecan's top priorities, but that being so they should not have put it on their product list 1.5 years ago.

I have been using the p-System for more than 4 years now on a Sage II computer, and I do think it is very good software. Stride's service, as well as JLB Enterprises's, Stride's agent for Sage software, is excellent, but Pecan's deserves a rather worse qualification. As some people said or feared in previous USUS NewsLetters, individual users are left in the cold by Pecan."

*I called Pecan's order number on November 23 and talked to Mark Abrams and ask about AFS. I was told that it was not sold as a separate package, but was only sold as part of the development package. I then informed him that I had already purchased the development system (through JLB Enterprises's). He then checked and told me that I could order the AFS for \$250.00 and it was shipping now.*

*I then informed him I was the USUS NewsLetter editor and had received the above letter from Marc. He stated that Pecan did not deal with Europe and that their European office must be contacted. Their address is Pecan Software Europe, Victoria House, 10 Kellaway Ave.,*

*Henleaze, Bristol, BS67XR and phone is 0272-425012.*

*I also contacted Robert Spitzer and asked him if he had received his copy of AFS. His reply was:*

I never did get AFS. I did speak to Liaison today, and they are finally sounding co-operative about honoring an order that goes back about three years. I believe they have been balking because they changed their minds and want to reserve AFS as one of the aces up their sleeve to promote their new high-end approach. I think they recognize that the low end beat-Turbo-at-their-own-game is a dead end, and have gone entirely with the "high end contract" approach.

I wish them luck, in general, because I believe that the survival of the p-system depends on their success. As to honoring prior orders, this is the proper thing to do, and I think the only approach is for folks who placed such orders to make a fuss with Liaison/Pecan directly, and hope they behave as gentlemen. I was told that AFS was running long before Pecan bought the p-system from SofTech, and I have spoken to numerous folks who had copies back then (including MicroSage) who said it worked.

Whether the current version is different I doubt. Liaison is claiming to include it with their current high-end system, which they claim is available now. What all this double-talk means is that I will believe it when I see it. I'll keep you posted.

*I had sent a message to Eli Willner prior to the talking to Mark Abrams asking "What is Pecan's official position on AFS (and for what machines)" and received the following reply afterward.*

AFS was made a Liaison only product. (This is a marketing, rather than a technical decision). AFS runs on all implementations that are up to IV.2.1 or later, including the VLM (32-bit) implementations. Using AFS implies that you have a totally different file system from the "standard" p-System filer system. The I/O units in SYSTEM.PASCAL, as well as the file management units, are AFS-specific. The filer is a totally different utility in AFS than it is in SFS.

*From this I conclude: AFS is finished and working. It is available from Liaison, but not*

*from Pecan. Those who ordered it from Pecan prior to it becoming a Liaison only product will, as suggested by Bob, need to keep pressuring Pecan to honor their commitments.*

### **We get Letters...**

Dear Mr Smith: November 28, 1988

I have been a member of USUS for more than 5 years, number 2881. The readers of the USUS NewsLetter might find it useful to learn about my unhappy experience with Pecan.

I ordered System-32 (VLM) update with Modula 2 from Pecan for my Mac II on June 28 after being told on the phone it would be available "in two weeks." I sent in my check for \$604.45 with my order. I heard nothing from Pecan. I wrote Eli Willner on September 3 and again on October 13 after having received no reply to my first letter. Still without a reply I wrote again on November 1 to cancel my order and receive an immediate refund. Finally, having no response I phoned Pecan to speak with Eli Willner and was told he was out to lunch (at 3 p.m.). These fact speak for themselves.

Sincerely, Lester G. Telser, Professor of Economics (The University of Chicago, Department of Economics, 1126 East 59th Street, Chicago, Illinois 60637).

*I received the following letter via EasyPlex. It nicely leads into my From the Editor column.*

Dear William,

I received my latest newsletter and though I have been meaning to write you a letter about it one thing after another has prevented me from sitting down at the word processor. Finally, I seem to have a few minutes free.

In my opinion, too much attention is being spent on board actions in the newsletter. I belong to an organization (the Writers Guild of America) which just went through a period of revitalization. It took us several years to resuscitate a bureaucracy that had become moribund and one of the keys to our success was the use of our newsletter.

The first thing we did was to identify the purpose of the newsletter. For the WGA, the purpose was to inform our membership of policy

decisions and to educate them so that when our contract expired they would understand the issues and support the Guild's positions. We, too, had board meetings and we wanted our members to be aware of everything that transpired. So as not to bore everyone to death, we published a "brief" of the minutes which was mailed out to all the members within twenty-four hours of the board meeting. When the detailed minutes were ready we published them as an addendum to the newsletter.

What has this got to do with USUS?

I think the NewsLetter is wandering, torn between reporting the bureaucratic functionings of the board of directors and also providing information about the p-System. I think the balance is skewed way too far toward reporting the activities of the Board of Directors and that this is going to cost you membership interest -- and ultimately -- members.

Over two thirds of the last issue were devoted to Board actions. There was very little room left for p-System talk. I think you have the mechanism, MUSUS, to provide a report of board actions for those who want it. If you set up a forum to report on and discuss board actions it would then leave the whole newsletter available for p-System discussions. Though I have not put it to a formal poll, I would guess that very few people (aside from the combatants) are interested in the details of Board infights. If anyone is, you could certainly mail them board minutes on request. My guess is that this would be very inexpensive.

A newsletter can be a very powerful instrument for revitalizing a moribund institution -- be it the WGA or USUS. Though it is coming out regularly now (a great improvement over the previous newsletter) I do not think that it is being used nearly as effectively as it could be. I think the general USUS membership wants to know the ins and outs of the p-System; that is where I think the focus of the newsletter ought to be.

For what it's worth...

**Adam Rodman**

P.S. -- I do not mean this letter to be insulting in any way. It is my hope that my experience with a similar situation might prove helpful. It is clear that intelligence and energy are being applied to

solving the problems of USUS and I wish you nothing but the best in this endeavor!

**From the Editor**

by William D. Smith

This is the first issue of the new year. Last year I demonstrated that USUS can put out this NewsLetter on a regular basis. Now we have to work on the content. As Adam pointed it out in the previous letter, the percentage of technical stuff in most NewsLetters is too low. He assumes that the NewsLetter will stay the same size as the December 1988 issue, 12 pages. On the contrary, that is the smallest size issue to go out. The normal size should be 24 pages, as in this issue. Of this one-third should be bureaucratic details and the remaining two-thirds should be technical or semi-technical in nature. This means I have to have the articles to print, which means YOU have to write and send them to me.

In answer to Adam's question, this NewsLetter has three purposes:

- 1) Keep the members informed of what the USUS is doing. This is in the form of the **Administrator Says** column, the **Treasurer's Report**, the **Board of Director Minutes**, the **SIG** columns and other committee and project reports. This is well covered at the present time.
- 2) Provide the members a way of raising issues which are of concern to themselves. This is done by the way of **We Get Letters...** and the **Q & / | A?** column. Letters and responses are starting to pick up and will hopefully continue.
- 3) Provide technical material of interest to the members. This is done through **articles** and the **Tips** and **MUSUS Tips** columns. This is the area we will need to work on this year. As the NewsLetter motto is "All the News that Fits, We Print" says, if it gets sent to me, it will be printed.

It seems that many USUS members use more than just the p-System. I know that I do. I use the p-System to support one client which has a complex database written in UCSD Pascal and running on a PDQ-3 (which uses the  $\mu$ Engine

chip set on an Dec LSI-11 board). This database is slowly being ported to run on a  $\mu$ Vax under Oracle. I also use Vax Pascal (very different from UCSD Pascal) and MPW Pascal and TML Pascal under MPW on the Mac. These last two Pascals both compile the same programs and also compile UCSD Pascal programs with very few changes (as long as don't want to use the Mac toolbox).

Good articles in this vein would be comparisons between the different Pascals (or Modula 2s) and UCSD Pascal. Reviews of the different implementations. What kind of issues or problems do you run into when moving from one implementation or language to another. What do you need to do to support multiple implementations in one source file. Etc.

Note that this is not strictly a p-System user group (although most of the members use or have used the p-System in some way). I will accept for publication articles on Pascal and Modula 2 regardless of the implementation.

Most USUS members use computers for something. Write and tell what you use them for. If you do programming, what languages and development systems you use. Share your knowledge and experience with others. With your help, we can make this NewsLetter something to look forward to.

Page	Article
1	SAGE/STRIDE News
1	Orphan Software Project
3	Administrator Says
4	Treasurer's report
4	Board of Directors Minutes (Dec. 6,1988)
6	Apple SIG Doings by Frank Lawyer
8	New Library Disk Directories
9	System Level Info About Apple Pascal 1.3 by John C. Miller
11	CommSupport Unit by Robert Spitzer
15	A Review of Apple's Lisa Pascal by David Craig
19	program LABEL_PRINTER
20	Q & / I A? ...
22	We get Letters...
23	From the Editor William D. Smith
End	You're reading it

### NewsLetter Publication Dates

NewsLetter	Due date	Code/Forms	Articles	Due date	Short stuff
March 89	02/01/89		02/10/89	02/17/89	02/17/89
April 89	03/01/89		03/10/89	03/17/89	03/17/89
May/June 89	04/15/89		04/28/89	05/05/89	05/05/89
July/Aug 89	06/15/89		06/30/89	07/07/89	07/07/89

**Next NewsLetter coming February**

USUS  
P.O BOX 1148  
LA JOLLA, CA 92038

ADDRESS CORRECTION REQUESTED