Washington Apple Pi

$2

π

Volume 6   November 1984   Number 11

Highlights

1001 BINARY TALES: MAC INNARDS - 2
MACPASCAL: WHAT'S ON THE MENU?
SUPPORTING INSIDE MACINTOSH
COMPUTER MUSIC - A REPLAY

In This Issue

There's only one place to buy apples.
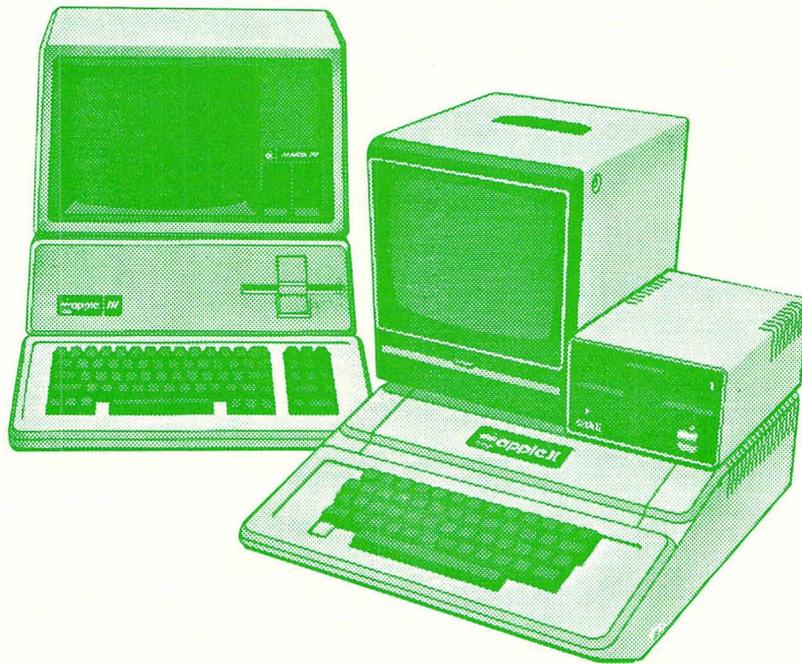
OFFICERS & STAFF

President	- David Morganstein	(301) 972-4263
Vice Pres-Programs	- Tom Warrick	(301) 656-4389
Vice Pres-SIGs	- Bob Platt	(806) 378-1065
Treasurer	- Edward Myerson	(703) 759-5479
Secretary	- Nancy Little	(301) 762-3215
Directors	- Bernie Benson	(301) 951-5294
	- Peter Combes	(301) 251-6369
	- J.T. (Tom) DeMay Jr	(301) 779-4632
	- Bruce Field	(301) 340-7038
	- Nancy Philipp	(301) 924-2354
	- Jay Thal	(202) 244-3649
	- Rich Wasserstrom	(703) 893-9147
Editor	- Bernie Urban	(301) 229-3458
Associate Editor	- Genevie Urban	(301) 229-3458

Journal Staff:
Store Distrbtn.	- Ray Hobbs	(301) 490-7484

Columnists:
Applesoft	- J.T. (Tom) DeMay Jr	(301) 779-4632
DisabledSIG	- Jay Thal	(202) 244-3649
EDSIG	- Peter Combes	(301) 251-6369
LOGOSIG	- Nancy Strange	(703) 691-1619
Q & A	- Bruce Field	(301) 340-7038
Telecomm	- Dave Harvey	(703) 527-2704
VisiCalc	- Walt Francis	(202) 966-5742

Review Coord.:
Hardware	- Scott Rullman	(301) 779-5714
Software	- Raymond Hobbs	(301) 490-7484
Group Purchases	- Rich Wasserstrom	*(301) 654-8060
	*(Call Kevin at office on M,W,Th,F)	
Disketeria Staff:	- John Malcolm	(301) 384-1070
	- Dave Weikert, Joy Aso, Ed Lang,	
	- Jim & Nancy Little, Pat Foreman,	
	- Fred Edwards, Gordon Stubbs	
New Disks	- Dana Schwartz	(301) 725-6281
Pascal Lib.	- Mike Hartman	(301) 445-1583
CP/M Lib.	-	
Head Reading Lib.	- Walt Francis	(202) 966-5742
Apple Tea Coord.	- Paula Benson	(202) 951-5294
Arrangements	- Jim Taylor	(301) 926-7869

Demonstrations	- Signe Larson	(703) 524-4541
General Counsel	- Jim Burger (Burger & Kendall)	
	day	(202) 293-7170
Membership	- Dana Schwartz	(301) 654-8060
Program	- Cara Cira	(301) 468-6118
Publicity Chairman	- Hunter Alexander	(703) 820-8304
Public Relations	- Lee Raesly	(301) 460-0754
Rules & Elections	- Bob Platt	(806) 378-1065

School Coordinators:
Virginia	- Barbara Larson	(703) 451-9373
	- Nancy Strange	(703) 691-1619
Mont. Co. MD	- Margie Stearns	(301) 657-2353
Pr. Geo. Co. MD	- Conrad Fleck	(301) 699-8200
Special Publctns.	- Betsy Harriman	(202) 363-5963
SYSOP	- Tom Warrick	(301) 656-4389
Tutorials	- Steve Stern	(301) 881-2543
	- Leon Raesly	(301) 460-0754
Volunteer Coord.	- Sue Roth	(703) 356-9025
SIG Coordinator	- Bob Platt	(806) 378-1065

SIG Chairmen:
Appleseeds	- Mike Forman	(703) 241-1216
Apple ///	- Jerry Chandler &	(703) 790-1651
	- Bill Hershey	(301) 588-1992
CESIG	- John Kapke/	(301) 694-5968
	- Roy Rosfeld	(301) 340-7962
CP/M	- Charles Franklin	(301) 997-9138
DisabledSIG	- Jay Thal	(202) 244-3649
EDSIG	- Peter Combes	(301) 251-6369
Forth SIG	- Kevin Nealon	(703) 280-1136
LAWSIG	-	
LISASIG	- Gordon Stubbs	(703) 750-0224
LOGOSIG	- Nancy Strange	(703) 691-1619
SIG Mac	- Steve Hunt	(301) 262-9080
NEWSIG	- Bernie Benson	(301) 951-5294
Pascal (PIG)	- Harry Bishop/	(703) 931-4937
	- Jim Harvison	(301) 593-2993
PI-SIG	- Raymond Hobbs	(301) 490-7484
STOCKSIG	- Robert Wood	(703) 893-9591
Telecomm. SIG	- George Kinal	(202) 546-7270

EDITORIAL

On a distant planet in a star system far, far away there was a small band of cam, asil and cIIa owners who called themselves SPAW. They had banded together for safety, mutual reassurance, increased knowledge and, yes, a tad of selfishness.

They subscribed to the attainment of the SPAW principle, "The whole is greater than the sum of its parts", but because of the tainted emmig which they drank they succumbed to a virulent disease which rendered them disinclined to contribute to the common good. Members began to cry that there weren't enough seidoog to go around.

There were some among them who, recognizing the dangers of too much emmig, began to partake rather of evig. Those who switched to evig produced more seidoog and soon became the leaders. And, lo, the band flourished. In time, more and more of the members saw that this was a good thing. They switched also and the SPAW principle was indeed attained. The moral of this story is, "I would rather evig than emmig". How about you?
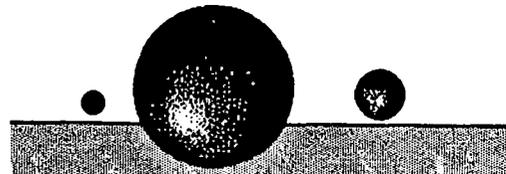
PRESIDENT'S CORNER
by David Morganstein

TRIBUTE TO BOB PLATT. As you read in Bernie's comments last month, Bob Platt, Vice-President in charge of SIGs and special publications has taken a position in Texas. The Texans' gain is, indeed, our loss. While we expect to hear from Bob regularly and to continue receiving his fine contributions to the journal, it won't be the same. He has served the WAP in too many ways to recant in this short column. His advice, dedication and energy can not help but be missed. We all thank you, Bob, for all you have done to help so many.

FAT MAC, SLOW SOFTWARE. The Macintosh has taken a major step forward with the introduction of the "Fat Mac", a 512K version of Apple's machine for the people. The Macintosh is by no means the answer to everyone's problem. It still suffers from being the newest kid on the block, especially because of its unique operating system, disk format and programming limitations. (As you may be aware, Apple expected only LISA equipped owners with 1Mbytes of memory and a hard disk would be able to write applications software for it.) However, almost 400 of our members own a Mac and that number continues to increase rapidly. Each month new software releases appear and the expansion to 512K should presage even more powerful programs.

WAP now has a modified Mac which can display video output on a projector. At this month's SIG Mac meeting, the 125 attendees were able to see the Mac video via an Electrohome projector.

THANKS TO KEVIN AND PAUL. Two of WAP's unsung heros spend about 20 hours at our office each week helping hundreds of members and doing the many tasks needed to keep us going. Kevin Nealon and Paul Koskos, bless them both, contribute a great deal to help insure the WAP can grow and continue to serve us all. Many thanks to you two!!

APPRECIATION TO RAY HOBBS. Have you been seeing Ray's name alot lately? Besides being the PI SIG chairman, Ray has conducted numerous tutorials, helps to distribute the newsletter to the local dealers and, on top of all that, coordinates the software reviews for the Journal. His energies and talents result in a lot of information to our members and we want him to know we appreciate it deeply.

SIG NEWS. While some of our SIGs continue to hold regular meetings with valuable programs, I am afraid that some seem to be in hibernation. Charles Field, due to other commitments, must bow out of the LAWSIG leadership. Are there any lawyers who want to help take up the challenge? We have spoken often of the possibility of a Medical SIG but there seems to be little activity there. Some //c owners have expressed interest in a separate group dedicated to their machine, especially since it does contain a few surprises. The once active Games SIG has just about disappeared and the Appleseeds have not shared with us any program they have planned in the near future. If there is interest in these or other topics, step right up and we will see what can be done to get things going again.

SIG ELECTIONS. As many of you are aware, the fall is time for SIG elections. It is important that volunteer organizations provide for fresh input and energy from new members and relief for the volunteer of long standing. Each SIG should elect a chairperson, sec-

retary and librarian. The secretary should forward regular, albeit brief, reports of SIG activities for publication in the Journal. In this manner others may decide to participate. The librarian should work with our new disk librarian, Dana Schwartz to direct the flow of software into our collection.

USUHS. As many of us are coming to realize, the USUHS entrances are being changed. Henceforth, you should plan on entering from the Wisconsin Ave entrance. Just follow the signs to the University. The Jones Bridge Road entrances are going to be closed on Saturdays and evenings.

MEMBER REFERENCE MANUAL. We are in the process of revising the reference manual. Bob Oringel and the Urbans have completed an outline and assembled the material for a printing. They are hoping to complete the editing sometime this fall and send the copy to the printers. This important project will bring much needed information to the newer members and may provide some hints for the oldtimer, as well. The format will consist of three volumes: a general one of interest to every member (the first to be printed), an Apple][family library description and a Macintosh library description. The later two volumes will go to members owning that particular computer and interested in disks for that machine.

JOURNAL GRAPHICS ADVISORY COMMITTEE. The JGAC (whew) is a new idea suggested by our Editor, Bernie Urban. He is looking for thoughts about improving the appearance of the Journal. If you have any you are de facto a member of his "informal" committee. Please call or visit with him if you have suggestions or talent that could help!

DEMISE OF SOFTALK. It was with considerable disappointment and sadness that we learned of the bankruptcy of Softalk publishing. The WAP has sent a letter to the Tommerviks expressing our concerns and hoping that they will be able to re-organize and return. In the letter we asked what plans they had to honor outstanding subscriptions. Their publications for the Apple, Softalk, and the Mac, St. Mac, were exceptional and served the user community well. Several years ago, they helped the WAP with a free ad, worth, we are sure, a considerable amount at the time. But they wanted to help user groups and offered that assistance to us. (We had also approached the International Apple Core about an ad in the Apple Orchard and were told we would have to pay the full commercial rate even though we were a member and a not-for-profit organization...) If there is anything we can do to help the Tommerviks re-start operations, we would like to know of it.

CLUB MAC OF NEW ORLEANS

Club Mac of New Orleans is a support group organized for the benefit of Macintosh computer owners. However, owners of other types of equipment are welcome, as are interested parties and their spouses and friends. If anyone would like additional information contact Simon Streiffer at (504) 831-8275 or write 111 Atherton Drive, Metairie, LA 70005. The meeting dates for the next few months are:

Tuesday November 6
Tuesday November 20
Tuesday December 11
Tuesday December 18

SIG NEWS

APPLE /// SIG meets on the second Thursday of the month at 7:30 PM. The next meeting will be on November 8 at Walter Reed Institute of Research. From 16th Street entrance go 3/4 around circle. Go in North entrance of the 4-story building on your right and ask the guard for Room 3092.

APPLESEEDS is the special interest group for our younger members. They meet during the regular WAP meeting.

DISABLEDSIG - See the DisabledSIG column elsewhere in this issue. Call Jay Thal for details.

EDSIG - the education special interest group - see the EDSIG Page elsewhere in this issue.

FORTHSIG will hold its next meeting on Saturday, November 17 at 1:00 PM in the WAP office.

LISA SIG meets after the SigMac meeting on the second Saturday of the month.

LOGOSIG meets monthly at 12:45 after the regular WAP meeting at the Barrie School, 13500 Layhill Road, Silver Spring, MD.

NEWSIG will meet just after the regular Washington Apple Pi meeting and conducts a "drop-in" for new Apple owners on Thursday evenings from 7:30-9:00 PM in the office. They will answer questions and try to help new owners get their systems up and running.

PIG, the Pascal Interest Group, meets on the third Thursday of each month at 7:30 PM at the Club Office. The round-table discussions for the next few months are:

November 15 - Discussion of Apple ATTACH BIOS 1.1 and 1.2

December 20 - p-System, Apple][(+,c,e) computers and ink pen plotters.

January 17 - p-system, Apple][(+,c,e) computers and input digitizers.

PI SIG (formerly ASMSIG) meets on the second Monday of each month at 8:00 PM in the WAP office. See PI Sig News elsewhere in this issue. For further details, call Ray Hobbs at 490-7484.

SigMac meets on the 1st Thursday of each month at 7:30 PM and on the 2nd Saturday from 9:00 AM to 12:30 PM at USUHS, in the auditorium.

STOCKSIG meetings are on the second Thursday at 8:00 PM at the WAP office. For further details call Robert Wood, (703) 893-9591.

Telecomm SIG usually meets after the regular WAP meeting.   ⑮

BUGS AND OTHER PARASITES

In the September 1984 article "Forth Answers the Challenge", an error was made in the definition of FACTORIAL. The proper definition is:

FACTORIAL DUP 2 < IF DROP 1 ELSE DUP 2 = IF ELSE
    DUP 1- MYSELF * THEN THEN ;.   ⑮

MINUTES

SEPTEMBER GENERAL MEETING

Washington Apple Pi, Ltd. met at the USUHS on September 22, 1984, at 10:00 AM, with David Morganstein presiding. Members were reminded that ours is a volunteer organization and that to help one another we volunteer our time. Proceedings were interrupted by the fire alarm and the building was evacuated in an orderly way. After reassembling, Ed Myerson presented the budget and moved that the budget be accepted with approval to raise the annual dues by $2, $4 or $6 if necessary. Motion seconded and carried. Dave Harvey reported from the ABBS committee. Executive Board meetings are open to all members. Open question and answer sessions are held at the WAP office on Thursday evenings. Dana Schwartz announced that the new membership directory would be mailed in about 8 weeks. The directory will be based on the membership as of September 22, 1984. Bob Platt has moved out of the area and Dana Schwartz will now receive programs contributed to the WAP disketeria.

SUMMARY OF OCTOBER EXECUTIVE BOARD MEETING

The Executive Board of Washington Apple Pi, Ltd. met on October 10, 1984 at the WAP office. Tom Warrick presided. The membership directory is being prepared to be sent to the printer on October 15. WAP's book edited by Bob Platt, "Pascal for the People" will be published in January, 1985. The new bookkeeper is hard at work. The ABBS final report and recommendations are to be presented at the January Board meeting. Bernie Urban reported on the status of the Member Handbook. The time and place for the November Boaard meeting has not yet been determined.   ⑮

NOTICE

In the August Washington Apple Pi Journal Dana Schwartz, Membership Chairman, notified the membership that copies of the forthcoming Membership Directory would be available only to those persons whose names appeared in it, and that no copies would be sold. In publishing this notification Dana expressed what was the then current intention of the Board of Directors.

On reconsideration, particularly in view of the needs of new members, the Board has changed the policy.

Extra copies will be printed and will be available for purchase only by new members who agree to being listed in the next issue of the directory.   ⑮

GENERAL INFORMATION

Apple user groups may reprint without prior permission any portion of the contents herein, provided proper author, title and publication credits are given.

Membership dues for Washington Apple Pi are $25.00 for the first year and 18.00 per year thereafter, beginning in the month joined. If you would like to join, please call the club office or write to the office address. A membership application will be mailed to you. Subscriptions to the Washington Apple Pi Journal are not available. The Journal is distributed as a benefit of membership.

Current Office hours are:

Monday - Friday - 10 AM to 2:30 PM
Tues. & Thurs. - 7 to 9:30 PM
Saturday - 9:30 AM to 12:30 PM (except meeting Sat)
- 12:00 to 3:30 PM (meeting Sat only)

* November 1984 *

WAP

SUNDAY	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY
				1 Sig Mac 7:30 PM USUHS; DisabledSIG 7 PM CCCC	2	3
4	5 Deadline for Decembr Journal Articles	6 Beginning Tutorial #1 7:30 PM Office	7	8 Stock-SIG 8:00 PM Office; Apple /// 7:30PM-WR	9	10 SigMac-9 AM USUHS; LISA SIG after SigMac
11	12 PI SIG 8:00 PM Office	13 Beginning Tutorial #2 7:30 PM Office	14	15 Pascal SIG 7:30 PM Office ATTACH BIOS	16 Executive Board 7:30 PM Office	17 Forth SIG 1:00 PM Office
18	19	20 Beginning Tutorial #3 7:30 PM Office	21	22 Happy Thanks-giving!	23	24 WAP Meeting 9:00 AM USUHS
25	26	27 Beginning Tutorial #4 7:30 PM Office	28	29**NEWSIG "Help"** Every Thurs 7:30-9:00PM Office	30 Deadline for January Journal Articles	

* December 1984 *

WAP

SUNDAY	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY
						1
2	3	4 Begin. Tutorial #1 7:30PM-Off; EDSIG-USUHS 7:30 PM	5	6 Sig Mac 7:30PM-Lady of Lourdes; DisabledSIG 7 PM CCCC	7	8 SigMac 9:00 AM USUHS; LISA SIG after SigMac
9	10 PI SIG 8:00 PM Office	11 Beginning Tutorial #2 7:30 PM Office	12 Executive Board 7:30 PM Office	13 Stock-SIG 8:00 PM Office; Apple /// 7:30PM UC	14	15 WAP Meeting 9:00 AM USUHS Garage Sale
16	17	18 Beginning Tutorial #3 7:30 PM Office	19 Happy Holidays	20 Pascal SIG 7:30 PM Office-ink pen plotrs.	21	22
23	24	25 Happy Holidays	26	27**NEWSIG Help** Every Thurs 7:30-9:00PM Office	28	29
30	31 New Year's Eve-Have a good one!					

CLASSIFIEDS

COMMERCIAL CLASSIFIED

JOB MART

TECHNICAL OCCUPATIONS EMPLOYMENT GROUP

The following information about Technical Occupations Employment Group (TOEG) may be of interest to WAP members:

TOEG is a private, non-profit organization that has been established in Montgomery County to provide a free job matching service for high technology professionals. If anyone should desire to make use of their services, please call 231-0044 to make an appointment.   ❦

EVENT QUEUE

Washington Apple Pi meets on the 4th Saturday (usually) of each month at the Uniformed Services University of the Health Sciences (USUHS), Building B, 4301 Jones Bridge Road, Bethesda, MD, on the campus of the National Naval Medical Center. Library transactions, journal pickup, memberships, etc. are from 8:45 - 10:00 AM. From 9:00 to 10:00 AM there is an informal "Help" session in the auditorium. The main meeting starts promptly at 10:00, at which time all sales and services close so that volunteers can attend the meeting. A sign interpreter and reserved seating are provided for the hearing impaired.

Following are dates and topics for upcoming months:

    November 24 -
    December 15 - Swap Meet
    January   26

The Executive Board of Washington Apple Pi meets on the second Wednesday of each month at 7:30 PM at the office. All members are welcome to attend. (The November meeting is being scheduled on an alternate date, Friday, November 16.)   ❦

WAP HOTLINE

Have a problem? The following club members have agreed to help. PLEASE, keep in mind that the people listed are VOLUNTEERS. Respect all telephone restrictions, where listed, and no calls after 10:00 PM except where indicated. Users of the Hotline are reminded that calls regarding commercial software packages should be limited to those you have purchased. Please do not call about copied software for which you have no documentation. If the person called has a telephone answering machine, and your call is not returned, don't assume that he did not try to return your call - perhaps you were not home. Try again.

General	John Day	(301) 672-1721
	Dave Harvey	(703) 527-2704
	Robert Martin	(301) 498-6074

Accounting Packages
| Accountant(Dec.Sup.) | Mark Pankin | (703) 524-0937 |
| Home Accountant | Leon Raesly | * (301) 460-0754 |

| APPLE SSC | Bernie Benson | (301) 951-5294 |

| Apple TechNotes | Lance Bell | (703) 550-9064 |
| | Shirley Weaver | (301) 761-2479 |

AppleWorks	Carl Eisen	(703) 354-4837
	J.J. Finkelstein	(301) 652-9375
	Jay Jones (Balt.)	(301) 969-1990

Communications Packages and Modems-Telecom.
Anchor Mark 12	George Kinal(7-10)	(202) 546-7270
	Jeremy Parker	(301) 229-2578
Apple CAT II	Ben Acton	(301) 428-3605
ASCII Express	Dave Harvey	(703) 527-2704
BIZCOMP Modem	Jeremy Parker	(301) 229-2578
General	Tom Nebiker	(216) 867-7463
Hayes Smartmodem	Bernie Benson	(301) 951-5294
Omninet	Tom Vier (1-6 PM)	(202) 887-7588
VISITERM	Steve Wildstrom	(301) 564-0039
XTALK CP/M Comm.	Bernie Benson	(301) 951-5294

Computers, Specific
Apple //c	John Day	(301) 672-1721
	Scott Rullman	(301) 779-5714
Apple //e	Scott Rullman	(301) 779-5714
Lisa	John Day	(301) 672-1721
	Don Kornreich	(301) 292-9225
Macintosh	Jay Heller	(301) 948-7440
	Scott Rullman	(301) 779-5714
	Tom Warrick	(301) 656-4389
	Donald Schmitt	(717) 334-3265

| Corvus Hard Disk | Tom Vier (1-6 PM) | (202) 887-7588 |

Data Bases
dBase II	Paul Bublitz	(301) 261-4124
	John Staples	(703) 759-3461
DB Master	Doug Daje	(301) 868-5487
	Dave Einhorn	(301) 593-8420
Data Perfect	Leon Raesly	* (301) 460-0754
Data Factory	Bob Schmidt	(301) 736-4698
General Manager	Normand Bernache	(301) 935-5617
	Leon Raesly	* (301) 460-0754
InfoMaster	Doug Daje	(301) 868-5487
List Handler	Jon Vaupel	(301) 977-3054
PFS	Bill Etue	(703) 620-2103
	Ben Ryan	(301) 469-6457
	Jenny Spevak	(202) 362-3887
QuickFile II	J.J. Finkelstein	(301) 652-9375

| Expediter Compiler | Peter Rosden | (301) 229-2288 |

| Graphics | Bill Schultheis | (703) 538-4575 |

Languages (A=Applesoft, I=Integer, P=Pascal, M=Machine
A	Peter Combes	(301) 251-6369
A,I	Jeff Dillon	(301) 422-6458
A	Richard Langston	(301) 258-9865
A	Mark Pankin	(703) 524-0937
A	Leon Raesly	* (301) 460-0754
A,I,P,M	Bill Schultheis	(703) 538-4575
A,I,M	Richard Untied	(703) 241-8678
A,I,M	John Love	(703) 569-2294
M	Raymond Hobbs	(301) 490-7484
P	Dottie Acton	(301) 428-3605
P	Donn Hoffman	* (412) 578-8905

Languages, contd.
Forth	Bruce Field	(301) 340-7038
LOGO	Ron Murray (eve.)	(202) 328-3553
LISP	Fred Naef	(703) 471-1479

| Math/ O.R. Applns. | Mark Pankin | (703) 524-0937 |

| Monitor, RGB Color | John Day | (301) 672-1721 |

Operating Systems
Apple DOS	Richard Langston	(301) 258-9865
	Richard Untied	(703) 241-8678
CP/M	Robert Fretwell	(703) 971-2621
	Ray Hobbs	(301) 490-7484
ProDOS	Richard Langston	(301) 258-9865

| Paddles | Tom Riley (eve.) | (301) 340-9432 |

Printers
General	Walt Francis	(202) 966-5742
	Leon Raesly	* (301) 460-0754
Apple Color Plotter	John Day	(301) 672-1721
Apple Daisy Wheel	John Day	(301) 672-1721
Apple Dot Matrix	Joan B. Dunham	* (301) 585-0989
Daisywriter 2000	Bill Etue	(703) 620-2103
	Henry Greene	(202) 363-1797
IDS 460	Jeff Stetekluh	(703) 521-4882
Imagewriter	John Day	(301) 672-1721
	Scott Rullman	(703) 779-5714
MX-80	Jeff Dillon	(301) 434-0405
NEC 8023	Bill Mark	(301) 779-8938
Okidata	Fred Feer	(703) 978-7724
	Scott Rullman	(301) 779-5714
Silentype	Bruce Field	(301) 340-7038

Spreadsheets	Leon Raesly	* (301) 460-0754
	Walt Francis	(202) 966-5742
Lotus 1-2-3	Walt Francis	(202) 966-5742
	Roy Rosfeld	(301) 340-7962
Multiplan	Terry Prudden	(301) 933-3065
VisiCalc	Walt Francis	(202) 966-5742
	Leon Raesly	* (301) 460-0754
Spreadsheet 2.0 (MagiCalc)	Leon Raesly	* (301) 460-0754

| Statistical Packages | Jim Carpenter | (301) 371-5263 |
| | Mark Pankin | (703) 524-0937 |

| Stock Market | Robert Wood | (703) 893-9591 |

| Tax Preparer-H.Soft | Leon Raesly | * (301) 460-0754 |

| Time-Sharing | Dave Harvey | (703) 527-2704 |

Word Processors | Walt Francis | (202) 966-5742 |
Apple Writer II	Doug Daje	(301) 868-5487
	Dianne Lorenz	(301) 530-7881
	Leon Raesly	* (301) 460-0754
Executive Secretary	Louis Biggie	(301) 967-3977
Format II	Henry Donahoe	(202) 298-9107
Gutenberg	Neil Muncy Can.	(416) 298-3964
Letter Perfect	Cara Cira	(301) 468-6118
	Leon Raesly	* (301) 460-0754
Magic Window and II	Joyce C. Little	(301) 321-2989
Peach Text	Carl Eisen	(703) 354-4837
PIE Writer/Apple PIE	Jim Graham	(703) 643-1848
ScreenWriter II	Peter Combes	(301) 251-6369
Supertext II	Doug Daje	(301) 868-5487
	Peter Rosden	(301) 229-2288
Word Handler	Jon Vaupel	(301) 977-3054
	Christopher Romero	(703) 471-1949
Work Juggler //e	Carl Eisen	(703) 354-4837
Word Star	Christopher Romero	(703) 471-1949

*Calls up until midnight are ok.

Q & A
by Bruce F. Field

I had a question a few months ago about using a Vision-80 card with Apple Writer II. Jim Clark of Martin, Tennessee writes that there is a public domain preboot disk available from Vista. (With any eighty column board the manufacturer is probably most likely to know of the existence of a preboot disk or software patches for their word processor.) Jim also notes that there is some difficulty when the Vision-80 card and some printer interfaces clash over the $C800 ROM space - "I get all italics and cannot turn underlining off once it has begun, but the price is right! Besides, one can always print from the 40 column mode."

Q. When using ScreenWriter with my Apple][+, how come I'm getting I/O errors when I try to save a "get buffer" to a disk? I thought I could save a couple of pages at least that way, but I keep getting errors when I try to use the file I save via $ as an input file. All that gets saved is from less than one screen to a couple of screens - not all that I put into the get buffer.

A. Markers are used in ScreenWriter to set off a piece of text that is to be moved to another part of the document, and this piece of text is stored in the "get buffer". Unfortunately although the Screenwriter documentation claims it is possible to store this buffer to disk there is a bug in the program that causes it to work incorrectly as you observed. There is no known fix at this time; you'll either have to not use it, or bug (pun intended) Sierra for a fix.

Q. I need to be able to keep track of deductible expenses for a year - business expenses, personal deductions, etc. I tried a couple of accounting programs and got fed up - I couldn't recover from an I/O error, couldn't print out all expenses for the year in a particular category, couldn't include entries from several checkbooks and from cash. At the moment I'm using VisiCalc, a column for the check ID, one for the payee, one for the amount, and one for an accounting code. But I don't know how to sort these according to accounting code. Could you recommend a sort program that will work with VisiCalc and an accounting or data base program that will do what I need?

A. The easiest (but not the least expensive) way to get around the sorting problem with VisiCalc is to use another spreadsheet program that has sorting capability such as Multiplan. All the data you presently have stored in VisiCalc can be moved over to Multiplan without retyping. There is a sorting program available that can be used with VisiCalc files; a friend remembers it as "VisiSort". I have not been able to find any references to this program under the VisiSort name, but I am assured that it does exist. This would require that you take your data from VisiCalc, sort it, and send it back. The bother may not be worth it.

It's hard to recommend an accounting program based on a little information, but from what you say a data base management program may suit your needs. If you want to be able to generate very specific reports and don't need more than column sums or simple arithmetic on columns, a filing program may be sufficient. PFS File coupled with PFS Report,

or Apple's Quickfile both provide simple filing and sorting with the ability to produce fairly sophisticated reports for small sets of data.

Q. I have an Apple //e with a Zoom Telefonics Net-worker modem. I purchased this modem with the software "The Netmaster" and I have been having problems downloading programs in machine language. Do you have any suggestions? Does ASCII Express Pro work with this modem?

A. I know of no problems peculiar to the Netmaster when downloading binary data. The usual precautions apply; make sure you have set the modem for the same number of bits as the sending system. This is usually 8 data bits with no parity bits for binary data. This is different than with ASCII (text) data where only 7 bits are used and an eighth bit is usually sent as a parity bit. I believe the Networker emulates the Apple Communications Card and thus is compatible with ASCII Express Pro.

Q. The //e uses 4164 DRAM memory chips which are 150 ns chips. Is there any advantage in using 120 ns chips.

A. The speed with which the memory runs in the Apple is determined by the clock signals. As long as the memory chips are fast enough there is no advantage in using faster chips. If on the other hand you have a bad chip and all you can get is the faster version it will work fine. In general there is a disadvantage to using faster chips than is necessary; the chips consume more power creating more heat, and are more expensive.

Before you ask me how to change the clock speed in the Apple to make it run faster, let me point out that the video generating circuitry is tied in very closely with the microprocessor clock circuits and one cannot be changed without the other. This is why the "speedup" boards on the market have another 6502 microprocessor running at a faster speed on the board.

Q. I have an Apple][+ with a Grappler+ and a Hayes Micromodem II. When the Micromodem is installed I am unable to print complete documents. The printer will start printing and will stop in what is the equivalent of half a page. To get more of the document printed I have to "de-select" the printer and re-select it. This problem occurs only when the Micromodem is installed; when removed all works well. I have moved the modem and Grappler to several different slots.

I have had long conversations with both Hayes and Orange Micro. Hayes agreed to look at the modem card. They returned it and there is no change. Can you offer a suggestion?

A. These two cards are very popular and there must be hundreds (if not thousands) of computers running with this combination. I think the problem may come from marginal operation of one or more of the cards, or the Apple itself. I suggest you remove all the cards from your Apple except the Grappler

contd.

and the modem and see if the problem continues. Then try your cards in someone else's Apple. You might also try someone else's Grappler with your modem (and vice-versa) if possible. None of these are much of a solution, but you may be able to identify the weak element and get a replacement.

You might also try shortening and/or rearranging the placement of the cable from your Grappler to the printer. If an especially long cable is used you could pickup unwanted signals from other sources (perhaps a modem).

Q. Using Apple Writer //e I inadvertently saved a blank screen with the same title as my text file. I can view the file using COPY II+ "View Files" program but I cannot retrieve the file for further editing. Any suggestions short of typing it back in again?

A. Every once in a while Apple Writer hiccups and does the same thing to me. Apple Writer II and //e save a file with a null character (hex 00) at the end. When you saved the new file all it consisted of was a hex 00 which overwrote the first byte of the old file. All you have to do is use a "disk zap" program that lets you read and write any sector on the disk and change the first byte in the first sector from 0 to some ASCII character, preferably what it was originally.

Q. When I tried to boot up a disk, I got the following message:

B6FF- A=D0 X=5F Y=F0 P=34 S=F9
*

What does that mean?

A. That means you have a problem. Actually Apples spit out messages like this very often when something goes wrong. What it means is that a machine language program, such as DOS or the Applesoft interpreter, has executed some incorrect instructions. The asterisk prompt at the end indicates you are running the Apple's Monitor program that is stored in ROM memory and it is waiting for an input command. The numbers on the line above indicate the address at which the bad instruction was encountered and the status of the microprocessor registers at that time. These numbers almost never provide any useful information for discovering what went wrong. (They are however useful for debugging your own machine language programs where you deliberately insert BREAK instructions to stop execution of the program.)

For your problem where the disk doesn't boot, I would try another disk. If the same problem occurs then try removing and inserting all the peripheral cards (WITH THE POWER TURNED OFF!). Usually the problem is corrosion (oxide) on the contact fingers on the edge of the card and is not particularly visible. Removing and replacing the card causes the contacts in the connectors on the Apple motherboard to scrape away some of the oxide and make a better contact.

Q. I have been trying to understand the "WAIT" command in Applesoft by reading the reference manual; however, I don't understand Greek. Could you shed some light as to how and when to use WAIT in programming?

A. This command is not very popular because it is not very useful. It is designed to stop an Applesoft program from running until some device sends a signal to the Apple, and then the program continues running again. For example, suppose you have a wire from your printer to the Apple that tells the Apple whether or not there is paper in the printer. Let's further suppose that the status of the wire can be found by reading memory location 49296. Memory location 49296 contains a value consisting of 8 bits of information. If the wire is connected so that the 8th bit is a one if the printer has paper and a zero if it doesn't, we don't care about the other 7 bits; they are meaningless in this case. Still with me? Now we have to understand the AND and XOR functions. In this context we are not talking about the Applesoft AND and OR functions; they are different. What we are doing here is to take two bytes of data and compare each bit in a byte with the corresponding bit in the other byte. For instance,

value	binary	
208	11010000	
128	10000000	
-----	--------	
128	10000000	AND
80	01010000	XOR

If we AND 208 and 128, for each column where there is a 1 in both numbers (when they are written in binary) there is a 1 in the ANDed result. XOR stands for exclusive-OR and means that there is a 1 in the result everywhere there is a 1 in one but not both of the numbers. Now how do we use WAIT? If WAIT is used with two arguments the first one is a memory location. WAIT gets the value of this memory location and ANDs it with the second argument. WAIT 49296,128 will get the value at 49296 (our printer status), AND it with 128 and produce either a zero (if the eighth bit is turned off, i.e. the value is less than 128) or a positive number (128, if the eighth bit is on). If the result is positive, Applesoft continues executing the next statement. If the result is zero it goes back and reads the memory location again. Thus it "waits" for the value of the memory location to change (to something greater than 128 in this example).

To make this more complicated, suppose the printer was wired backwards and the 8th bit of memory location 49296 was a one if the printer was out of paper. Using our example above the program would wait until you took the paper out, not exactly what you usually want. WAIT can also be used with a third argument to fix this problem. In this case the value of the memory location (the first argument) is XORed with the third argument. For every 1 in the third argument the corresponding 1 in the value gets changed. Then this new number get ANDed with the second argument and if the result is not zero the program continues. WAIT 49496,128,128 XORs the printer status with 128; if the paper is out the 8th bit is a 1 and XORing it with 128 changes it to a zero. Then we AND this result with 128 and since the 8th bit is a zero the result is a zero, and the program does not continue.

The most practical use is with the keyboard. WAIT can be used to wait until a key is pressed and will not display any cursor. Use it like this.

```
100 POKE -16368,0 : REM RESET KEYBOARD STROBE
110 WAIT -16384,128 : REM WAIT FOR KEYPRESS
120 A = PEEK(-16384) : REM A CONTAINS THE ASCII
                        CODE
```

Q. After reviewing the "Fast Garbage Collection" source code I found something I couldn't understand. In line 59 (line 60 in the "All About Applesoft" printing) there is a JSR NZTAB. NZTAB is a short routine which sets the values in the work areas to zero. The unusual thing is that NZTAB doesn't end in an RTS, it ends in a JMP FNDVAR2, which seems to be the top of the main loop. This leaves four seemingly important instructions unexecuted, waiting for an RTS. Unless I'm mistaken, "Fast Garbage" seems to execute twice through for each call to it: once after the first JSR NZTAB... JMP FNDVAR2, and then again after the whole routine finishes and the address of the original JSR NZTAB gets popped off the stack by the final RTS. Of course, the second time works correctly because the four "skipped" instructions are finally executed. Please tell me, am I right about this?

A. Looks to me like you are. The easiest way to find out (although I haven't tried it) is to change the instruction JSR NZTAB at line 59 to JMP NZTAB and move it to just before FNDVAR2. The four instructions that were missed on the first pass before will be executed before jumping to NZTAB and it is safe to return to FNDVAR2. This should speed up the program considerably. Since this was first printed in January 1981 and reprinted in Call A.P.P.L.E. in Depth, volume 1, All About Applesoft, it seems funny that no one noticed this (including me and I typed it in by hand). As you commented though, the program does eventually operate correctly as written.

non-VERBAL COMMUNICATION AIDS
by Bob Mackay & Betty Wawrzaszek

At the October 4 meeting of the DisabledSIG, Sean Wawrzaszek and Yancy Mackay demonstrated their communications systems to a large gathering of those interested in augmentive communications for the speech impaired.

Both Sean and Yancy have cerebral palsy and are wheelchair users. The gross motor ability of each young man is impaired, with fine motor control such as finger touch or pointing being non-existent. Both received an evaluation for determination of augmentive communication systems suitable to their needs. Sean had his first evaluation at Tufts University Biomedical Engineering Dept. several years ago. A ZYGO 100 communication system was prescribed. His second evaluation was conducted at George Mason University last year. Yancy had his evaluation conducted at TRACE center, University of Wisconsin. His prescribed communication system is a Prentke Romich EXPRESS 3.

The presentations revolved around each demonstrator and were delivered by their fathers, with the young men structuring messages on their respective systems. An explanation of Sean's interface with the Apple //e computer was described. This is a modified leaf switch which is mounted on his wheelchair headrest. Sean's reliable control interface is the lateral movement of his head. An adaptive firmware card is used to bypass the keyboard with alphanumeric characters appearing on the lower portion of the display screen. He demonstrated some software programs, one having several menus containing information which can be retrieved and coupled with messages on the screen. Another program demonstrated an ability to retrieve whole words from a sizable vocabulary. This involves scanning pages of words which are alphabetically arranged and structuring a message therefrom. Words not appearing in this program can be spelled out individually. The immediate objective is to speedup word character selection and they are considering the use of a second switch to make this possible. Much discussion on this ensued and some participants offered some excellent suggestions.

The need to explore the suitability of available software programs was discussed at some length. Again, some valuable comments and suggestions were obtained. The availability of WALDO and its capability to control Sean's environment was discussed. Since this technology is relatively complex, Sean has not yet mastered the use of WALDO and was therefore unable to demonstrate it. It was mentioned that a modem was being installed which will enable Sean to communicate with others. He is looking forward to this.

Next, Yancy's father described the workings of the EXPRESS 3 and its capabilities. The EXPRESS 3 is a microprocessor, has storage memory capability and can be hooked up with a computer such as the Apple //e. In this latter case, the EXPRESS 3 uses an adaptive device which bypasses the computer keyboard. The EXPRESS 3 is capable of row-column scanning or direct selection. Yancy uses a foot switch to operate his device. While in school last year his communication system was often used in conjunction with the Apple //e. He now relies on the system's capability to synthesize his messages or to print them. The EXPRESS 3 has a visual readout which enables Yancy to edit his work before the message is emitted via voice or print.

A question and answer period ensued and both young men communicated their answers. They would appreciate some assistance in developing methods for speedier communication, having software use explained and instruction in basic programming skills. A contribution of a couple of hours per month (by anyone who is willing to assist) would be most welcome. Call Bob Mackay, home 256-3798, work 351-5582; or S. Wawrzaszek home 256-4889.

CERTIFICATE OF APPRECIATION

Stan Gowin, one of the early members of Washington Apple Pi (#617) has been awarded a Certificate of Appreciation by the Columbia Lighthouse for the Blind for the time and services he donated this past summer to the Lighthouse's Day Camp for Visually Impaired Children. Stan not only instructed the children in the use of computers but also brought his personal Apple][+ to the campsite.

While presenting the Certificate, the Executive Director of the Columbia Lighthouse expressed sincere appreciation on behalf of the staff and the campers, "for Mr. Gowin's civic awareness in contributing his time and efforts to instruct the visually handicapped."

Presently Stan is installing a computer system for the local chapter of the American Foundation for the Blind. The system, which will generate correspondence in Braille and print, will consist of an Apple //e with the Echo II Speech Synthesizer, Visualtek DP-10 large print processor, a Kramner modified Perkins Brailler, and an AJ-832 printer.

Stan, who is legally blind, has been a programmer analyst at the American Security Bank for the past 2 1/2 years. His interests include reading science fiction, collecting records from classical to rock, and playing chess - often with his computer. Stan and his wife Susan (also a WAP member) live in Annandale, VA.

DISABLEDSIG NEWS
by Jay M. Thal

* * * * * * * * * * * * * * * * * * * *

DISABLEDSIG NOVEMBER MEETING
Thursday, November 1, 1983, 7:00 P.M.
Chevy Chase Community Center
Connecticut Ave. & McKinley St.,NW, D.C.
SPEAKER: Martin E. Salsbury, Director of Education
Patuxent Institution, Jessup, MD
SUBJECT: Symbiosis between prison inmates and the
disabled which benefits both in rehabila-
tation. Inmates have been providing ser-
vices to the blind in the past, and are
now working with computers.

* * * * * * * * * * * * * * * * * * *

On October 4, the DisabledSIG managed to pack the hall
for what turned out to be a very important meeting.
The significance of the meeting was a demonstration by
two youths, ages 18 and 21, of how computers and other
adaptive hardware and software allow them to communi-
cate. The two, limited to only controllable head and
foot movements by Cerebral Palsy can now exercise some
control over their environment.

Frankly, I feel inadequate in describing their equip-
ment, efforts and courage, so I am leaving that to
Stephen Wawrzaszek and Bob Mackay, parents, to
describe it elsewhere in this issue.

* * * * *

This coming spring, March 20-25, 1985, the Johns
Hopkins University School of Continuing Studies will
present a conference designed for special education
teachers, administrators, rehabilitation and service
personnel.

This conference will be offered in cooperation with
TAM (Technology and Media), a division of the Council
for Exceptional Children. The program will examine
computer applications that address the needs of
individuals who are mildly-to-severely handicapped.
The program will feature nationally recognized pro-
fessionals experienced with applying computer tech-
nology to the needs of persons with handicaps. Edward
J. Cain, Jr., president of TAM, will be the keynote
speaker.

Lectures and workshops will be held March 21-23, at
the Omni International Hotel in Baltimore, MD. A
pre-conference workshop, on March 20, will include
tours of the John F. Kennedy Institution, the Maryland
Rehabilitation Center, and Johns Hopkins' Applied
Physics Laboratory.

For more information, or to register, contact the
Division of Continuing Education, Johns Hopkins
University, 102 Macauley Hall, Baltimore, MD 21218,
(301) 338-8500.

* * * * *

ANNOUNCEMENT: Visually impaired individuals who are
reading this column, as well as the WAP Newsletter,
from diskette rather than the printed page are asked
to return those diskettes as soon as the articles are
digested (do not feed the disks to your pet Piranha).
The number of disks outstanding is considerable and
this service is more costly to maint in than the
printed page.

* * * * *
NEW PRODUCTS FOR THE DISABLED

About the time you read this the November '84 BYTE
will be out with Steve Ciarcia's newest project -- a
VOICE RECOGNITION and speech synthesis board called
the Lis'ner 1000. Details are not known, but for
price it promises to be the least expensive yet, even
assembled. It will support at least a 64 word vocab-
ulary, and comes with the SSI 263 third-generation
phonetic synthesizer chip. A & T $259, kit $219. If
you already have the SSI 263 and software (if you have
the earlier Sweet Talker II) you can insert the chip
on the basic recognition board, A & T $189, kit $149.
All units come with software and headband microphone.
The system is based on General Instrument's SP1000
chip, which G.I. claims will also "run backwards" as a
synthesizer with a soon-to-be-released allophone set.
The above items are for the Apple. MicroMint, 561
Willow Ave., Cedarhurst, N.Y.

TALKING TERMINAL PROGRAM: Exec Software announced
Taking TermExec which works with an Apple][and an
Echo][synthesizer. It is known to work with the
Apple Super Serial card (and some clones), the Hayes
Micromodem and the Volksmodem. Documentation comes
both printed and on audio cassette. $80. Exec
Software, 201 Watham St, Lexington, MA 02173; (617)
862-3170.

For the HEARING IMPAIRED two firms have announced
wrist-watch based receivers which detect up to six
sound sources (via radio - doorbell, phone, etc), vi-
brates on the wrist, and identifies the source.
Price range - $150-175. SwissTronics, Executive Tower
Suite 502, 3300 W. Mockingbird, Dallas, TX 75235;
(214)350-2892 AND Quest Electronics, 510 S. Worthing-
ton St., Oconomowoc, WI 53066; (414) 567-9157.

* * * * * * * * * * * * * * * * * * *

DECEMBER 6, MEETING PREVIEW: Dr. Lawrence Scadden, on
how technology will figure in the life of the dis-
abled -- education and employment.

EDSIG NEWS
by Peter Combes

EDSIG Calendar

THERE WILL BE NO MEETING IN NOVEMBER - so as to improve the turn-out for the U.S. Presidential elections.

Tuesday, December 4, 1984 at 7.30 p.m.

"The Apple runs educational machines."

All EDSIG meetings are held in the Auditorium, Building B, of the Uniformed Services University of the Health Sciences, on the campus of the National Naval Medical Center, 4301 Jones Bridge Road, Bethesda, MD.

Meeting Report

Tuesday, October 2, at 7.30 p.m.

"Educational software". Demonstrations and discussions by EDSIG members.

Some of the gates to USUHS were closed for obscure military reasons, but those members who found their way around enjoyed a "hands on" evening looking at various educational authoring systems.

Paul Dragun showed software on a //e loaned for the occasion by Comm Center of Laurel.. The portability of the //c was much in evidence when sitting beside the hulking mass of the chairman's][Plus, with its disk drives and clumsy monitor.

Paul gave EDSIG a rare opportunity to see the ITTS Authoring System, by Scott Stewart and Phil Wissenberg. This is not currently available commercially, but "up for grabs", should an enterprising publisher care to take it on. It enables the teacher to integrate multiple choice questions into a visually attractive "board game". Comments can be built in for correct and incorrect answers. The documentation is simple but straightforward, and the entire package is most appealing.

Micro-Lab Computer Products of Highland Park, Illinois has a wide range of commercially available software, from arcade games to data base management systems. Their arcade game Miner 2049er is perhaps the best known. Their educational section -- Micro Learn -- produces The Learning System, by Scott Kamins, the author of "How to Make Your Programs User Friendly". This program makes tests that can be edited by the teacher. The system can be ordered to produce a randomly selected set of questions from a master file, thus making it possible to give each member of a class a different test. A score is produced on the teacher's disk, which can also analyze which questions cause the most difficulty. Unusually, the program is in BASIC, and is open to the user for modification. The manual helps such modification by explaining how the files are arranged and used.

CLASS, by Touch Technologies, is written in 6502 assembly. It is reported to be very easy to learn and use. It can be used to put "blanks" in a text for the pupil to fill. Most unusually, it incorporates the line editor from DOS Tool Kit for text editing. It provides no pupil scoring and no record keeping.

Datatech Software Systems produce Mentor Master, a system "custom made for Biology". It permits the teacher to write instruction, to call up hi-res pictures from a picture file, and creates a student diskette. Paul mentioned a school in Texas that was establishing itself as a clearing house for educational hi-res pictures.

Project Seraphim, working under a grant from the National Science Foundation, is developing a formidable collection of CAI, available at four dollars a disk. Paul showed a lake ecology simulation, in which the student could follow a number of investigative paths to find out why the fish in a lake were dying. Although Seraphim copyrights its programs, BASIC source code is supplied so that users can "modify" and adapt them to suit local conventions and tastes".

The Privens - Judy and Lou - gave us an insight into work being done for Edu-Ware. At earlier meetings, the Privens have shown us something of the extensive work they have done for Edu-Ware in the area of SAT testing. Their use of high resolution graphics throughout, and careful attention to screen layout have resulted in some very attractive looking presentations.   ✇

//c SIG NEWS

Your contributions, please!

PI SIG NEWS
by Raymond Hobbs

PI SIG meets at the WAP office, 8227 Woodmont Avenue, on the second Monday of each month, at 8:00 P.M.

Last month we continued our investigations into operating systems, this time focusing on PRODOS. We have also been discussing music synthesis and the Macintosh ROM routines and operating concepts. Some of the results of our investigations will be appearing in the WAP Journal, but it's more informative (and more fun) to come out and participate in the meetings!

Elections will be held in October for SIG officers for the coming year - results will be reported here.

We at PI SIG would like to extend our best wishes to Bob Platt, down in armadillo land. Like so many others at WAP, we will miss him.

For more information on PI SIG (who we are, what we do), call Ray Hobbs (301) 490-7484 or Mike Vore (301) 992-4953.   ✇

FORTH ASSEMBLY OF UNSTRUCTURED CONDITIONAL BRANCHING

by Chester H. Page

There are many situations in which the simplest logic flowchart leads to conditional branches which are easy to program in conventional 6502 assembly, but are not allowed in structured programming. The following example is part of a STEP/TRACE program:

```
LOOP      BIT $C000
          BPL LOOP
          LDA $C000
          CMP #$D4
          BEQ PROCEED
          CMP #$D3
          BNE LOOP
          BIT $C010
PROCEED --------
        --------
```

Note that there are 2 returns to LOOP and 2 chances to escape from the large loop. In FORTH, this would involve 2 upaired IFs inside a BEGIN-UNTIL loop, with both THENs outside the loop. How to achieve the same logic with "properly" nested paired conditionals is not obvious, but there is an obvious solution to the problem, namely, provide for arbitrary conditional jumps in the assembler.

I added four words to Ragsdale's FORTH assembler, as follows:

```
: BACK.JUMP, ( N---) 2* HERE SWAP ! ;
: JUMP.BACK, ( N---) 4C C, 2* @ , ;
: FORWARD.JUMP, ( N---) 4C C, 2* HERE SWAP ! 0 , ;
: END.JUMP, ( N---D) 2* @ HERE SWAP ! ;
```

Using there, the above routine can be written as

```
HEX CODE TEST
1 BACK .JUMP,
C000 BIT,
0< NOT IF, 1 JUMP.BACK, THEN,
D4 CMP,
0= IF, 2 FORWARD.JUMP, THEN,
D3 CMP,
0= NOT IF, 1 JUMP.BACK, THEN,
C010 BIT,
2 END.JUMP,
(PROCEED ETC.)
END-CODE DECIMAL
```

Note that additional overlapped IF, THEN, pairs can be handled by 3 FORWARD.JUMP and 3 END.JUMP, etc.   ⌘

BINARY CATALOG PROGRAM
by Steven Pearce

In the course of teaching myself 6502 assembly language, I have found that I often need to know the starting address and length of binary disk files. I searched through all the reference manuals and magazines that I have for a program to do this, and was unable to find anything suitable. I then took the time to write a program called "Binary Catalog" which will do just this.

The program is written in standard assembly language on the A.P.P.L.E. BIG MAC Assembler. It searches a disk in drive #1 for all locked and unlocked binary files, prints the filename, then finds the first file sector of each file and prints the starting address and length in hexadecimal. The program lives below $800 where Applesoft starts, and uses the existing DOS file buffers at $9600 and $9700, so you can BRUN the program without crashing a BASIC program which might also be in memory.

The program is set up to run on an Apple //e with an 80-column card without modification. It will also run on a 40-column //e, but will add two linefeeds to each carriage return without an 80-column card active. I have not tested it with other 80-column cards.

The program starts out by setting up a Data Control Table and Input/Output Block using DOS buffer space at $9600 and $9700 for file buffers. Thus, MAXFILES must be 3 or greater when running the program or it will clobber an Applesoft program in memory. The program itself uses memory in the input buffer area, so it's best if it is always BRUN from disk rather than BLOADing it and doing a CALL 704.

It begins by loading the first sector of the Catalog and scanning for locked and unlocked binary files. It will ignore non-binary and deleted files. When it finds a binary file, it locates the sector with that file's first Track/Sector list and loads it into the second buffer at $9700. Using this information, it then locates the disk sector containing the beginning of the binary file. This sector is then loaded and the first four bytes are the starting address and length of that file. These are then printed with the appropriate "$") and the Catalog is searched for the next binary file. With the Apple //e 80-column card active, the standard monitor routine CROUT issues a linefeed to the screen without a carriage return, so the cursor must be explicitly set back to zero (lines 162-165) to obtain a carriage return. An alternative would be to send a hex 8D to the screen.

That's how it works. I have seen many programs which will print out the start and length of the last BRUN program using $EA60 and $EA72, but none which do a full catalog of the disk (except within commercial packages such as COPY][+ by Central Point Software). The program is completely relocatable. Below is an example of the output and the listing of the program.

Example (WAP New Member's Disk):

```
  2COG3
DEMO.TABLE                  A$6000   L$002E
CPLOT+                      A$0300   L$0090
UPPER CASE                  A$0800   L$0200
LOWER CASE                  A$0A00   L$0200
AMPERSORT II                A$9000   L$0600
LHS DISK MAP                A$0803   L$0444
B.MAS.CAT.48K               A$9000   L$051B
INTEGER BASIC-DISK          A$0800   L$14B9
AMPERSORT DOC. (BRUN)       A$0B5C   L$28A7
HIRES.BIN                   A$0C00   L$0400
SUPPLEMENT                  A$8D00   L$0428
DSPEED.OBJ                  A$0900   L$00F0
DOMUMENTATION               A$0B5C   L$1C08
```

```
:ASM
1     *******************************************************
2     *                                                     *
3     *                  BINARY CATALOG                      *
4     *                      BY                              *
5     *                  STEVE PEARCE                        *
6     *                                                     *
7     *                                                     *
8     *     PRINTS HEX STARTING ADDRESS AND LENGTH           *
9     *            OF BINARY FILES                           *
10    *                                                     *
11    *              SEPTEMBER, 1984                         *
12    *           BIG MAC.LC ASSEMBLER                       *
13    *                                                     *
14    *******************************************************
15    *
16    RWTS     EQU   $3D9        ;DOS RWTS ROUTINE
17    BUFF1    EQU   $9600       ;BUFFER FOR CATALOG SECTOR
18    BUFF2    EQU   $9700       ;BUFFER FOR FILE SECTOR
19    CROUT    EQU   $FC66       ;MONITOR CR ROUTINE
20    COUT     EQU   $FDED       ;MONITOR CHAR OUTPUT ROUTINE
21    PRBLNK   EQU   $F948       ;MONITOR PRINT 3-SPACES
22    PRNTAX   EQU   $F941       ;MONITOR ROUTINE - PRINT HEX
23    *                           OF A,X REGS
24    CATT     EQU   $EC         ;STORE TRACK OF NEXT CATALOG
25    *                           SECTOR
26    CATS     EQU   $ED         ;STORE SECTOR OF NEXT CATALOG
27    *                           SECTOR
28    DOSWRM   EQU   $3D0        ;WARMSTART DOS
29    HOME     EQU   $FC58       ;HOME CURSOR
30    COUNTER  EQU   $EE         ;STORE LENGTH OF FILENAME
```

Computer Specials

```
                       31  *                               TO BE PRINTED
                       32  CH       EQU    $57B            ;HORIZ CURSOR POSITION,
                       33  *                               //e 80-COL CARD ($24 FOR 40 COL)
                       34  *
                       35           ORG    $2C0
02C0: 20 58 FC         36           JSR    HOME
02C3: D8               37           CLD                    ;SET TO BINARY ARITHMETIC
                       38  *
                       39  * SET UP I/O BLOCK WITH ADDRESS OF
                       40  * DATA CONTROL TABLE (DCT).
                       41  *
02C4: A9 CB            42           LDA    #DCT
02C6: 8D C0 03         43           STA    DCTL
02C9: A9 03            44           LDA    #>DCT
02CB: 8D C1 03         45           STA    DCTH
                       46  *
                       47  * LOAD FIRST CATALOG SECTOR INTO BUFF1
02CE: A9 11            48           LDA    #$11
02D0: 8D BE 03         49           STA    TRACK
02D3: A9 0F            50           LDA    #$0F
02D5: 8D BF 03         51           STA    SECTOR
02D8: A9 96            52           LDA    #$96
02DA: 8D C3 03         53           STA    BUFFER
02DD: 20 B2 03         54           JSR    READISK
02E0: A2 01            55  START1   LDX    #1
02E2: BD 00 96         56           LDA    BUFF1,X
                       57  * TRACK OF NEXT CATALOG
02E5: 85 EC            58           STA    CATT
02E7: E8               59           INX
02E8: BD 00 96         60           LDA    BUFF1,X
                       61  * SECTOR OF NEXT CATALOG
02EB: 85 ED            62           STA    CATS
02ED: A2 0B            63           LDX    #$B             ;TRACK OFFSET
02EF: A0 0D            64           LDY    #$D             ;FILE-TYPE OFFSET
                       65  *
                       66  * GET TRACK OF FILE. IF =$FF, THEN FILE
                       67  * HAS BEEN DELETED.  IF =0, THEN END OF
                       68  * CATALOG.  IF NEITHER, THEN CHECK FILETYPE
                       69  * BYTE TO SEE IF IT'S A BINARY FILE.  IF SO,
                       70  * THEN PRINT FILENAME AND FIND T/S LIST AND
                       71  * FIRST FILE SECTOR.  THEN LIST START ADDRESS
                       72  * AND LENGTH.  THEN INCREMENT TRACK BYTE
                       73  * BY 35 ($23) TO GET TRACK OF NEXT FILE.
                       74  *
02F1: BD 00 96         75  GETFILE  LDA    BUFF1,X
02F4: D0 03            76           BNE    MORE1
02F6: 4C A0 03         77           JMP    DONE
02F9: C9 FF            78  MORE1    CMP    #$FF
02FB: F0 0E            79           BEQ    ONWARD
02FD: 8D BE 03         80           STA    TRACK           ;GET TRACK OF T/S LIST.
0300: B9 00 96         81           LDA    BUFF1,Y
0303: C9 04            82           CMP    #$04            ;UNLOCKED BINARY FILE?
0305: F0 28            83           BEQ    BINFL
0307: C9 84            84           CMP    #$84            ;LOCKED BINARY FILE?
0309: F0 24            85           BEQ    BINFL
030B: 18               86  ONWARD   CLC                    ;NO, ON TO NEXT FILE
030C: 8A               87           TXA
030D: 69 23            88           ADC    #$23            ;$23=35.
030F: F0 09            89           BEQ    NEWCAT          ;LAST FILE ENTRY
                       90  *                               IN THIS CATALOG SECTOR?
0311: AA               91           TAX
0312: 18               92           CLC
0313: 98               93           TYA
0314: 69 23            94           ADC    #$23
0316: A8               95           TAY
0317: 4C F1 02         96           JMP    GETFILE
                       97  *
031A: A5 EC            98  NEWCAT   LDA    CATT            ;LOAD NEXT SECTOR OF
031C: 8D BE 03         99           STA    TRACK           ;CATALOG INTO BUFF1
031F: A5 ED           100           LDA    CATS
0321: 8D BF 03        101           STA    SECTOR
0324: A9 96           102           LDA    #$96
0326: 8D C3 03        103           STA    BUFFER
0329: 20 B2 03        104           JSR    READISK
032C: 4C E0 02        105           JMP    START1
                      106  *
032F: 8A              107  BINFL    TXA                    ;X HAS TRACK# OFFSET
0330: 48              108           PHA
0331: 98              109           TYA                    ;Y HAS FILETYPE OFFSET
0332: 48              110           PHA
```

```
0333: 18            111              CLC
0334: 69 19         112              ADC      #$19        ;PRINT ONLY FIRST 25 CHARS
0336: 85 EE         113              STA      COUNTER     ;OF FILENAME TO KEEP
0338: C8            114     LOOP2    INY                  ;LINE < 40 COLUMNS.
0339: B9 00 96      115              LDA      BUFF1,Y
033C: 20 ED FD      116              JSR      COUT
033F: C4 EE         117              CPY      COUNTER
0341: F0 03         118              BEQ      MORE3
0343: 4C 38 03      119              JMP      LOOP2
0346: E8            120     MORE3    INX
0347: BD 00 96      121              LDA      BUFF1,X     ;GET SECTOR OF T/S LIST.
034A: 8D BF 03      122              STA      SECTOR
034D: A9 97         123              LDA      #$97
034F: 8D C3 03      124              STA      BUFFER
0352: 20 B2 03      125              JSR      READISK
                    126     * T/S LIST NOW IN BUFF2
0355: A0 0C         127              LDY      #$0C        ;T/S LIST STARTS AT $0C
0357: B9 00 97      128              LDA      BUFF2,Y
035A: 8D BE 03      129              STA      TRACK       ;GET TRACK AND SECTOR #'S
035D: C8            130              INY                  ;OF FIRST SECTOR OF ACTUAL FILE.
035E: B9 00 97      131              LDA      BUFF2,Y
0361: 8D BF 03      132              STA      SECTOR
0364: A9 97         133              LDA      #$97        ;FIRST SECTOR OF BINARY
0366: 8D C3 03      134              STA      BUFFER      ;FILE IS FINALLY IN BUFF2!
0369: 20 B2 03      135              JSR      READISK     ;NOW GET START AND LENGTH.
036C: A9 C1         136              LDA      #$C1        ;LOAD A-REG WITH "A"
036E: 20 ED FD      137              JSR      COUT        ;AND PRINT IT
0371: A0 00         138              LDY      #0
0373: 20 AC 03      139              JSR      DLRSN
0376: BE 00 97      140              LDX      BUFF2,Y     ;PRINT STARTING ADDRESS AS
0379: C8            141              INY                  ;TWO-BYTE FORMAT.
037A: B9 00 97      142              LDA      BUFF2,Y
037D: 20 41 F9      143              JSR      PRNTAX
0380: 20 48 F9      144              JSR      PRBLNK
0383: A9 CC         145              LDA      #$CC        ;LOAD A-REG WITH "L"
0385: 20 ED FD      146              JSR      COUT        ;AND PRINT IT
0388: 20 AC 03      147              JSR      DLRSN
038B: C8            148              INY
038C: BE 00 97      149              LDX      BUFF2,Y     ;PRINT FILE LENGTH AS
038F: C8            150              INY                  ;STANDARD TWO-BYTE FORMAT.
0390: B9 00 97      151              LDA      BUFF2,Y
0393: 20 41 F9      152              JSR      PRNTAX
0396: 20 A3 03      153              JSR      CR
0399: 68            154              PLA                  ;RETRIEVE X AND Y VALUES
039A: A8            155              TAY                  ;FROM THE STACK AND
039B: 68            156              PLA                  ;GO TO NEXT CATALOG ENTRY.
039C: AA            157              TAX
039D: 4C 0B 03      158              JMP      ONWARD
                    159     *
03A0: 4C D0 03      160     DONE     JMP      DOSWRM
                    161     *
03A3: A9 00         162     CR       LDA      #0          ;RETURN CURSOR TO LEFT
03A5: 8D 78 05      163              STA      CH          ;AND DO CARRIAGE RETURN
03A8: 20 66 FC      164              JSR      CROUT
03AB: 60            165              RTS
03AC: A9 A4         166     DLRSN    LDA      #$A4        ;LOAD A-REG WITH "$"
03AE: 20 ED FD      167              JSR      COUT        ;AND PRINT IT.
03B1: 60            168              RTS
03B2: A9 03         169     READISK  LDA      #>IOB2
03B4: A0 BA         170              LDY      #IOB2
03B6: 20 D9 03      171              JSR      RWTS
03B9: 60            172              RTS
                    173     *
                    174     * THE IOB IS SET UP FOR APPLE DISK ][ DRIVES.
                    175     * WILL ONLY WORK FOR DISK IN DRIVE 1, UNLESS
                    176     * IOB IS SET TO DRIVE 2 (SEE DOS MANUAL).
                    177     *
03BA: 01 60 01
03BD: 00            178     IOB2     HEX      01,60,01,00
03BE: 00            179     TRACK    HEX      00
03BF: 00            180     SECTOR   HEX      00
03C0: 00            181     DCTL     HEX      00
03C1: 00            182     DCTH     HEX      00
03C2: 00            183              HEX      00
03C3: 00            184     BUFFER   HEX      00
03C4: 00 00 01
03C7: 00 00 60
03CA: 01            185              HEX      00,00,01,00,00,60,01
03CB: 00 01 EF
03CE: D8            186     DCT      HEX      00,01,EF,D8
--END ASSEMBLY--
```

APPLE //E TO NBI TELECOMMUNICATION

by Lynn R. Trusal, Ph.D.

If a survey were taken of Apple owners as to the number one use of their computer, it would probably be word processing. This is evidenced by the fact that five of the top seven programs for the Apple][are word processing programs (Softalk, August issue). Today's sophisticated word processing programs, such as "AppleWorks" (Apple Computer, Inc.), permit the use of such features as superscript, subscript, underlining and bold face. For many users, a personal letter printed out on a dot matrix printer will suffice, but even with the ability to access these features by a dot matrix printer, many professional needs require letter quality printouts.

Fortunately, many of us work for employers who have purchased professional word processor systems from such companies as NBI, IBM (perish the thought), Wang and Xerox. During the course of my work, I write scientific manuscripts for submission to professional journals. Although many journals will accept dot matrix printouts, others will not; and access to Greek or special symbols is not always possible. In order to address this issue, I will describe a method for the transfer of Apple][text files to an NBI professional word processing system. A similar procedure should be adaptable to almost any commercial word processor.

The word processing program that I have found the best and that I use extensively for manuscript preparation is the ProDOS-based "Appleworks". I can see why it is the number one selling program for the Apple //e or //c, and I highly recommend it. For our telecommunication needs, we use "ASCII Express the Professional" (United Software), which requires DOS 3.3 text files for telecommunication. If your particular word processing program saves DOS 3.3 text files, you're in business. If not, you must first convert the files to the required format. United Software assures me that an "ASCII Express" version that handles ProDOS text files is now available for the price of $50 to owners of previous versions. Also, keep in mind that "ASCII Express" contains utilities for conversion of CP/M and Pascal files to DOS 3.3, so theoretically it should be possible to use CP/M and Pascal-based word processing programs, in addition to DOS-based ones.

"AppleWorks" files may be printed to disk (option 4, "Print Menu") as ASCII text files. If you do not have the new version of "ASCII Express", the final preparation step involves conversion of these text files to DOS 3.3. The text file conversion may be done easily by using the "DOS-ProDOS conversion option" on the ProDOS program ("Convert Menu"). Once DOS 3.3 text files have been saved to disk, transfer can begin.

Our professional NBI word processing system utilizes a 120 Mbyte hard disk called an integrated control unit (ICU), which is in turn hardwired to workstations located throughout the Research Institute. The ICU is connected to several Hayes Smart Modems (Hayes Microcomputer Products, Inc.), each of which has a separate phone line. On the Apple end, we utilize an acoustic-coupled modem because it is more flexible and because dedicated phone lines are a luxury. Note that unless a dedicated data phone line exists, acoustic modems may be limited to 300 baud transmission due to inherent noise in commercial phone lines. The NBI workstation is then set up for communication by logging on to a "communication directory" that has been tailored for ASCII text files or a specific word

processing program. Your particular professional word processor representative can assist you if reference manuals are not clear on this point. It is worthy to note that when "AppleWorks" files are saved as ASCII text files, special printer control codes are stripped but hard and soft carriage returns are maintained. The particular professional word processor used in your office will dictate how much tailoring can be done to preserve these special codes. On the Apple end of the line, I use an Apple Super Serial Card that was previously set to the following parameters: 300 baud, 8 data bits, and 1 stop bit. "ASCII Express" allows many of these parameters to be reset from the keyboard should a change become necessary.

Once the NBI is in the "conversational" mode, the "micro" operator establishes a carrier by dialing the ICU modem and connecting the phone to the acoustic modem which has been set externally at 300 baud. By using the "conversational" mode, it is possible to determine that a successful communication link has been established. Keyboard input on the Apple shows up on the NBI screen and vice versa. The NBI workstation operator then switches to the "receive" mode and the Apple][operator begins the file transfer by evoking the "send a file" command of the telecommunication program ("ASCII Express Pro Menu 1"). During the transfer, it may appear that some of the text lines are being overwritten on the NBI screen, but once the transfer is completed, it can be seen that the transferred files are intact. The NBI operator may then reformat the document by entering NBI control codes for the ones that were removed when the files were originally saved. Figure 1 illustrates the overall procedure involved in the Apple][to NBI transfer of ASCII text files.

I have not attempted to give a step-by-step procedure, since that would be not only laborious but inapplicable to other word processing programs for the Apple][, or to other professional word processors. With the will to succeed, right motivation and some trial and error, you should be successful with your file transfers. Good luck!

Figure 1.

CONVERSION OF A MANUSCRIPT WORD PROCESSED ON AN APPLE TO AN NBI PROFESSIONAL WORD PROCESSOR

AppleWorks ProDOS › Text File

Convert to DOS 3.3 or use ASCII Express ProDOS version

| ASCII Express | Letter Quality Printer | NBI Work Station |
| APPLE][| Apple Modem | ICU Modem | NBI ICU |

APPLEWORKS UTILITY
by Walt Mossberg

APPLEWORKS INTERFACE CARD CONFIGURATION UTILITY

This Applesoft program, issued by Apple Computer, Inc. on disk and made free to dealers, is intended to perform a one-time-only modification of the AppleWorks Startup Disk so that the program will function smoothly with nonstandard parallel cards, including Grappler, and with non-listed printers. I downloaded this program listing from the database of the Micronet Apple Users Group (MAUG), on CompuServe.

To use the utility program:

1. Boot the ProDOS Users Disk and enter Applesoft Basic from Main Menu.

2. Type in the attached program listing.

3. SAVE the Applesoft program now in memory, using a name of your choice.

4. Make a COPY of the AppleWorks Startup disk. DO NOT modify the original!

5. From ProDOS, RUN the program saved in step 3. Follow the directions on the screen to configure the BACKUP of the Startup disk.

6. Boot the reconfigured Startup disk. You may have to reenter printer specs if you changed these on your earlier backup.

That's it! Remember, ONLY modify a backup. Also, the configuring program has ProDOS commands embedded within and MUST be run from ProDOS! If you don't have the ProDOS user's disk, see if your dealer will let you use his.

NOTE: To fully emulate the Apple Configuration disk, you should have ProDOS on the disk, and Basic.System. Then the Applesoft file should be renamed Startup.

NOTE TO OKIDATA PRINTER OWNERS: Apple has provided these minor changes to the program listing in order to make it modify Appleworks for use with the Okidata 92 printer. It is said to enable the printer to use all its functions with AppleWorks, and to halt the printing of stray control codes like "80N". Just type in these lines:

```
380 FOR I = 11556 TO 11561
450 DATA 4,137,184,176,206,225
```

This procedure comes from the Apple office in Charlotte, NC. The phone there is (704) 527-6170.

```
10  TEXT : HOME
20  D$ = CHR$ (4)
30  PRINT D$;"PR#3": PRINT
40  PRINT "AppleWorks Interface Card Configuration Utility";
50  POKE 36,69: PRINT "Version A01";
60  PRINT "Copyright Apple Computer, Inc 1984";
70  POKE 36,67: PRINT "April 5, 1984";
80  VTAB 6: PRINT "This utility allows you to configure a copy of the AppleWorks Startup disk to"
90  PRINT "work with several parallel interface cards.  Important notes:"
100 PRINT : PRINT "- DO NOT USE YOUR ORIGINAL APPLEWORKS STARTUP DISK.  Use this utility only on"
110 PRINT " a backup copy (an exact copy) of the AppleWorks Startup Disk."
120 PRINT : PRINT "- DO NOT USE THIS UTILITY IF YOU HAVE AN APPLE II PARALLEL INTERFACE  CARD"
130 PRINT " (A2B0021) OR FULLY COMPATIBLE CARD.  AppleWorks prints properly on these"
140 PRINT " cards without modification.  A disk that has been configured with this"
150 PRINT " utility will no longer run correctly with the Apple II Parallel Interface"
160 PRINT " Card."
170 PRINT : PRINT "- Apple cannot guarantee that this utility will make AppleWorks compatible"
180 PRINT " with any given parallel interface card.  The only reliable way to check"
190 PRINT " compatibility is to test your configured AppleWorks with the card in"
200 PRINT " question.  For additional information, see your authorized Apple dealer."
210 VTAB 4: FOR I = 1 TO 80: PRINT "_";: NEXT I
220 VTAB 22: FOR I = 1 TO 80: PRINT "_";: NEXT I: PRINT
230 VTAB 24: PRINT "Press RETURN to continue or ESCAPE to stop. ";: INVERSE : PRINT " ";: NORMAL : PRINT  CHR$ (8);
240 IF  PEEK (49152) < 128 THEN 240
250 IF  PEEK (49152) = 155 THEN  POKE 49168,0: TEXT : HOME : END
260 POKE 49168,0
270 POKE 34,4: POKE 35,21: HOME
280 VTAB 11: PRINT "Remove the AppleWorks Interface Configuration Utility": PRINT "from the disk drive."
290 VTAB 14: PRINT "Place your COPY of the AppleWorks STARTUP disk in Drive 1."
300 VTAB 24: PRINT "Press RETURN to continue or ESCAPE to stop. ";: INVERSE : PRINT " ";: NORMAL : PRINT  CHR$ (8);
310 IF  PEEK (49152) < 128 THEN 310
320 IF  PEEK (49152) = 155 THEN  POKE 49168,0: TEXT : HOME : END
330 HOME : VTAB 12: HTAB 26: PRINT "Configuring AppleWorks Disk"
340 POKE 49168,0
350 ONERR  GOTO 470
360 PRINT D$;"BLOAD /APPLEWORKS/APLWORKS.SYSTEM,A$2000,T$FF"
```

```
370  RESTORE
380  FOR I = 11556 TO 11560
390  READ P
400  POKE I,P
410  NEXT I
420  POKE 12902,71
430  PRINT D$;"BSAVE /APPLEWORKS/APLWORKS.SYSTEM,A$2000,L6499,T$FF"
440  HOME : VTAB 12: HTAB 33: PRINT "Configuration Successful": VTAB 18: PRINT "To configure another copy, replace the disk in Drive 1"
445  PRINT "with another copy of the AppleWorks Startup Disk.": GOTO 300
450  DATA 3,137,176,206,0: REM   (String length)(CTRL-I)(ZERO)(N)(null)
460  END
470  POKE 216,0
480  IF  PEEK (222) = 8 THEN 510: REM  I/O ERROR
490  IF  PEEK (222) = 6 THEN 530: REM  COULD NOT FIND FILE
500  IF  PEEK (222) = 4 THEN 550: REM  DISK IS WRITE PROTECTED
510  HOME : VTAB 10: PRINT "Cannot read your Disk": PRINT : PRINT "Make sure that you have a copy of AppleWorks in the drive,"
515  PRINT "and the drive door is closed"
520  GOTO 560
530  HOME : VTAB 10: PRINT "Cannot find the disk or file that requires changing."
531  PRINT "Make sure you have a copy of the AppleWorks STARTUP"
532  PRINT "disk in the drive and the drive door is closed."
540  GOTO 560
550  HOME : VTAB 10: PRINT "Your disk is Write Protected.": PRINT : PRINT "Remove the Write Protect Tab from your disk, place it back"
555  PRINT "in the drive, and close the door"
560  POKE 34,23: HOME : POKE 34,4: VTAB 24: PRINT "Press RETURN to continue ";: GET 224: PRINT : HOME : GOTO 290
```

DEC.   HEX

$11556 = 2D24$ $216 = D8$

$137 = 89$ $12902 = 3266$

$176 = B0$ $71 = 47$

$206 = CE$

$184 = B8$

$225 = E1$

LOST

Lost forever are the words of wisdom which could have occupied this and other white space on: applications for, hints and solutions to problems with the LISA, //c and /// computers.

REWARD

Finders will be rewarded with contributions in kind.

SCREENWRITER HOTLINE
by Peter Combes & Tom Vier

Just for a change, Tom Vier will talk about Screen-Writer, and I will recklessly declaim about Apple Writer (Tom Warrick, stand back).

ScreenWriter and Apple Writer both do not bother to reset the margins on the printer, but calculate how many spaces they would have to print in order to look as if they were printing with the required margins. The problem comes with underlining. Both processors are capable of setting an underscore token at the beginning of an underscore section and setting the same token at the end of that section. ScreenWriter is intelligent enough to realize that the "extra" spaces it is using for margins should not be under-scored. A well known "bug" in Apple Writer is that it tends to underscore all over the margins as well, with ridiculous results.

Working recently with Apple Writer, a //e, and an Okidata printer, I struggled through to the following solution:

The left margin was at 10, the right margin at 70.

Reset the Apple Writer margins by the embedded com-mands .lm0<RETURN>, .rm60 <RETURN>.

Set the printer left hand margin by <CONTROL>V%120 <RETURN>. This sets the printer left margin to one inch from the left -- 10 characters at 10 cpi.

Start underscore with <CONTROL>V <ESCAPE>C and end it with <CONTROL>V <ESCAPE>D.

This works, but who said computers saved people time?

Now, over to Tom Vier, who reports two (more!) Screen-Writer][bugs:

If you have been having trouble with the speed or scrambling of output to the screen and DOS after using the 70 column display, try this:

```
10  PRINT  CHR$ (4)"BLOAD EDITOR PART1 .OBJO"
20  REM    FIX SPEED BUG
30  IF PEEK (5522) = 241 THEN  POKE 5522,251:I = I + 1
40  IF PEEK (5542) = 241 THEN  POKE 5542,251:I = I + 1
50  REM    FIX SCRAMBLE BUG
60  IF PEEK (5514) = 243 THEN  POKE 5514,206:I = I + 1
70  IF PEEK (5531) = 243 THEN  POKE 5531,206:I = I + 1
80  IF I < > 4 THEN  END
90  PRINT CHR$ (4)"BSAVE EDITOR
    PART1.OBJO,A3072,L1400"
```

ScreenWriter was using the Applesoft SPEED variable location in the 70 column display routine and leaving it with a v e r y s l o w value. Another Applesoft location it was clobbering was the ORMASK, used for all output. If invalid, you get text in, garbage out! On the other hand, you can change it with POKE 243,32 to convert output to lower case. Restore it with POKE 243,0.

Saving Unformatted ScreenWriter][Files to Hard Disk:

Once you have successfully mounted ScreenWriter on your hard disk and have selected your floppy drive(s) in CUSTOMIZE for your text disks, the following patch allows you to save unformatted files to your hard disk. Create a text file with these commands and EXEC them on a backup copy only!

```
BLOAD EDITOR.OBJO
CALL-151
5CE1:00
5E32:A4 0D DO 08 8C FF
5E38:1F C9 8D D0 3D C8 84 0C
5E40:20 7D 16 20 3E 17 90 03
5E48:20 DE 16 A9 8D 20 7F 5E
5E50:AD FD 1F FO 05 A9 8A 20
5E58:7F 5E EE F8 27 AD F8 27
5E60:CD FE 1F D0 11 A9 8C 20
5E68:7F 5E A9 8D 20 7F 5E A0
5E70:00 8C F8 27 FO 04 C6 0C
5E78:DO C6
3D0G
UNLOCK EDITOR.OBJO
BSAVE EDITOR.OBJO,A$4000,L$2E00
```

To save an unformatted file once this patch has been installed, use the List command as follows:

```
<CTRL>D OPEN filename<CR>
<CTRL>D WRITE filename<CR>
Lx,0,0,$9EBD<CR>
<CTRL>D CLOSE<CR>
```

The List command now allows zero spacing, but will remove the left margin after the first line. Thus, to save a file entirely verbatim, including embedded commands, make the first line a <CR> only. The line length (x), is ignored. To reset the left margin, you must reboot or BRUN START again.

As you may be aware, RUNOFF can use files directly from hard disk. However, the Editor cannot load files except from diskette.

```
1   •••••••••••••••••••••••••••••••
2   •                             •
3   •      LIST COMMAND PATCH      •
4   •                             •
5   •     for Screenwriter ][     •
6   •                             •
7   •   by Tom Vier  Reston,Va.   •
8   •                             •
9   •          1/26/84            •
10  •                             •
11  •••••••••••••••••••••••••••••••

13           ORG   $5E32

15  SPCCOUNT EQU   $0C      ; Line spacing counter
16  SPACING  EQU   $00      ; Parsed spacing
17  LFFLAG   EQU   $1FFD    ; Line feed flag
18  PGLENGTH EQU   $1FFE    ; No. of lines per page
19  LMARGIN  EQU   $1FFF    ; Left margin
20  LINECNT  EQU   $27F8    ; Current line
21  PRINT    EQU   $5E7F    ; Character output routine

5E32: A4 0D    23           LDY   SPACING       .
5E34: DO 08    24           BNE   SKIP
5E36: 8C FF 1F 25           STY   LMARGIN       ; Zero left margin
5E39: C9 8D    26           CMP   #$8D          ; Carriage return from text?
5E3B: DO 3D    27           BNE   RESUME        ; if not
5E3D: C8       28           INY                 ; if so, do it once
5E3E: 84 0C    29  SKIP     STY   SPCCOUNT
5E40: 20 7D 16 30  LOOP     JSR   $167D
5E43: 20 3E 17 31           JSR   $173E
5E46: 90 03    32           BCC   EOL
5E48: 20 DE 16 33           JSR   $16DE
5E4B: A9 8D    34  EOL      LDA   #$8D          ; Carriage return
5E4D: 20 7F 5E 35           JSR   PRINT
5E50: AD FD 1F 36           LDA   LFFLAG
5E53: FO 05    37           BEQ   NEXTLINE
5E55: A9 8A    38           LDA   #$8A          ; Line feed
5E57: 20 7F 5E 39           JSR   PRINT
5E5A: EE F8 27 40  NEXTLINE INC   LINECNT
5E5D: AD F8 27 41           LDA   LINECNT
5E60: CD FE 1F 42           CMP   PGLENGTH
5E63: DO 11    43           BNE   NEXTLOOP
5E65: A9 8C    44           LDA   #$8C          ; Form feed
5E67: 20 7F 5E 45           JSR   PRINT
5E6A: A9 8D    46           LDA   #$8D          ; Carriage return
```

IDLE THOUGHTS
by Bob Cosgrove

TO ERR IS HUMAN....Each of us makes silly mistakes from time to time. One of the most common errors made (especially by beginners) is the erasing of a program that has been typed but not saved to disk. (Usually this is performed by having typed NEW and then pressing RETURN.) In the December 1982 issue of the WAP Journal, there was a handy little program in the 'Q & A' column that magically brings your erased program back to life. We have added a few thousand new WAP members since its initial appearance, so I thought it might be a good idea to show it again. I have listed it without all of the other stuff that we non-technical folk don't have to be concerned with.

Enter the System Monitor by typing CALL -151 and pressing RETURN. You should find an asterisk (*) as a prompt instead of the usual Applesoft Basic prompt (]). Type the program exactly as shown, being sure that there are no spaces between the colons and the first pairs of hexadecimal digits in each line. After that, be sure there is a space between every pair. Press RETURN at the end of each line.

```
0300:A2 04 E8 BD 00 08 D0 FA E8
0309:8E 01 08 A9 08 8D 02 08
0311:86 3C 85 3D A5 73 85 3E A5 74
031B:85 3F A0 FD 20 BA FC B0 20
0324:B1 3C D0 F5 C8 30 F9 20 BA FC
032E:A5 3C 85 69 85 6B 85 6D 85 AF
0338:E6 3D A5 3D 85 6A 85 6C 85 6E
0342:85 B0 60
```

When you've finished typing, press RETURN. You can go back into Basic by pressing CONTROL-C and RETURN. You should now see the familiar ']' prompt. Save this program to disk by typing BSAVE BRING BACK MY PROGRAM, A$300,L$40 and pressing RETURN.

Examine the value of this program by typing NEW, pressing RETURN, then adding a little two-liner such as this:

```
10  PRINT "Will this program return?"
20  PRINT "Yes it will!"
```

Now hit the RETURN key, and type NEW after the next prompt. (If you type LIST, the only thing you'll find on your screen is another prompt. It thus appears that your program is gone forever. But hold on....)

Type BRUN BRING BACK MY PROGRAM. Hit RETURN. Now type LIST, and you should see your original program on the screen, ready for modification or saving to disk!

AFRAID TO USE MAIL ORDER??? Calm your fears with two companies I've found that provide excellent service: Rising Sun Software in California and Silicon Express in Columbus, Ohio....Rising Sun does not have a holding period for personal checks, and it usually takes less than ten days to get software delivered to your door. Silicon Express declares a two-week waiting period, but I have received software in less than two weeks....Silicon includes in its shipping a brochure that includes hundreds of Apple programs... Both companies advertise in most Apple-based publications.

I WEEP, BUT I ALSO WANT MY MONEY BACK....It's a shame that Softalk has ceased publication. The magazine's cover could have been more issue-related at times, but its 'departments' section was second to none, and

there was always something of interest for every Apple owner....In its last published Statement of Ownership (November 1983, page 20), it was reported that 153,774 copies of its previous issue were printed, but only 61,926 were actually sold on the street or covered by paid subscription. Thus 89,435 copies were sent free to new Apple owners and other parties. Was it no wonder that on page 303 of that same issue the Softalk folks were "thankful inCider and A+ haven't put us out of business yet"? (And I can distinctly remember laughing at their self-deprecating humor, even going as far as reading this 'hilarious' statement to my friends!)....There are a few magazines in the world that generate so much advertising revenue that subscription income is all gravy for its owners, but I'm sure this wasn't the case here. Publisher Al Tommervik will undoubtedly be telling his story in the near future....Let's hope that the authors of features such as DOStalk, Follow the Floating Point, Mind Your Business, and Beginner's Corner will be appearing regularly in other Apple-based publications....Now, about that subscription refund....   ⑤

```
5E6C: 20 7F 5E   47          JSR  PRINT
5E6F: A0 00      48          LDY  #$00
5E71: 8C F8 27   49          STY  LINECNT
5E74: F0 04      50          BEQ  RESUME    ; always
5E76: C6 0C      51  NEXTLOOP DEC  SPCCOUNT
5E78: D0 C6      52          BNE  LOOP
5E7A: 84 02      53  RESUME  STY  $02
5E7C: 4C 20 5E   54          JMP  $5E20
```

--End assembly, 77 bytes, Errors: 0   ⑤

DAISYWHEELING: The Lure of Code
by Arsen Darnay

(Ed. Note: Reprinted from the September 1984 issue of mini'app'les.)

The overwhelming majority of those who own computers are quite content to let programmers program, while users use.

A year of two ago I still saw ads - I remember one on television for the Texas Instruments machine - that held out the promise of programming to the uniniti- ated. That was bound to be both a passing and a peripheral theme. More and more the theme is ease. You don't have to be a mechanic to drive a car. That sort of thing. It's almost as if Ignorance is Wisdom, Incompetence is Skill. But then the year is still 1984.

No, I am not rattling the cage of Reality or harking back to some Golden Age. It's the way of technology in a mass culture. A few are creative. The rest consume. And the companies make all the money. Well, they have their reward. My point today is quite another one: the advent of the micros has provided some of us entry into a marvelous world of abstrac- tion with strange but real rewards. That's because we came in at the beginning of things.

If you look at early airplanes or at renditions of prehistoric birds, you will notice that all the struts and wires are still showing. Those early birds were not more bony than their descendants. The bones merely showed. The sailing ships of old displayed their masts and rigging. Deeply hidden turbines propel the modern ships.

I've often thought that as things evolve, their nerves and skeletons move inward and their skins grow sleek and smooth. It's happening right now in small com- puters. The //c's box is already sealed. You still have to feed it diskettes, but that will be the next piece of the machinery to go. Computers will soon come with built in sotware, the code burned in ROM. You'll get a built-in spreadsheet, word processor, filing system, graphics package. The computer will have become a commodity. And the wild people who love today's Model T will move on to where the action will be then.

The action for me is still inside the small computers, in their structure, in the code that drives them. They're fine for work and play and all that sort of things, but you get used to that. But programming is a joy. It's a new and strange species of art.

The greatest contribution of those guys in Cupertino - the early guys, not the corporate giant of today - lies in opening the door of that odd world to hundreds of thousands of us. They did it, perhaps, of necessi- ty. There was no software in the early days. The market was the hobbyist. They produced machines for that market with a program in ROM to let you look into the memory. They drew maps of the territory. Do your own programming. The world of computers became acces- sible to those who had the patience and the stamina.

Of necessity or not, they gave us what Herman Hesse merely wrote about - The Glass Bead Game, that game of games which combines high art with science and philos- ophy.

So many of us came, and after looking at the thing for a while, we shook our heads in puzzlement. How can a device that turns switches on or off - yes/no, zero/ one, yin/yang - produce such fascination? Could the answer just be speed. No. It's more than that. The computer's utter simplicity, its mass of tiny switches, its near-light speed of action - all this together creates a wondrous imitation of life. And that, I suppose, is what attracts the artist in all of us. Writing a book or painting a picture can never satisfy its creator quite like a program that _runs_.

Life. In a newsletter on microbiology last summer, a man laid out in clearest fashion the great similarity between genetic and computer code. He traced identi- cal structures in both, item by item. I'd add that music, in the arts, also fits into this category. As in computers and inside the gene, the musical code has little resemblance to that which is produced. All three result in movement, vision, sound. All three are simple and unambiguously "mathematical" in essence, complex in effect.

Once I'd caught on to the strange nature of this game - the right-brain, left-brain duality which it unites, logician and child - I knew that Wozniak and friends had stolen divine fire from dull corporate gods to enrich our souls.

Excited like a child, I used to show printouts of my early programs to anyone who'd look - and talked about the great kinship between those lines of type on fan- fold paper and the musical score. The programs didn't do much, but making them had been reward enough.

And that gives a clue to the lure of code. To write a program is to enter a world of purity - the pure com- plexity of formal systems, quite unlike the Fallen World of Man. Do the right thing and you'll get pre- dictable results. Make an error and you bomb. No long lapse between the deed, its consequence. Run the program and find out. Knowledge is rewarded, ignor- ance and carelessness punished - every time. Each step is simplicity itself, but when you add step to step in a rising spiral of abstract architecture, the whole transcends the parts. When it runs, it's para- dise. When it crashes, utter hell. In between is the joy of work. The sleazy realms of Maybe and Perhaps, the grime of Ambiguity, resume their sway when you kill the engine and face the rest of the world.

It's an experience - and _what_ you're making is far less important than the act.

Since those days I've found myself progressing ever deeper - from simple BASIC to BASIC with lots of POKEs to 6502 machine code and now on to the Motorola 68000 - always seeking greater speed so that I may have "life" more abundantly.

Curiously enough, the deeper you go, the greater is the divergence between the visible result and the means of creating it. BASIC still resembles English, but while it is more comprehensible, it is slow and keeps you at a distance from the core of things. It's ritualistic, like formal religion. A priesthood stands between you and the heart of the mystery.

High skill and grand effects - million dollar speed and all that it brings, efficient use of memory, utter control - require mastery of machine code. The highest wizardry rests on mole-like grubbing on the

contd.

level of electronic genes. The joy, oddly, comes less from the superior results and more from an almost intangible "feel" - a perception of routines smartly clicking through stacatto steps to clean conclusions.

Strange, very strange. Those of you who've visited this realm will understand. As for the rest of you, you're probably wise to boot up VisiCalc and not worry too much about the mysticism that produced it.

Views and Reviews contd from pg 50

or joystick control in the former, and partly to the fact that I have been playing Wolfenstein over a year longer. Also, the graphics were somewhat more difficult to follow, mainly because so much more is included in each screen. Nonetheless, Aztec provides hours of entertainment for the nimble-fingered. It is challenging, and the difficulty level can be increased to an unbelievable level. I feel like a kid again.

** Footnote: My thanks to Steve Grimm, for his input in these reviews, as well as his help in the destruction of the axis powers.

Software City

OUR APPROACH

- All Products Always Discounted
- Reference Literature & Guides For Browsing
- Demo Machines
- Special Orders Our Speciality
- Gift Certificates Available
- Rainchecks For Out-Of-Stock Specials
- Comfortable Showroom Environment
- Special Corporate, Government, Educational & User Group Accounts

SOFTWARE

- Business
- Scientific
- Education
- Recreation
- Programming
- Home & Personal
- Specialized

HARDWARE

- Monitors
- Printers
- Drives
- Interfaces
- Modems
- Expansion Boards
- Cables

LITERATURE

- Books
- Magazines
- Free Product Information

ACCESSORIES & SUPPLIES

- Media & Media Storage
- PC Protection & Care Products
- Paper & Custom Forms

STORE HOURS

Mon-Fri 10-6
Thurs 10-8
Sat 10-5

468-1001

11621 Nebel Street (Extended)
Rockville, Maryland 20852

ROCKVILLE
Montrose Rd Randolph Rd
Rt. 270 to Frederick
Marinelli Rd
Nebel St
Rockville Pike
Software City
POTOMAC
Montgomery Mall
Nicholson Lane
White Flint Mall
I-495
Beltway
To DC   CHEVY CHASE
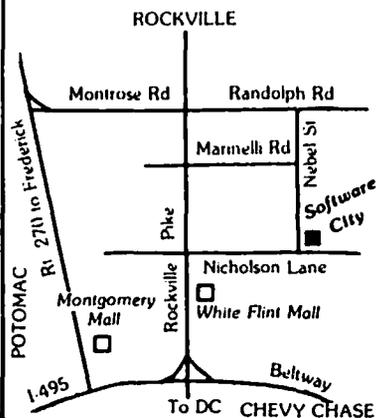
Complete coupon to be placed on Software City Special Mailing List.

Name _____

Address _____

_____

Phone _____

Type PC at Work _____

Type PC at Home _____

SIG MAC NEWS
by Ellen L. Bouwkamp
& Steve Hunt

QUESTION AND ANSWER PERIOD:

On October 12, we had a lively one-hour question and answer period addressing a wide range of issues including questions on applications, use of hard disk drives, modem characteristics, and many other current topics. The question and answer session closed with Steve Hunt's admonition to buy a surge protector for your Mac. During a previous SigMac meeting his machine was hit with more power than it liked, and Steve had to replace the main logic board (motherboard) - at a cost of over $100! Save yourself money and agony - buy a good surge protector. (Ed. Note: Not enough! See "Wednesday's Child" elsewhere in this issue.) These range in price and have different features; even an investment of $50 is still a savings over the charge for new Mac innards.

BUSINESS MEETING:

Tom Warrick opened our formal meeting with a notice that the Jones Bridge Road entrances will be closed on Saturdays. In the future, we all need to use the main entrances on Wisconsin Avenue. To those who gave up, we sincerely apologize for the confusion or total frustration caused by this change of policy. Hope to see you next meeting.

The next SigMac programming meeting (November 1) will be held at the USUHS campus. However, the programming meetings will move beginning with the December 6th meeting to Our Lady of Lourdes School, 7500 Pearl Street, in Bethesda.

SigMac elections were held swiftly and with little discussion. A new post, Vice Chairman for Programs, was approved. The person in this position is responsible for planning monthly meetings and coordinating room and audiovisual requirements. By acclamation, the following officers will serve SigMac members until October 31, 1985:

        Chairman - Steve Hunt
        Vice Chairman for Programs - Don Landing
        Correspondent - Ellen Bouwkamp
        Librarian - Tony Anderson (and any others that
                Tony may appoint)

We want to thank Tom Warrick for his generous gift of time and wisdom in getting SigMac off the ground. Kudos also to Ron Grognet for sharing the Librarian position in our first months of operation.

LATEST INFORMATION:

Bruce Field warns of a bug in 128K Mac firmware that affects development of telecommunication programs (it has NO impact on USE of a program like MacTerminal. If you desire 7 data bits, even parity, when you use the serial controller, you must specify 6 data bits, even parity. Similarly, 7 bits, odd is entered as 6 bits, odd. This bug affects both serial ports. Fortunately for most of us, if you need a setting of 8 data bits, no parity, that is exactly what you tell your Mac.

Expect an announcement of the Apple laser printer, possibly at the January Apple stockholders meeting. The printer uses 64 of the 256K chips, and speculation about the price runs from $3000 to $5000. Tom had samples of the print quality, which is excellent.

C LANGUAGES OUT:

Tom reported that there are at least four C Compilers for the Mac now available. (Reference is made in parentheses to pages where these products are referenced in the November 1984 issue of MacWorld.)

● Manx Software Systems, Aztex C68K-c, $499.00
    (p. 23 and 90)

● Megamax, Inc., Megamax C Compiler, $299.95
    (p. 91 and 145)

● Consulair Corp., Mac C/Mac C Toolkit, $425.00
    (p. 89)

● Softworks, C (price not known)

The Softworks C does not presently operate on the Fat Mac; however, the Consulair product does. One of the next Thursday programmers meetings will discuss the pros and cons of each system.

GROUP BUYS:

WAP is taking orders on a group buy for the Fat Mac upgrade. Send your $5.00 in to the office if you want to be in on this deal. (Include your WAP No., address, home and work phone.)

Interest was expressed in having a group buy on Mac-Pascal, Sargon, Odesta's Helix and Smoothtalker. To repeat a well-worn phrase - if you want to keep on top of group buys, you need to call the office or attend SigMac meetings. Since MacPascal is an Apple product sold only to dealers, our group buy program is unlikely to provide as good a discount as for other software.

ODDS AND ENDS:

Members were cautioned to be skeptical about doing business with a company that only advertises a post office box. If you have need to find out the real address of a business using a P.O. Box, write the Postmaster at the zipcode of the post office box. Include evidence that the person/company is using the P.O. Box for business purposes.

Thanks to Tom Berilla and a New Jersey member for their response to our call for articles on the Mac. See their contributions elsewhere in the Journal. To repeat - we need good articles about your experiences with the Mac or applications. You may write the article yourself and mail it to Bernie Urban (preferably on a diskette which will be returned to you) at the WAP office or you may ask for assistance from me (see previous issue for address and phone number) or from Steve Hunt, (301) 262-9080.

If you have questions you can direct them to Steve Hunt, Don Landing (Metro number in Woodbridge, (703) 690-1010) or Withers Morgan, (703) 971-0666, work or (703) 823-6033, home.

SOFTWARE DEMONSTRATIONS:

Thanks to David Morganstein, we were blessed with access to a large screen projector hooked to the Club's Mac. This eliminated the need for everyone to crane their necks looking at a single 9" screen. What

a welcome improvement that WILL continue.

We were treated to demonstrations of a wonderful array of software. We both saw and heard progress - our patience is being rewarded with some outstanding products! Judging by audience response, games still win the greater part of our enthusiasm, but applications programs like Odesta's Helix and new technology like Smoothtalker win our admiration and occasionally our awe. The demonstrations included:

MacCheckers by Videx ($49.00). This package provides both Checkers and Reversi. The Mac user interface is very well implemented and there are several options which make the games more challenging - three levels of difficulty, a recap of all prior moves, a menu command which will indicate the available moves, and several other similar options.

Sargon III by Hayden Software ($49.95). This chess game provides a superb example of effective use of the Mac user interface and high resolution screen. The program offers many excellent features - to allow the Mac to play against itself, provide varying levels of experience or skill for the Mac to use against you, to permit you to set up a game in mid-play, or to follow along with famous games. A definite hit. For those of you who own the game, a double click directly on the hand in the icon will open this application.

Disk-Based MacWrite by Apple (free upgrade). This upgrade removes the current 8-10 page limit on Mac-Write documents (now limited only by the amount of space on the disk), provides printing at 6 lines per inch and fixes a few other bugs. It will be available from your dealers in November.

Mouse Stampede by Mark of the Unicorn ($49.00). This is a hilarious animated, musical arcade game that is similar to (but better than) Centipede. Look out for the moving sneakers, knives, shopping carts and paint brushes.

Helix by Odesta, pre-release Version .3, ($399.00). This is a totally new concept in developing programs using icons and visual relationships to define records, manipulate data (over 30 functions going far beyond adding, subtracting, multiplying...), sorting information, and producing reports. This will be a most impressive product when released (rumored to be October/November 84).

Smoothtalker by First Byte ($149.95). Cheers and applause from those who witnessed this software product (no added hardware required) that reads standard MacWrite files. You can teach pronunciation of "words" it can't readily understand (like $, Mrs., or etc.). There are software controls to select a male or female voice, to vary pitch, to vary speed, and to interface these routines to your own programs. Excellent quality voice output.

SPECIAL SESSIONS:

After the demonstrations, we split into two groups, new users and advanced users, to address additional questions and help users resolve specific problems. Based on the degree of interest shown, we intend to continue this approach for future Saturday meetings.

If you weren't there, you missed an outstanding meeing. We hope this will set the standard and that we will see you in November.   ⌘

NOTE: Early deadlines for next 2 issues of the Journal - November 5 and November 31.

MAC NOTES
by Basil & Arlene Decina

LATEST VERSIONS OF MAC SOFTWARE

Product	Manufacturer	Version
Finder	Apple, Inc.	1.1g
MacWrite	Apple, Inc.	2.20
MacPaint	Apple, Inc.	1.3
Multiplan	Microsoft	1.02

MACWRITE BUGS

The following bug was encountered in our Version 2.20 of Apple's MacWrite:

A problem occurs with word wraparound when creating numbered paragraphs using hanging indentation and center justification.

1. This paragraph only contains one line and is correct.

   This paragraph contains more than one line. The paragraph number ("2.") is visible while creating the paragraph but disappears when the automatic word wraparound takes effect. (Note: There is a TAB character between the number and the first word.) The paragraph number still exists (it will reappear if a RETURN is inserted anywhere in the first) but it is neither displayed nor printed.

3. There are two "work-arounds", each of which has a problem.

4. The first "work-around" consists of replacing the TAB character (between the paragraph number and the first word) with an "appropriate" number of BLANKS. This paragraph is such an example. The problem with this solution is that after the first word is lined up, the automatic word wraparound and center justification features will cause the space to expand (or contract) in order to fill up the line. One can compensate for this by deleting (or inserting) extra blanks, but if the paragraph is modified the first line may wrap on a different word and the number of blanks will have to be changed again. In addition, one can not always place the first word in the correct position (due to proportional spacing).

5. The second "work-around" is to give up center justification and use left justification (as in this case). The problem with this is that one admits defeat and gives in to the COMPUTER.

MULTIPLAN BUGS

The following bugs were encountered in our Version 1.02 of Microsoft's Multiplan:

Bug 1: A problem occurs when performing a function over a large selected area within a worksheet that uses 70 to 80 percent of the available memory of a 128K Mac. Multiplan attempts to execute the function but becomes locked up in a state that causes the disk to spin indefinitely (we can't tell whether the disk is reading, writing or seeking).

Cause: Multiplan has run out of physical memory but doesn't know it.

Solution: One way to recover is to hit the interrupt

1001 BINARY TALES: Mac Innards 2
by Raymond Hobbs

MAC INNARDS -- Part 2 of what may be a bunch. Last month we looked at some of the fundamental concepts underlying the core (no pun) of the MAC'S ROM routines - Quickdraw. If you haven't read last month's discussion, I advise you to get it under your belt before you tackle this month's article, which goes into Quickdraw's routines in greater detail.

In the last article, I devoted quite a bit of space to discussing the manipulation of grafports. I also defined a grafport as a record of 25 fields (or properties), which may have seemed to beg the question of really providing a hands-on definition. However, that definition was (and at this juncture still is) the best I could do. It is best to think of the properties of grafports as describing what is done, and the grafports themselves as describing how and where it is done. Each concept must really be defined in terms of the other.

In order to round out the preliminary discussion, let's study some of the properties associated with grafports, keeping in mind that the grafport itself is the black box wherein these properties are made manifest to the programmer.

GRAFPORT PROPERTIES.

Properties of grafports are typed; that is, they consist of data types which can be either immediately recognized (integer, real, character, boolean) or broken down into recognizable sub-types. Therefore, a type might be "style", which consists of several sub-types, each of which is of type "bitmap", which in turn consists of an array of type "integer". As you will see, when you manipulate a grafport, you are manipulating a rather amazing amount of data. The best way (in my opinion) to look at these properties is by group. Keep in mind, though, that every image on your MAC screen comes to you courtesy of a grafport, and that every grafport carries with it all of these properties (although they might not all be used by the programmer).

PORT DEFINITION PROPERTIES (where you operate).

1.   Device. This property (field of the grafport record) is an integer which indicates the device upon which the grafport will be displayed. This will normally be the Macintosh screen. Images which are transmitted to the printer or other output device can be passed through a ROM routine which translates the output so that it interfaces properly. There is another property of grafports, which we shall look at anon, which assists in this operation. The programmer, therefore, does not normally have to worry about output to any device other than the screen.

2.   PortBits. This property is a bitmap of the image within the grafport. You will recall that last month I mentioned that the grafport was a rectangle, but that the image could be any shape. The portBits bitmap defines that shape. Note that the bitmap will define only the shape, not other properties of the image.

3.   PortRect. This field describes a rectangle which defines the grafport limits for height and width. These limits are not constrained by the Macintosh screen size. In other words, your grafport may be larger than the screen. As a matter of fact, it may be larger that Macintosh's internal memory. The grafport is a logical, rather than a physical concept, although there is a logical/physical overlay available to the programmer.

4.   VisRgn. This field is a record describing the portion of the grafport which may be made visible. This area need not be rectangular, since it too may have an associated bitmap. As a matter of fact, any region of a grafport may consist of multiple areas, either disjointed or connected, and may have "holes" in it.

5.   ClipRgn. This property is another record, this time describing the area of the grafport which may be affected by pen operations. The clipRgn has the same types of attribute as does the visRgn. The drawingboard of MacPaint is a simple example of a clipRgn.

IMAGING PROPERTIES (what you see, what you draw)

1.   BkPat. This field is a pattern (an 8 x 8 bit square depicting a repeating design) used to define what is seen as the background of an image (what you see when you erase an image).

2.   FillPat. This property is another pattern, this time describing the design used to fill in areas of an image.

3.   PnPat. The third pattern property defines the pattern drawn by pen operations such as line drawing, etc. The use of three pattern properties affords the programmer the opportunity of easily drawing a shape (pnPat), filling it in (fillPat), and erasing it (bkPat), without worrying about what the screen looks like after the completion of each operation. Compare this with the Apple]['s DRAW, XDRAW, or Assembly Language OR and EOR operations.

4.   PnSize. This property defines the height and width of the pen used for drawing operations. Again, this is not constrained by the screen size.

5.   PnLoc. This field indicates the position of the pen within the grafport. Note that since the grafport is a logical entity not constrained by the physical limitations of the MAC, the pen location never refers to a pixel on the screen, but rather to a coordinate pair within the grafport's defined coordinate plane. In other words, you never have to worry about whether you are drawing on a visible line, or to one side of it or to the other.

6.   PnMode. This property determines the manner in which a pen drawing operation interacts with whatever image (or pattern, or anything) is already on the drawing region. This property corresponds to a determination of whether to DRAW, XDRAW, AND, OR, or EOR on the Apple graphics screen; however, in Quickdraw, there are 8 boolean operations from which to choose.

7.   PnVis. This field determines whether or not the pen draws on the screen in a visible manner. It is a "hide pen/show pen" type of property.

contd.

TEXT PROPERTIES (typesetting modern-style).

1. TxFont. This field is an integer specifying which text font the Font Manager should use to print to the screen.

2. TxFace. This property specifies the appearance of the selected font (bold, underline, italic, etc.). Since this field consists of a set of style types, more than one style can be selected at once (such as bold outline, italic underline, etc.).

3. TxSize. This field determines the size of the typeface in points (one point = 1/72 inch). If the selected size is not contained in the Font Manager's font sets, the Font Manager will scale a size to meet the specifications set in txSize.

4. TxMode. This property does for text what pnMode does for graphics.

5. SpExtra. This field is an integer which specifies the number of pixels by which each space character may be extended to fill out a line in full justification writing (such that both left and right margins are justified).

COLOR PROPERTIES (support for the future).

NOTE: Although MAC hardware does not provide support for color graphics, Quickdraw grafports do provide that support. Therefore, at such time as color hardware becomes available, programs which have been written using Quickdraw's color support options will run with color automatically. Since we are in the preliminary investigation stage, however, we will only take a short glance at these three color properties.

1. FgColor. This property sets the foreground color to one of 8 standard colors (black, white, red, green, blue, cyan, magenta and yellow).

2. BkColor. This property sets the background color to one of the 8 colors.

3. ColrBit. This property defines the way in which a color picture is imaged onto the graphic plane. The way it works depends upon the color information transferred to the output device and is (at present) beyond the scope of this article. Since color output is not yet supported in the hardware, it is also beyond the scope of the present MAC.

PICTURE PROPERTIES (record/playback).

NOTE: The Macintosh has a journalling facility which is of minicomputer quality. Essentially, journalling gives the computer the capability of storing events as they occur in a buffer. Quickdraw has procedures which presumably interface with this journalling facility in order to create files of graphic regions as they are created. Such files are then saved, and a handle returned to point to each file (similar to BSAVEing an Apple binary picture file for later reloading, only a million times faster). The following properties provide the access handles to saved pictures.

1. RgnSave. This property reflects the state of a region (graphic area). If no region has been saved, it contains NIL, else it contains a handle to the region definition information.

2. PicSave. This field reflects the state of a picture image, and is treated the same way as RgnSave.

3. PolySave. This field reflects the state of a polygon, and is treated the same way as RgnSave.

MISCELLANEOUS (getting hardcopy)

1. PatStretch. This field is used by Quickdraw itself. It contains information used by Quickdraw to expand patterns (if necessary) during output to a printer. The programmer should not change its value.

GRAFPORT USAGE.

Now that you have seen what properties grafports have, your next logical question is, "How do I use the grafport itself?". The answer to this good question is that Quickdraw provides over a hundred procedures and functions for using grafports. The first of these... ...well, I have to leave something for next month, don't I?   ♉

Mac Notes contd from pg 29

key (using your programmer's switch) and cause the system to bomb. The Mac will then restart but you will have lost your latest changes (not your original worksheet). According to Microsoft, this problem cannot be corrected on the 128K Mac; we will have to keep our spreadsheets small and wait for the 512K version. They will try, on a future release, to catch this error and report an "out of memory" condition.

Bug 2: Another problem occurs when trying to print a large selected area within a worksheet that uses 60 to 70 percent of the available memory of a 128K Mac. Multiplan attempts to print the selection but causes the Mac to hang.

Cause: Multiplan has run out of physical memory but doesn't know it. (Same as before).

Solution: Recover the same way: hit the interrupt key (using your programmer's switch) and cause the system to bomb. The Mac will then restart but a dummy print file will be found on your disk. Microsoft responded the same way: this problem cannot be corrected on the 128K Mac; we will have to keep our spreadsheets small and wait for the 512K version. Again (on a future release), they will try to catch this error and report an "out of memory" condition.   ♉

MACPASCAL: What's on the Menu?
by Rich Cogan, Tom Riddle &
B.T. Sawyer, csc

INTRODUCTION:

Our initial excitement with Macintosh Pascal began to fade when we found that the manuals do not document any menu handling routines. The disk has a note on what they call "Inline" routines and says they are for experienced programmers. This is true because the note gives only a sketch of the routines. Perhaps someone wanted to leave material out in order to publish a book on "What Macintosh Pascal Dosen't Tell You." Menus are supposed to be the heart of the user interface and Mac's "claim to fame", so it would seem that the manuals should have spent some time explaining how to create and use menus. As it is, programmers will tend to use the "older" style of giving a list of options and asking the user to type in a letter or whatever - exactly what Macintosh was supposed to avoid.

The sample program, Text Editor, is given as an example of Inline usage. It is an example but memory limitations prevent any kind of documentation. Again, it's useful only for experienced programmers. In fact, we figured out the purpose of Inline before we found the note on it. The figuring wasn't always easy and we generated a lot of system "bombs" in the process. The result of our figuring is the material that follows. Creating menus dosen't have to be difficult. As with everything else, if you follow directions (and have a programmer's switch installed and perhaps write protect the disk) you can do it.

INLINE SYNTAX:

InLineP - a procedure call that directly accesses the Macintosh Toolbox.

    ::= InLineP(Trapnumber [,parameter list])

    Trapnumber ::= $A8nn|$A9nn   {nn is the trapnumber used to trigger the given procedure. The $ indicates a hexadecimal number}

    parameter list ::= {whatever parameters are required by the called routine}

cInLineF - a function call that returns values of the type indicated by c.

    BInLineF(Trapnumber [,parameter list]) - returns an 8 bit word (Boolean)

    WInlineF(Trapnumber [,parameter list]) - returns a 16 bit word

    LInLineF(Trapnumber [,parameter list]) - returns a 32 bit word (longword)

Pointer - a function that returns the handle (pointer) of the trapnumber passed to it.

    Pointer(LInLineF(Trapnumber [,parameter list]))

OUTLINE OF A MENU MAKER:

This little program will let you create your own menus (temporary ones). It can be inserted in the program Menuse (listed below) if you wish. MacPascal does no type checking on parameters given in InLine routines so if you make a mistake, you won't get an error message. Rather, the system will bomb. Therefore, save before you run and proofread carefully.

```
PROGRAM Menusee;

{an example of setting up menus for Mac. rc, tr, bt}
{10/9/84}

CONST
   Getmenubar = $A93B;      {These are the handles for}
   Clearmbar  = $A934;      {the Toolbox routines used}
   Newmenubar = $A931;      {in this program.}
   Appendmenu = $A933;      {There are others.}
   Insertmenu = $A935;
   Drawmenubar= $A937;
   Setmenubar = $A93C;
   maxmen     = 5;

TYPE
   signedbyte = -128..127;
   ptr = ^signedbyte;
   handle = ^ptr;

VAR
   oldmenuptr : handle;
   menuptr : ARRAY[0..maxmen] OF handle;
   Applesign,ch : CHAR;
   menutitle, labels : str255;
   menuid,count,j : INTEGER;

BEGIN
   Applesign := CHR(20);   {the ASCII number for the}
                           {apple logo}
   menuid := 100;          {first menu on the bar}
   count := 0;
   showtext;               {make text screen visible}

   oldmenu := Pointer(LInLineF(Getmenubar));
        {get the handle of the present menu bar}
        {the menu bar includes all of the individual}
        {menus - this allows you to restore the}
        {original at the end of the program}

   menutitle := Stringof(Applesign);
   {we chose the apple to be the title of the first}
   {menu in the bar}
   {The next three steps are repeated for each menu}
        {1. Get the handle for a particular menu}
        {2. Append text to that menu}
        {3. Get a name for the next menu (or exit)}

   REPEAT
     writeln('label please');
     readln(labels);        {see label syntax rules below}
     IF labels = '' THEN
       labels := ' ';  {a label may not be null}
     menuptr[count] :=
       Pointer(LInLineF(newmenubar, menuid, menutitle));
          {get the handle for first menu (ID = 100)}

     InLineP(appendmenu,menuptr[count],labels);
          {append labels to menu being created}
     menuid := menuid + 1;  {next menu's id number}
     count := count + 1;
     writeln('title for next menu or <RETURN> to exit');
     readln(menutitle)
   UNTIL (menutitle = '') OR (count >= maxmen);

   InLineP(Clearmbar);  {clear the present menu bar}

   FOR j := 0 TO count - 1 DO
     InLineP(insertmenu,menuptr[j],0);
```
contd.

```
     {put the menus just created into the bar}
     {one after the other. 0 means last in line}

   InLineP(Drawmenubar);
        {puts your newly created menu bar in place}

   {do whatever} {e.g. read(ch)}
   {so you have time to see what you've made}

   InLineP(Setmenubar,oldmenuptr);
        {restore pointer to original menu}
   InLineP(Drawmenubar);
        {put the original one back}
END.
```

LABELS SYNTAX:

Labels are made up of strings of characters, one string
to a menu. The syntax statements below are in Backus-
Naur form.

labels ::= 'item [qualifier] [{;item [qualifier]}]'

item ::= substring

qualifier ::= ^n | !c | <style | /c | (

style ::= B | I | U | O | S

          {bold, italic, underline, outline, shadow}

^n - caret followed by an icon number. The icon will be
     shown in the menu.
!c - exclamation point followed by a character that
     will be used to mark the item.
/c - slash followed by a character which gives the
     keyboard equivalent for the menu choice.
(- disables the item in the menu.

A label for a menu could look like this:

'Pause;Halt'   or
'Cut/X;Copy/C;Paste/V;Clear;Select All/A'

If you want to put a check mark { CHR(18) } into the
label string you have to declare it first and then
insert it.

labelstr := 'Cut/X! ';   labelstr[7] := CHR(18);

To create a blank entry leave at least one space.
           e.g. label := 'Howdy; ;Doody'

HANDLING MENU CHOICES:

The Mac has built-in routines to monitor events that
occur in the system. One of these events is called
"mousedown". If this event occurs in the menubar the
system can use it to identify which menu and item were
chosen. You can use this to direct the actions of your
program.

```
PROGRAM Menuse;
     {outline on detecting and acting on menu events}
     { rc, tr, and bt    10/9/84 }

CONST
  menuselect = $A93D;
  menukey    = $A93E;

TYPE
  EventRecord = RECORD        {Eventrecord is known to}
        What : Integer;       {the system and need not}
        Message : Integer;    {be declared.}
        When : Longint;
        Where : Point;
        Modifiers : Integer
      END;
```

```
VAR
  event : EventRecord;

PROCEDURE CheckMenu (event:EventRecord);
{If a menu event has occured, do something}

VAR
  ch : CHAR;
  selection : LONGINT;
BEGIN
  ch := CHR(Event.Message);
  IF Event.What = 1 THEN    { "mouse down" }
    selection := LInLineF(menuselect,Event.where)
                         { See note on cheating }
  ELSE
    selection := LInLineF(menukey,ch);
  CASE Hiword(selection) OF     {Hiword gives menu ID}
    100:
        CASE Loword(selection) OF {Loword gives}
                             {item ID}
           1: {first item action};
           2: {second item action };
                 { etc. }
           otherwise  {Macintosh Pascal supports this}
        END;
    101:
        CASE Loword(selection) OF
           1:
           2:
           otherwise
        END:
     etc
     otherwise
  END; {case Hiword}
END; {Check Menu}

PROCEDURE CheckKey(event:EventRecord);
     {a keypress event has occurred.}
     {If it is a command key, send it to Checkmenu}
VAR
  ch : CHAR;
BEGIN
  event.message := BitAnd(event.message,$FF+0);
  ch := CHR(event.message);
    IF  BitAnd(event.modifiers+0,256+0) = 256   THEN
CheckMenu(event)              {command key event}
  ELSE
  {do whatever with the key press}
END;

BEGIN { Main Program }

  {Setup menu bar.   e.g. insert Menusee}

  REPEAT
    REPEAT
      {wait around}
    UNTIL GetNextEvent(-1,event)
          { -1 is a mask for get any/all events }
          { any event makes the function TRUE }
    CASE event.what OF
      1: CheckMenu(event);
           { 1 means a "mousedown" event }
      3,5 : CheckKey(event);
           { 3 = keypress  5 = autokey    }
      {other events}
      otherwise
    END;
  UNTIL  {all finished}
  { restore original menu bar }
END.
```

NOTE ON CHEATING:

In PROCEDURE MenuCheck, event.where is treated as a
longinteger even though it is of type point. This type
conflict is not noticed and the proper results are
produced. It makes it much easier to run the CASE
statements that follow because the Hiword and Loword

A PAGE FROM THE STACK
by Dana J. Schwartz

As the New Disk Librarian I've emerged from the confusion left in the wake of a surprise departure by the previous tenant of this office, to bring you the first page of a new chapter from "the Stack". I'm sure you will all join with me in thanking Bob Platt for the fine work he has done for Washington Apple Pi, both in this job and during his short term as Vice President.

Our library has grown to its present size chiefly due to the contributions of our members. Lately, however, the library's growth rate has declined and we are finding it more and more difficult to produce a new disk each month as we have in the past. We welcome any contribution in any Apple language running on any Apple computer, as long as it is not copyrighted. Remember that you will receive, in return for a disk with your program(s) on it, any WAP Library disk of your choice. Please help us to keep your library expanding! If you have any questions about the operation of our library, see your Member Reference Handbook, or contact me at any time.

Two new Macintosh disks will premiere this month, SigMac 6 and SigMac 7. They were created by editing six public domain disks which we received recently from the San Diego Macintosh Users Group. SigMac Librarian Tony Anderson has provided the following descriptions:

SigMac Disk 6: Mac Paintings

This is the first single application disk in the SigMac library. It consists entirely of 13 MacPaint documents, and is filled with art you can see, touch and modify to your heart's content.

SigMac Disk 7:

In the MacWrite folder:

Acme Burger (by Susan Kare) - You may have seen this in the Premiere Issue of MacWorld. It was done for fun but gives you ideas about ways to make your next memo more interesting.

Go Documentation (by George Acton) - Instructions for the GO game in the MS-BASIC folder.

User Group List - A short and already outdated list of Macintosh User Groups.

In the Bits & Pieces folder:

Desk Cal - Here is the calendar that we have all been waiting for. It can be used as an application or as a desk accessory. It is perpetual, it allows you to keep notes on each day, and it's almost free. It does however use over 20K of disk space.

MacAsm Demo (by B. Shillings) - This is a commercial for MacAsm from a company called Mainstay. It is short and not at all hard sell. What it does however may very much make you want to buy their product.

In the MS-BASIC folder:

DISASSEMBLER (by R. Nicholson) - Now you can start exploring the wonderful world of machine code. The Mac toolbox routines will soon be at your command. Just be sure to keep the caps lock key down for this one. Let the rest of us know what you find inside

Macintosh.

TEXT.ASM (by R. Nicholson) - This file is used by the disassembler.

Analog Clock - The clock from the March St. (rest his/her soul) Mac.

Living Art (by Paul Edl) - A little kinetic art work to sooth the mind and rest the eyes.

Go (by George Acton) - The Japanese game upon which Othello is based. See the instructions in the MacWrite Folder.

The Nerd (by Robert Eberhart) - This doesn't look like anyone I know.

Abe & Mona (by Patrick Cosgrove) - Two digitized photographs of old favorites.

ABM-ICBM (by Keith Nemitz) - A story of mice and missiles.

Calculator (by Brady Graham) - A basic version of the desk calculator with added scientific functions.

CR Stripper (by Don Melton) - Got too many carriage returns in that new text file? Well, use this program and send them all to the bit bucket.

Dvorak Keybd (by Ronald Nicholson) - This interesting little program shows how the Mac keyboard can be completely redefined. Could the extra option keys be redefined to act as function keys or cursor keys?

FancyList (by Buzz Hamilton and Doug Forman) - Page numbers your program, prints each command in a multi-command line on a new line, and more. The program to be printed must be saved in ASCII format and end with the suffix .bas for this routine to work.

File Compare (by John W. Bacter) - Compares two text files and notes the differences on the screen or on the printer.

File Printer (by Jim Hanson) - This program will allow you to send any ASCII text file to the printer no matter how large. Good for text files downloaded from another machine that are too large to print with MacWrite.

Font Display (by Alan Galumbeck) - Displays a chart of the selected font showing the numeric code associated with each character.

MacCopy 1.4 (by Dennis Brothers) - An updated copy of a Brothers favorite.

Pattern Editor (by Pat Cosgrove) - This editor allows you to create a pattern in an 8 by 8 grid of cells. The pattern and the hex values for a single row of cells will be displayed. This is useful for pointer creation. Also, several toolbox routine calls in MS-BASIC use a pattern.

SetNewPointer - A Basic demo to show how to redefine the mouse pointer.

REM Remover (by Christopher Allen) - This programming aid will remove all but the first remark from any

TWO MAC "HOW-TO" SUGGESTIONS
by Tom Berilla

Following are solutions to two problems using a hard disk drive, the Tecmar Macdrive (MD), with a Macintosh. These were prepared by the author with help from a phone conversation with a nice guy named Russ, of Tecmar.

PROBLEM #1: HOW TO BOOT "MULTIPLAN"

The Multiplan master floppy disk is protected. It must be booted in the MD for the Multiplan program to work from the MD. The Multiplan program will not work on the MD if it is only booted to the Mac, and not to the MD. This procedure is not documented in the MD Manual. (This may be a problem for other brands of hard disk drives as well.)

THIS IS HOW RUSS AND I DID IT:

1. Insert "MDSystem Disk" into the Mac's floppy disk drive.

2. Insert "MacDrive Hard Disk" into the MD.

3. Boot both disks as instructed by the MD Manual.

4. Open the "MacDrive Window" on the Mac's screen.

5. Open a "MacPaint" file.

6. Open "File" to "Save As" or "Save" on the pull down menu.

7A. If "MDSystem Disk" is now shown in the window above "Eject", click "Eject"
    -or-
7B. If "MacDrive" is now shown in the window above "Eject", click "Drive" to get "MDSystem Disk" to show in the window.

8. Click "Eject" to remove the "MDSystem Disk". (Mac's floppy disk drive will remain empty until later.

9. Click "Cancel"

10. "Save Changes" appears. Click "No".

11. Open "File" and click "Quit".

12. Open any "Multiplan File".

13. Mac will call for "Multiplan Master".

14. Insert "Multiplan Master" disk into Mac's floppy disk drive.

15. Mac will call for "MacDrive" hard disk.

16. Push "White Button" on the MD hard disk drive door. (This will open MD door, eject the "Hard Disk" and the "Red Light" will blink on the door.)

17. Wait until "Red Light" stops blinking on the MD door and listen for the disk to stop its sound of turning.

18. Close MD door. This reboots with the "Multiplan Master" in the Mac's floppy disk drive. (This will take at least 15 seconds.)
    (The MD is now booted for Multiplan". The MD is

in control instead of the Mac and the "MacDrive System" disk).

19. If desired, eject the "Multiplan Master" from the Mac's floppy disk drive.

20. This completes the Multiplan boot with the MD hard disk drive.

PROBLEM #2: HOW TO MAKE "EMPTY FOLDER" COPIES

DON'T COPY EMPTY FOLDERS, DUPLICATE THEM!

If you "Copy", instead of "Duplicate", "Empty Folder", all or many of the individual files will jump into the original "Empty Folder" like garbage goes into a garbage can, when you shut down and then reboot. The screen will also show, "Copy of XXX, Copy of a Copy XXX, Copy of a Copy of a Copy of XXX," etc. for each icon.

THIS IS WHAT TO DO AND WHAT NOT TO DO:

1. When you want to copy a bunch of "Empty Folder" icons to fill with files of different categories, do not "Copy" an "Empty Folder" from a floppy disk window to the MD window, after you have one and only one "Empty Folder" in the MD window.

2. Use "Duplicate" from the "File" pull down menu to make copies of "Empty Folders", i.e. one or more copies.

3. Name each folder for the category you need.

4. Shut down and reboot.

(The folders and files should all be as neat and as organized as they were before.)   ⑤

A Page from the Stack contd from pg 34

MS-BASIC file stored in the ASCII format. Helps to save space and time.

The following four files are all music programs based on the original Dennis Brothers program:

Music/Star Wars (music by John Williams, routine by Dennis Brothers, adapted by Joe Gagnon).

Music/Bach Minuet (music by JS Bach, routine by Dennis Brothers, adapted by Joe Gagnon).

Music/Leia's Theme (music by John Williams, routine by Dennis Brothers, adapted by Joe Gagnon).

Music/New World Symphony (music by Anton Dvorak, routine by Dennis Brothers, adapted by Joe Gagnon). ⑤

LETTER TO THE EDITOR:
Supporting "Inside Macintosh"
by Michael Hartman

Dear Bernie,

I would like to correct two statements about "Inside Macintosh" made by Raymond Hobbs in the October 1984 WAP Journal, and also address some other common misinformation and misunderstandings about "Inside Macintosh" and the difficulty in doing Macintosh programming. First, the preliminary version of "Inside Macintosh" is available for purchase by anyone, not just certified developers. Regular updates to the document, the final published version, and three disks of rough-form Macintosh utilities are available for an additional $100. Five Lisa Pascal Workshop diskettes are also included. Second, a listing of Macintosh ROM trap values IS contained in the document in the section labeled "Debugging" (where it has been for the past several months) and in the new "Operating System Utilites" section (released in August). As the existence of MacTep and other programs from MAUG on the SigMac disks demonstrates, this is sufficient to allow use of the ROM routines from MS-BASIC (although their use is limited by the interpretive nature of MS-BASIC). Note that the ROM routines are not invoked by the usual jump or branch instructions, but by the use of an inline trap mechanism. Thus, the traditional table of calling addresses is not appropriate.

Many people regard "Inside Macintosh" as opaque, and have been frustrated in their attempts to gain an understanding of how to use the Operating System (OS), Toolbox, and QuickDraw routines. I believe that this is not due mainly to shortcomings of the document, but a result of people's incorrect expectations about the nature of the ROM routines, often based upon their experience with the Apple // Monitor ROM. Also, considerable prerequisite knowledge is demanded by the nature of the Macintosh operating system and the complexity of the ROM routines themselves. There are some gaps and rough spots in the documentation (it is, after all, a preliminary release), but comprehending and using "Inside Macintosh" are not seriously restricted by these.

On the Apple //, the 2K of Monitor ROM (the rest is Applesoft BASIC) is a collection of approximately fifty subroutines which perform simple character I/O, lo-res graphics, and assorted other tasks. The routines can for the most part be used independently of each other, require only one or two arguments which are passed in the 6502 registers, and are simple enough to be documented in two or three lines each. In the Macintosh 64K ROM, there are over 400 routines which work closely together to form about 20 functional units. These include such complex and powerful modules as the Window Manager (which controls window creation, positioning, selection, and sizing) and the Control Manager (which performs the drawing, monitoring, and activation of such items as scroll bars, buttons, and check boxes). And the routines can't be used one or two at a time: to use a module at all typically requires calls to at least three or four of its routines. Any "Macintosh-style" program will use ten or more modules and well over sixty different ROM routines. The routines accept and return parameters through both the 68000 registers and the stack; these may be five or more Pascal-type data structures, and some of these may be composed of ten or more data elements. Obviously, two or three lines will not suffice to explain one of these ROM routines and its

interrelationship with the others in its unit.

The fact that "Inside Macintosh" is directed towards the Pascal programmer is not an arbitrary quirk. The internal workings of the Macintosh OS are fundamentally based upon such concepts as stacks, heaps, pointers, and structured data-types, which are some of the things Pascal was designed to support. Most BASIC and many assembly-language programmers have little or no experience with these concepts, and thus find the discussions of memory management and parameter passing hard to follow. Even for the accomplished Pascal programmer, the use of "handles" (essentially double pointers), the large number of supporting data structures, and the presence of both global and local coordinate systems in QuickDraw render "Inside Macintosh" and the actual programming hard going at times. But I believe this is because the material is intrinsically complicated, not because Apple is not producing documentation up to the high standards it has set for itself. People who want to understand "Inside Macintosh" and do "Macintosh-style" programming are going to have to bring some background in computer science and significant programming experience with them. Apple cannot be expected to provide either of these, and cannot be faulted for producing a software system complex enough to require them.

Unfortunately for the weekend computer hacker (meaning programmer/hobbyist), both applications and operating system software for microcomputers are rapidly increasing in complexity. A short three years ago, state-of-the-art commercial applications software could be written by one person in a few months. Now, most programs from that time are regarded as limited in both function and the much-touted "user-friendliness". VisiCalc is replaced by Multiplan, and PIE Writer is supplanted by Word Juggler. The new applications programs are written by teams of programmers, consuming man-years of effort. Similarly, the operating systems themselves are now expected to support such things as windows, pull-down menus, multi-tasking, and attachable device drivers. To take advantage of these environments requires a new level of interaction between the applications and OS software. Learning to do this demands an investment of time beyond that which most home computer programmers are willing to make. For these people, the Macintosh will host programming environments such as MacPascal and MacBasic, which provide limited access to the Macintosh OS and Toolbox routines. (Note that MacPascal will make the power of Pascal available to people who found it too much trouble to use on the Apple //.) Full-scale development environments on the Macintosh are coming, but Mac owners whose programming experience is 200-line BASIC programs probably won't be able to take full advantage of them. This is frustrating and unfortunate, but also true.

In my opinion, Apple's Macintosh operating system and the user-interface routines constitute a powerful, logically designed, and highly-integrated piece of software. It is an achievement which should not be underestimated. What can be done with a few calls to TextEdit surpasses what many word processing programs are capable of; the power of QuickDraw is truly unbelievable; and the user interface I can create with them will make my programs easier and more pleasurable to use. But the modeless user-interaction which is

DESK-TOP APPOINTMENT CALENDAR
by Steve Hunt & Put Armstrong

In addition to the Desk Accessory Mover discussed in a separate article, a perpetual calendar program has been developed to provide many additional features. This Desk Calendar will be distributed on a future Sig Mac disk. Stay tuned - it may be available by the time you read this article.

Depending on your needs, this FREE calendar may be better than Habadex.

Figure 1 shows a "page" from the calendar for the 10th of November 1984. The menu of months down the side allows you to quickly select another month. The bottom two blocks on the side menu ('83' and '85') allow you to move to the prior year or the following year. This is a true perpetual calendar which correctly calculates leap years.

You can enter up to 12 lines of information for each day. When information is entered for a particular day, that date is shown in "hollow" type style so that you can get a quick "month a a glance" overview of the appointments entered.

This program is an assembly language application which includes the necessary routines to load the calendar to the Apple Menu of any disk. You can load the Calendar where needed and use the Desk Accessory Mover to delete the program if it is no longer needed.

The price (FREE) is definitely right for this application but there is a caution. It requires over 20K of memory (which is not available for other applications) and the disk file can become quite large if you have many appointments. On a 128K Mac this application will use up quite a lot of space. Thus, we suggest that you load this calendar on a disk which you dedicate to the calendar or which contains small files.

♥ File Edit

Calendar

Saturday November 10, 1984   9:21 PM

Sun	Mon	Tue	Wed	Thu	Fri	Sat		Jan	Jul
				1	2	3		Feb	Aug
4	5	6	7	8	9	10		Mar	Sep
11	12	13	14	15	16	17		Apr	Oct
18	19	20	21	22	23	24		May	Nov
25	26	27	28	29	30			Jun	Dec
								83	85

0900 SIGMAC meeting at USUHS
1530 Racketball
1900 Dinner at Pleasant Peasant
2200 ???
2400+ Homeward bound?

"Inside Macintosh" contd from pg 36

the "Macintosh style" and which people want to write into their programs must be reflected in their structure. You can't lead the user around by the nose with a chain of menus - you have to let the user do whatever he wants whenever he wants to do it. So even with all the nice user-interface routines, a program has to be designed and coded to expect and respond to anything. Apple has made it possible for you to do this, but it couldn't make it easy. I hope that people will come to appreciate this; once tutorials on the Macintosh system software become available I think they will, and I think many will ultimately learn how to do what they want. Until then, I hope the word will spread that understanding "Inside Macintosh" and programming the Macintosh are activities which, while difficult, are far from impossible, and that Apple is doing its job to make them possible for people willing to put forth the required, unavoidable effort.

Sincerely,

Michael Hartman

MacPascal contd from pg 33

functions only work on longintegers (giving the high and low words respectively).

FINALE:

We are writing a utility that allows you to create and save your own menus in the system Resource file but we haven't finished and the deadline for articles is now. This article should give you something to fool around with until the next time. If something goes amiss during your experiments, remember that into every programmer's life a few bombs must fall.

DESK ACCESSORY MOVER
by Steve Hunt & Put Armstrong

Relief is at hand - at least for those who have grown tired of the ol' puzzle under the Apple Menu or who long for some new goodies to play with when you are bored.

Sig Mac Disk 5 (prepared by Tony Anderson and now available at the Apple Pi office) contains a folder - among others - which contains a new application called "Desk Accessory Mover" by C.E. Software of Des Moines, Iowa.

The first thing you notice is an "Honor System" request for you to send $15 to the developers (shown on the initial screen) if you choose to keep this program. From my perspective, this is a great step forward for lowering the cost of software. I strongly recommend that you keep this program AND that you send the requested donation to reward them for their fine efforts. If this approach works - if all Mac owners honor this type of request - then we can expect to see it used more often. If the developers don't get reasonable return on their investment, we can expect to continue to pay high costs for software distributed through normal channels. DO YOUR PART TO HELP LOWER THE COST OF MAC SOFTWARE.

Now on to the goodies. The Desk Accessory Mover allows you to delete any existing Desk Accessory on any disk.

CAUTION - We do NOT recommend deleting either the Alarm Clock OR the Control Panel. Deleting these accessories can cause your system to crash and require reloading the System File.

The Desk Accessory Folder comes with several new Desk Accessories that can be ADDED TO the Apple Menu on any disk. The screen image below shows the Desk Accessories that are availabe with the Sig Mac disk.

There are several slick little accessories now available. For example:

The "Executive Decision Maker" illustrates the laws of probability like a "Piccincko Machine".

The "Magnifying Glass" provides a somewhat useful utility to examine a portion of the screen in magnified form. This may help some of you with development of detailed MacPaint drawings.

The "Hex Calculator" provides a valuable tool for programmers who need to do calculations between decimal and hexadecimal conversions.

There is also a perpetual calendar which may be of value to many, but is not as complete as the Desk Top Calendar discussed in a separate article.

In summary, this utility program and the companion Desk Accessories are a welcome start to allow you to tailor your Macintosh to provide exactly the types of functions which are of the greatest value to you.
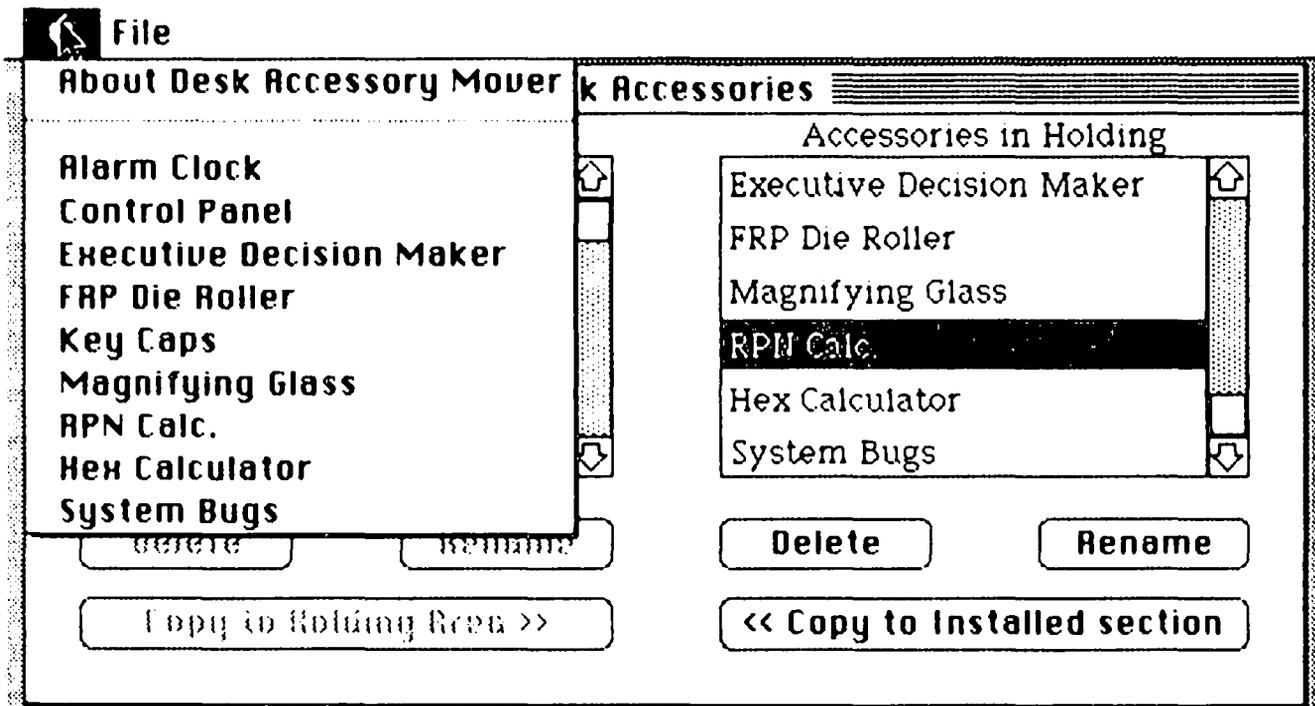
—— Operant Systems ——

—— HARDWARE ——

PRINTERS—
```
Epson FX-80 (160 cps, tractor/single sheet, graphics).. 435
       FX-100 (wide carriage version of the above)...... 625
       RX-80F/T (100 cps tractor/single sheet, graphics) 305
       LQ-1500 (200 cps, fantastic letter-quality mode) 1095
Fingerprint (pushbutton font selection for RX/FX).... 39/49
Okidata 92 (160 cps, graphics, BEST print for price)... 395
       93 (132 column version of above)............. 629
Toshiba 1340 (144 cps draft, BEST letter-qual matrix).. 799
Texas Instruments TI-855 (150 cps draft, 35 cps NLQ)... 795
IDS Prism-80 (200 cps, color/sheet-feed options avail). 999
Gemini 10X (120 cps, tractor/single sheet, graphics)... 289
       15X (wide-carriage, same features as Epson MX).. 409
       Powertype (18 cps daisywheel)................... 375
C.Itoh F-10 Starwriter (40 cps daisywheel, best for $) 1049
       A-10 (18 cps version of the above)............ 559
Silver-Reed 550 (18 cps daisywheel, best at this speed) 459
       500 (14 cps daisywheel, as above).......... 379
NEC 7710 (55 cps daisywheel, built like a tank)....... 1795
Diablo 620 API (25 cps daisywheel).................... 839
Qume Sprint 11/40 (40 cps daisywheel)................. 1295
```

MODEMS—
```
Hayes Micromodem JCe (tone dialing/speaker/Smartcom I). 239
      Smartmodem 1200 (1200 baud, RS-232, auto-dial)... 469
      Smartmodem 300 (300 baud, as above).............. 199
Novation Apple-Cat II (w/ Comware; 1200 baud capable).. 235
      Smart-Cat 300 (RS-232, direct-connect)........ 175
      Smart-Cat 300/1200 (as above)................. 389
Prometheus Promodem (300/1200 baud, RS-232, auto-dial). 365
Microcom ERA 2 (300/1200 baud card with software)...... 359
Zoom Telephonics Networker (300 baud card w/software).. 149
US Robotics Password (300/1200 baud, auto-dial/answer). 335
Anchor Automation Mark 12 (300/1200 baud, RS-232)..... 269
              Volksmodem (300 baud, RS-232)....... 59
```

DISK DRIVES—
```
MicroSci A2 drive (100% Apple-compatible Shugart 390).. 199
Rana Systems Elite One drive (40 track, 163K).......... 259
             Elite Two (40 track, double side, 326K)... 389
             Elite Three (80 track, double side, 652K). 449
             10-MEG Winchester drive................. 1149
TEAC Thinline drive (40 track, 100% Apple compatible).. 219
Axlon 320K Ramdisk.................................... 829
Davong, Corvus, and Corona Winchester drives.........(call)
```

CP/M & 6502C SYSTEMS—
```
Applicard (6 Mhz Z-80, 64K to 192K RAM, 70-col video).. 249
Microsoft Softcard JCe (Z-80, 80 col & 64K on one card) 265
         Softcard (includes CP/M 2.2 and MBASIC)...... 225
DR Gold Card (6 Mhz, 80-col, 64-192K, CP/M 3.0, CBASIC) 259
Titan Systems Accelerator JC (3.6 Mhz 6502C processor). 425
Speed Demon (6502C high-speed coprocessor)............. 219
```

MONITORS—
```
Amdek 300G (12" green anti-glare screen, 18Mhz)........ 125
      300A (12" amber anti-glare screen, 18Mhz)........ 149
NEC JB-1201/1205 (green/amber anti-glare screen, 20Mhz) 155
    JB-1260 (12" green, 15Mhz, best value for money)... 109
USI PI-2 (12" green anti-glare screen, 20 Mhz)......... 125
    PI-3 (12" amber anti-glare screen, 20 Mhz)......... 125
```

INTERFACES & BUFFERS & CLOCKS—
```
Pkaso/U printer interface (superior graphics & more !!) 129
ShuffleBuffer (32K—128K w/cut & paste/ser and par)..... 269
Grappler+ printer interface (parallel w/ graphics)..... 109
Buffered Grappler+ (16K to 64K buffer plus graphics)... 169
CCS 7711 Super Serial (for printers & modems).......... 95
Wizard IPI (graphics, text dump, page formatting )..... 69
Practical Peripherals ProClock (ProDos compatible)..... 109
```

VIDEO & KEYBOARD—
```
Videx Ultraterm (160 column/48 row display !!)......... 249
      Videoterm (80-column w/ softswitch & inverse).... 209
MicroSci 80-col card w/64K RAM for JCe................. 125
Keytronic KB-200 low-profile detached keyboard......... 235
```

MEMORY EXPANSION—
```
Wespercorp Wizard 16K RAM card (2-yr warranty)......... 59
Titan Systems 128K RAM card........................... 349
```

—— SOFTWARE ——

WORD PROCESSING—
```
Wordstar 3.3 (including 6 Mhz Z-80 Applicard !!!).. 299
ScreenWriter II (70 col display, spooling, more!!).. 85
Super-Text Professional............................ 115
Bank Street Writer or Speller...................... 45
pfs: Write JCe or Proof JCe........................ 85/49
Pie Writer......................................... 99
Word Handler + List Handler + Spell Handler........ 109
HomeWord or HomeWord Speller....................... 45/35
Sensible Speller IV (checks ANY file type)......... 85
The Word Plus (super spelling checker for CP/M).... 99
```

SPREADSHEETS & FORTUNE-TELLERS—
```
Multiplan (state-of-the-art spreadsheet)........... 129
FlashCalc (VisiCalc gets a facelift)............... 75
SuperCalc 2 (powerful CP/M spreadsheet)............ 179
```

INFORMATION MANAGEMENT—
```
dBASE II (w/ ZIP screen generator)................. 299
pfs: File, Report, or Graph........................ 85
The General Manager 2.0............................ 149
Thinktank (electronic thought organizer)........... 99
DB Master 4.0 (latest version)..................... 225
```

BUSINESS & ACCOUNTING—
```
Dollars & Sense (accounting + graphics)........69 & 79
Peachtree Peachpak (GL/AR/AP)........... all three: 229
          Back to Basics Accounting (GL/AR/AP)..... 129
BPI Accounting (AR/AP/PAYROLL/INVENTORY) ea module: 249
```

COMMUNICATIONS—
```
Ascii Express Professional (for DOS 3.3)........... 89
Z-term Professional (for CP/M)..................... 99
pfs:Access (basic communications for the JCe)...... 49
CompuServe Starter Kit (password & 5 free hours)... 29
```

LANGUAGES—
```
Microsoft TASC Applesoft compiler.................. 119
Softech UCSD p-System.............................. 425
Microsoft Fortran-80............................... 129
BDS C compiler..................................... 115
Micromotion FORTH-83............................... 79
Terrapin Logo...................................... 69
```

ODDS & ENDS—
```
Kensington System Saver Fan (w/surge protector).... 66
Kraft Joystick / TG Joystick / Hayes III...... 35/35/40
KoalaPad Graphics Tablet (with graphics software).. 85
Maxell, Verbatim, Dysan, Memorex, Wabash disks.. (call)
```

M	Filevision (database system).......................	129
A	Odesta Helix (another database system).............	249
C	Main Street Filer (yes...yet another database).....	145
I	Dollars & Sense....................................	99
N	Microsoft Word (what MacWrite should have been)....	129
T	Chart...	85
O	Mac the Knife (canned doodles).....................	29
S	T/Maker Click Art (...more doodles)...............	39
H	Kensington MacModem (300 baud).....................	99
	Surge Protector (replaces power cord)...	39
	Tecmar, Davong hard disk systems............... (call)

COMPUTER MUSIC - A REPLAY
by Raymond Hobbs

To be the first on the block is not always the way to gain fame, glory and respect. As H. H. Munro put it, "The early Christian gets the fattest lion", and one excellent way of throwing oneself to the lions is to be an innovator in a traditional field.

My experience in innovation has been in the way of computer music. I bought a Mountain Computer Music System in 1982, before it was incorporated into the Soundchaser and Alpha Syntauri synthesizers. The Alf system was (as it still is) the chief competitor to the Mountain system.

My family was the first audience to hear my Music System. Now, the family is small (9 total members, up to and including first cousins), but it is quite musically inclined, with four who are, or have been, professional classical musicians, and between the nine of us, we can play 19 different instruments. I supposed it to be the ideal audience for computer-generated music. Boy, was I wrong!

The reactions ranged from amused tolerance to abject horror and disgust. What happened? It wasn't the selection (Mendelssohn's "On Wings Of Song") or the orchestration (piano and clarinet with string bass). No, it was the perceived invasion of technology into a traditional area. One studies piano or clarinet or bass for years in order to play it, and my Apple was doing it with only a few seconds to learn to play. How irreverent. How condescending. How brutal.

Two years have wandered by, and my family has not been subjected to computer music since my first audition. I still design and play on my Music System, sometimes with traditional "instruments" and sometimes with instruments of my own creation. During the last two years, I've learned quite a bit about the way sound is produced and the way it is perceived, but mainly I've learned a lot about the way in which innovation finds its place in a traditional arena.

TRADITIONAL vs. COMPUTER MUSIC

Tradition is not to be trod upon, but rather to be built upon. In this way, computer music is not likely to find its place in musical tradition by replacing existing instruments or performers, but rather to generate new forms of sound. This poses no threat to today's traditional musicians - after all, Prokofiev did not replace Mozart or Haydn, but merely added to an already rich store of musical tradition, and as long as Prokofiev's or Mozart's works are performed, there will be a demand for musicians to play those works. Similarly, when computer-generated music attains the stature of the music of Bach or Chopin there will always be a demand for computer musicians - with one twist, which I will get to in a moment. First, I must take a brief excursion into the technical side of music synthesis. Please keep in mind that I am not a technical type, having only taken one science course (Astronomy) in college, and so my slant on this subject is that of a lay person - not a scientist.

TONES

Since music is comprised of a series of tones, tone generation is a good place to begin, and my discussion will start with the simplest form of tone generation - that of a pure tone.

A pure tone is the fundamental element of a complex tone. This fundamental element is called a sine tone because its waveform can be mathematically represented by a sine function. For those of you who (like me) have trouble with the math involved, imagine a graph of the action of a loudspeaker, where the cone's out-and-in action is shown as a positive-and-negative displacement over time:

FIGURE 1 :
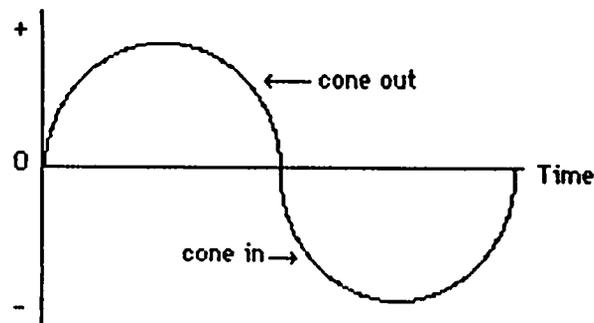
Bingo! A sine wave! If the horizontal axis (Time) represents a period of, say, 1/100 second, we can hear the resulting wave as a pure tone (human hearing varies, but tones in the range of 60 to 15,000 cycles per second are a pretty good bet). That is, if one wave is completed from starting point X=0 to ending point X=0 in both positive and negative phases, the resulting wave has a period of 1/100 second, or a frequency of 100 cycles per second (I note here that the scientific community refers to cycles per second as "hertz", a term no doubt used to try to confuse me. I will stick to CPS).

All sorts of things make tonal sounds, but instrument sounds normally come down to four groups: 1) those produced by a vibrating string; 2) those produced by a vibrating reed; 3) those produced by a vibrating column of air; and 4) those produced by striking a surface (which then vibrates). For expositional purposes, I will use group 1, stringed instruments.

When a string is excited (plucked or bowed or just nervous), it begins to vibrate. The fundamental tone produced by this action can be graphed as a sine wave. Suppose that the period of that wave is 1/440 second - the frequency is 440 CPS, which corresponds to what our ears perceive as "concert A" - the concertmaster has successfully given the orchestra the pitch to which tuning will commence. However, as the string vibrates, it sets up other vibrations (called "sympathetic" or "harmonic" vibrations) which are multiples of the fundamental. Therefore, the string, while vibrating at 440 CPS, is also vibrating at 880, 1760, 3520, and other multiples of 440 cycles. All at the same time. And therein lies the greatest problem of music synthesis.

HARMONICS

The simultaneous generation of harmonics, or overtones presents a formidable problem to the computer musician - and a considerable boon both to the traditional musician and to our ears, for it is almost impossible to precisely duplicate the sound of a traditional
contd.

instrument perfectly, even over a small range of notes.

If you generate a pure tone (a single sine wave), it will sound flat, lifeless. As a matter of fact, you probably cannot tolerate hearing such a tone for more than a couple of seconds (don't believe it? The Apple "SYNTAX ERROR" beep is a pure tone). The harmonics add much (but not all) of the "color" to a tone. We lump this, plus a number of other things, into what we term "timbre". Generating good harmonics is enough to keep a computer musician busy for months, but it is only a part of producing good tone color. For now, I will only briefly explain some of the problems of harmonics, then we will go on to other areas. Later, we can return to harmonics again to take a more detailed look at it.

Turning again to our sine-wave graph, we know that the Y-axis (horizontal plane) represents time. However, we didn't define the X-axis (vertical plane) previously. We will do so now. The height of the function plotted on the graph will be measured in terms of

FIGURE 2:

1/440 sec.
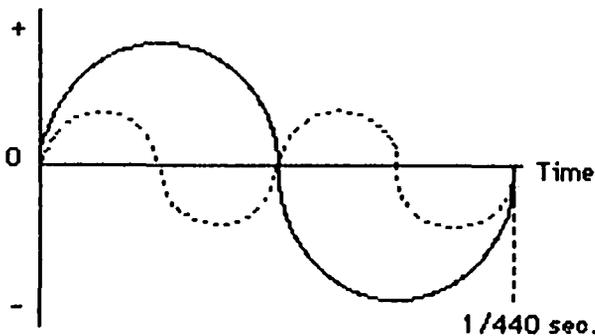
amplitude, such that the dotted line graphs a tone of the same frequency as the solid line, but at half the amplitude. Now, if we again suppose a tone at a frequency of 440 CPS (solid line), we can also plot the second harmonic (2 x 440 = 880 CPS) on the same graph. This has been done, assigning the second harmonic (dotted line) half the amplitude of the fundamental tone. The resulting sound can be graphed as the sum of the two functions, and appears as the ADDITIVE function graphed below. Note that when the fundamental and the harmonic are out of phase with each other (on opposite sides of the horizontal axis), their positive and negative amplitudes are cross-cancelling. Note also that if the harmonic were greater or less than 1/2 the amplitude of the fundamental, the resulting additive waveform would reflect the aggregate change.

FIGURE 3:

1/440 sec.
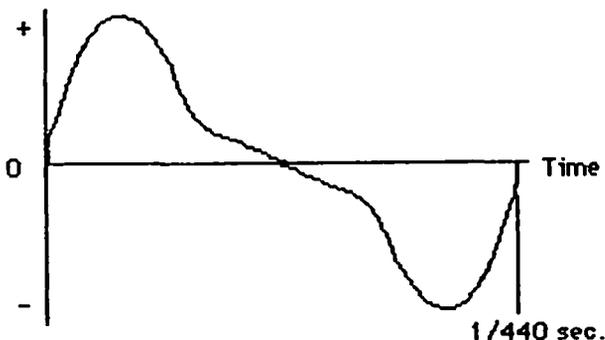
What it all means is that the amplitudes of all harmonics can be altered in order to change the timbre of the resulting tone. What controls the frequency of the wave? In our example of the bowed string, it is the length of the string, and this is controlled by the fingers pressing the string onto the fingerboard of the instrument. The harmonics are produced by sympathetic vibrations of the string, focusing at its half-length, third-length, fourth-length and so on. In the figure below, we see a string vibrating in four modes, first at the fundamental, then the second harmonic, third harmonic, fourth harmonic, then all together. Now, suppose that:

FIGURE 4:

Fundamental
2nd harmonic
3rd harmonic
4th harmonic
Combined
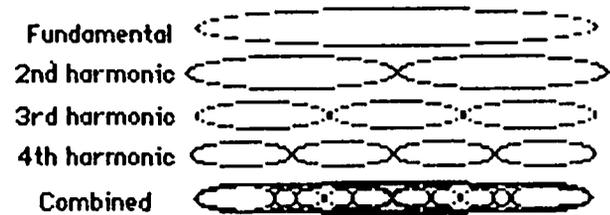
the violinist bows the instrument at precisely the one-third length of the string: the result is that the third harmonic would be muted, and the resulting sound would be different from that produced if the violinist bowed at, say, the 1/25 length of the string. If we were attempting to re-create such a tone, we would have to control the following variables:

1) What harmonics are present;
2) The amplitude of each harmonic.

There are other things which would have to be controlled in the waveform as well, but more on that later.

ENVELOPES

If we produced a waveform such as we described above, the result would still be a bland sound, without much character. Tones produced by traditional instruments have lives of their own - the sound doesn't just occur, but rather it is given life, it grows, matures, ages and dies. The phases of tonal life are called attack, decay, sustain and final decay, and these phases also have mathematical properties which can be simulated by a computer.

Consider the difference between the lives of the same tone played on the piano and the violin: the piano's tone results from the hammer's striking the strings, while the violin's tone is produced by sliding a bow across the string. In the first instance, the attack is quite dynamic - the sound bursts into existence, whereupon it begins to die at once. The violin's tone, however, grows as the bow excites the string more and more, so the attack is more gradual, and the decay (relative to the tone's life as a whole) is shorter. The violin's tone is also sustained at a given amplitude as long as the bow continues to excite the string. Such tonal attributes are contained in a function which can be graphed as an envelope containing a history of the life of the tone. Each instrument has its own distinctive envelope. Typical envelopes for the piano and the violin are graphed below.
                                                                contd.

FIGURE 5:

As it turns out, however, envelopes exist not only for the tones themselves, but for each harmonic there is a distinct envelope. Therefore, if we attempt to re-create a particular instrument's sound, we must control the envelope for the fundamental and for each present harmonic, in addition to the waveform items discussed above.

COMPLEXITY

We are almost finished with the technical stuff. Only one further area remains to be covered (for this discussion, at least).

If we controlled the envelope and amplitudes of the fundamental tone and all harmonics, plus the presence, absence or muting of individual harmonics (plus other components of the waveform, such as phase differences, about which more will be said later), we could approximate the sound of, say, a vibrating string. Unfortunately, that would leave behind the violin itself, since we made no attempt at re-creating the sound of the soundbox or the rasp of the bow itself. These sounds, plus others I have neglected, have their own waveforms, harmonics and envelopes. Their fundamental tones may or may not be at the same frequency as that of the vibrating string. In short, the task of re-creating instrument sounds is quite complex.

THE COMPUTER SIMULATION

If we are able to identify all component functions of all component waveforms of an instrument's tone, there remains the task of getting the information into the computer and then generating the electrical impulses necessary to produce the sound through a loudspeaker. I will also go into this in more detail anon, but for now I will pass through the process with the quick note that it involves sampling the waveform data, forming a table of discrete points on the plot of the waveform function and feeding that table, along with the envelope's tonal history, as output to a digital-to-analog converter (DAC). In essence, that is the function performed by the music synthesizer. While I subscribe to the black box theory (let the thing do its job and don't worry about it), I will spend some time on this process later, since it involves constraints put on the computer musician by the CPU. For now, though, I'll return to the discussion of the computer musician himself/herself, and explain the "twist" to which I referred earlier.

TECHNOLOGY AND THE ARTIST

As we have seen, the computer musician must understand some of the components of sound generation in order to design the sound which the computer will synthesize. It is not necessary for him or her to actually play the music, however. After all, once the work has been done, anyone can type "RUN" at the keyboard. The computer musician must be responsible, however, for giving the computer precise instructions on the notes to play, their durations and combinations, diacritical markings, instrument definitions, assignments and orchestration. In other words, he or she must compose the music, orchestrate it, and theoretically construct the instruments to be used to play it. Given the scope of responsibilities for computer generation of music, it would not be surprising to see specialization in these areas in the future.

WHAT NEXT?

Far from trying to replace traditional music, computer-generated music has the potential to carry us to new vistas. Unlike previous instruments, the computer will require synthesis of the composer's, orchestra-orchestrater's and instrument maker's crafts into a single unit. Much of the spontaneity of live performance may be lost, but the artist can refine and hone the musical product to degrees unknown before. The computer composer can now carry the artistic image from birth to performance.

Almost all of this lies in the future. It is doubtful that this new instrument has yet found its way into the hands of the artists who can begin to realize its true artistic potential. Even if it has, not enough time has elapsed for computer musicians to begin to master the instrument, for most of art is discipline, and just as the composer must spend many years studying and playing the music of others before he or she has mastered the craft sufficiently to create masterful original works, the computer musician must learn to re-create the known before he or she can envision the desired tonal colors and proceed masterfully toward them.

The first on the block? Perhaps the innovator's path is not swept with fame, glory and respect, but to be in the forefront of a new artistic potential has another, more personal reward. Pardon me, while I put on Hobbs' Concerto #3 in M-flat major.....

(Note: This is the first of what will probably be a five or six-part series.)

Answer to Last Month's
Crossword Puzzle

A BERNOULLI UPDATE
by Lawrence Husick

When I last wrote in the journal about my experiences with the Infax 101A drive for the Apple, I said some things were in the future for the unit. Well, the future, as they say, is now.

My "little" 10 megabyte cartridge disk unit has now grown to 20 MB, and is supporting both my Apple][+ and my IBM-PC with ease. Both machines have access to the entire disk if I want them to, or to 10 MB each if I keep them separate. At the moment, they run separate cartridges due to a lack of partitioning in both environments, but there's no reason why the situation can't easily change.

Now that you've heard that, the techical questions can start...

Firstly, how is it done, what kind of network is that? The answer is SCSI, (Small Computer System Interface) which is a type of bus. See American National Standards Committee (ANSC) X3T9.2. The innards are simply a very smart parallel card and a defined set of commands understood by all SCSI (pronounced Scuzzi) devices. The commands are mostly 6 bytes long and carry out all of the common I/O functions. The format of a command is:

```
Bit --> 7  6  5  4  3  2  1  0
        ------------------------------------------------
1       ! Group Code....... ! Opcode ..............   !
2       ! Logical Unit No. ! MSB Logical Block Addr   !
3       ! Logical.....Block.....Address ...........   !
4       ! LSB .... Logical......Block.....Address..   !
5       ! Number .......of.......Blocks............   !
6       !   VU  VU  R.E.S.E.R.V.E.D   FLAG LINK       !
        ------------------------------------------------
B
Y
T
E
```

By building a specialized parallel (SCSI) interface for both the Apple and the PC, Infax has allowed both computers to "talk" to their disk unit on the same cable. In addition, each computer has been programmed to respect a busy signal sent by the other, so that they don't collide with each other.

While the SCSI protocol is useful in talking to the disk drive, the more exciting possibilities are in its use for interfacing one computer directly to another. With proper driver software, it should be possible to have the Apple send infromation directly to the bus of the IBM and vice versa. While that driver is still in the future, other devices and drivers are available today. Not only hard disks, but 9-track tape drives are using SCSI to communicate. (Imagine writing an industry-standard 9-track tape with your little Apple!)

While the cost of the Infax unit is not low ($2395 for 10 MB), the relative security of data cartridges, and the speed and reliability of the unit make it a good buy for the business user. Backup with the 20 MB unit takes seven minutes for a 10 megabyte cartridge (regardless of the amount of data on it!) while the PC's higher data rate cuts that time to four minutes.

The drive understands Apple DOS 3.3, ProDOS, Pascal, SOS, and CP/M (only SoftCard and PCPI at the moment). All software is included. The utilities allow formatting for the different operating systems and disk backup and restore, as well as several other useful functions. On the PC side, the drive uses PC-DOS 2.X and also has its own utilities.

Finally, the drive comes with something that is worth more than anything else...superior support! The people at Infax have an 800 number, and will take time to answer questions. Service, though performed at their factory and not at dealers, is quick and reliable.

Infax can be reached at 800-241-1119.

LOST

Lost forever are the words of wisdom which could have occupied this and other white space on: applications for, hints and solutions to problems with the LISA, //c and /// computers.

REWARD

Finders will be rewarded with contributions in kind.

COLOUR? WHAT COLOUR?
by Joe Moldovan

(Ed. Note: Reprinted from the June 1984 issue of Applecations, Sydney, Australia.)

I am the proud owner of an Apple][+ and the not-so-proud owner of an Apple PAL Colour Card. It is a shock to discover that although the Apple is a very good colour machine, the available colour cards do not live up to expectations - if one expects clean, stable, bright, "crawl" free colour. Eventually I had enough of looking at the "creepy" output fom my system and not wishing to spend more money than the over-priced colour card had already cost me, I set out to investigate how it could be improved.

There were two small articles in Applecations Nov/Dec 1982 which gave me a few hints for improving things. In summary they covered three areas:

A. The adjustment of the Chroma balance pots on the PAL card to minimise the subcarrier breakthrough to the composite video signal.

B. Setting the 3.58 MHz trap capacitor (C18) to an optimum value so that the output resolution does not suffer.

C. Switching the colour burst signal off during text output.

I found that all three adjustments gave a worthwhile improvement; however, I was still far from satisfied with the result. I decided to dig a little deeper into the "guts" of the colour card.

I began by trying to analyze the objectionable features of the video output. I wrote a small program that drew colour bars on the screen and had a close look at the display. Four things were immediately apparent:

1. The overall saturation and hue of the color bars was excellent as was the resolution of the pixels. (The display is a modified colour televsion set and this gives me about 10-12 MHz of bandwidth - certainly adequate for anything that the Apple can produce.)

2. There was a noticeable set of thin dark bars moving diagonally across the display. Their width suggested that they were remnants of the 3.58 MHz noise on the composite video output.

3. There was a set of pulsating "herring-bone" bars across the screen. I traced these to be interference from the motherboard and disk controller. Their effect on the picture was weak and not very annoying (and probably impossible to eliminate in a pracical way), so I decided that I could live with them for the moment.

4. Where the colour bars met there was a variable speed "chroma crawl". The speed could be varied by adjusting the colour burst oscillator trimmer (C20). This was a very annoying problem and one that, at a casual glance, gave the impression that the colour card output was unstable.

I decided to fix the problems of the 3.58 MHz noise and the chroma crawl. The following is a description of my attempts to do so. First, however, I will give a brief description of how the colour card works. If you do not understand some of the concepts involved, but would like to find out more, the best place to look is your nearest library. Any good book on PAL colour television receivers will give you the answers you need.

The composite video output of the Apple consists of an (almost) NTSC video signal - the USA television standard. This is fed into the colour card which decomposes it into three components - the sync signal, the luminance signal and the chrominance (or colour) signal. (A small diversion: The last stage of the Apple video output carefully combines these three signals and it seems a shame that the colour card has to take them apart again. We could, conceivably, connect three wires to the motherboard at appropriate spots and eliminate some of this complex signal processing. I once plugged a Taiwanese PAL card into my [genuine] Apple. This required three wires to be connected to the motherboard. The Taiwanese card cost $30. The repairs to my Apple cost considerably more!)

The colour card contains a PAL chrominance demodulator chip and a 4.433 MHz colour burst oscillator. The output of these is a restructured PAL chrominance signal. This and the unaltered luminance and sync signals are mixed to give a new PAL compatible composite video signal. The block diagram form is given in Figure A.

Both of my problems involve one part of the chrominance signal - the colour burst. After looking at the colour card circuitry and scopeing the actual card, I found the reason for the first problem (the 3.58 MHz noise). This is caused by the fact that the low pass filter (L2, C13, R18) is not very effective (and neither is the subcarrier trap [R28, L4, C19] for that matter). Some of the NTSC subcarrier still gets through to the base of the mixing transistor (Q5) and into the final output. A big improvement can be made by connecting the output of R18 to the other end of R28, thus passing the sync signal through the sub-carrier trap as well. I found that this gives acceptable results. (The complete solution involves putting a very good 3.58 MHz trap on the composite video output. This should be either a crystal or ceramic resonator obtained from the USA or a multi-stage tunable filter with a high Q.) The circuit to be modified is as shown in Figure B.

The full circuit diagram for the colour card is printed on the back of the installation pamphlet that came with it. The component numbers shown in Figure C were taken from there.

All signals, including the video signal, in the Apple are derived from the master clock oscillator which produces a signal labeled 14M. This is why the Apple video display is rock-steady and no "refresh flicks" appear when the screen is updated. (Next time you get the chance have a close look at the display from a TRS-80 or Microbee.) In the block diagram shown you will notice that the colour burst oscillator is not controlled by any other circuit - in technical terms it is free-running. Because it is a crystal oscillator its frequency will be the same as the colour-burst signal from the Apple - within a small margin of error. Unfortunately there will always be a detectable phase difference between the 14M signal on which the sync and luminance signals are based and the PAL colour burst. This is the reason for the colour crawl

contd.

in the output. In fact, the problem is not that there is a phase difference but that it is constantly changing. To solve the problem the oscillator on the card has to be phase locked with the 14M signal or one of its derivatives.

After some reading and head scratching I arrived at the conclusion that a 4.3361875 MHz cyrstal oscillator (the true PAL colour burst frequency) should be included in a phase locked loop and a reference signal should be derived from 14M by dividing it by 45 and multiplying it by 14. The actual frequency of 14M is about 14.25 MHz. Thus,

f(Ref) = (14M / 45) * 14
       = (14250000 / 45) * 14
       = 4433333 Hz

This frequency was within the colour locking range of my television set and with some trimming here and there it could be set even closer to the PAL burst frequency.

I attempted to build a simple phase locked loop and oscillator to substitute for the one on the board. At last count it contained six IC's, a $10 varactor and

almost as many other components as the whole colour card by itself!!! It does not work because the colour burst somehow gets into the output as luminance information - filling the screen with a half saturated red color. At any rate it is now obvious that the whole concept is uneconomical and that the colour card cannot be upgraded in this fashion. Some of you may be able to find a simpler way of locking the two signals together. If so, please let me know.

I have made significant improvements to the quality of the colour display on my system but it is still far from satifactory. My next step is to save up enough money for an RGB card and modify it to drive the Red, Green and Blue circuits of my television set.

I know that a large number of Apple colour cards are floating around in the user community. I had hoped that a simple fix was available to improve its poor performance, but this has turned out not to be the case. So I am afraid all of you out there who have shelled out your hard earned cash in the hope of using your Apple for colour display are stuck with a dud. As one of my visitors viewing the mess on my screen remarked, "Colour? What colour???"

Figure A

Figure B

TO FILTER 3.58 3.58 MHZ FROM
THE SYNC SIGNAL, DISCONNECT
POINT A, AND CONNECT IT TO
POINT B.
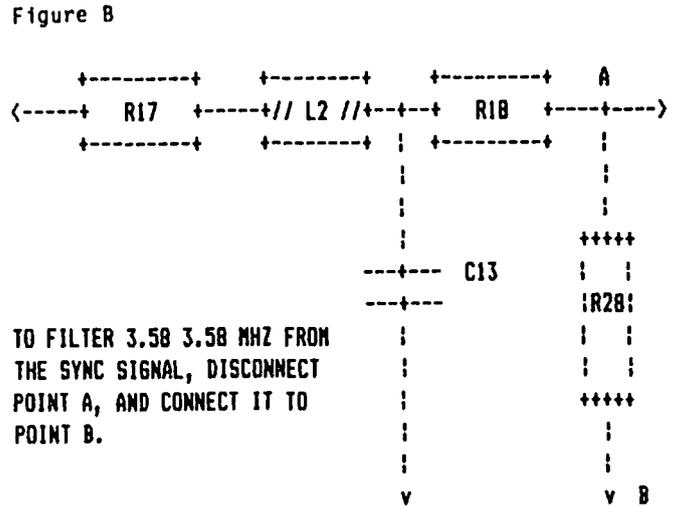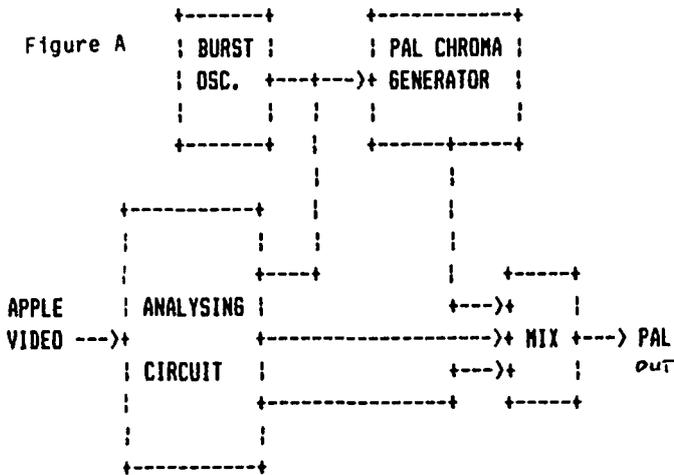
Schematic Diagram of the Apple II PAL Color Encoder Card

Figure C
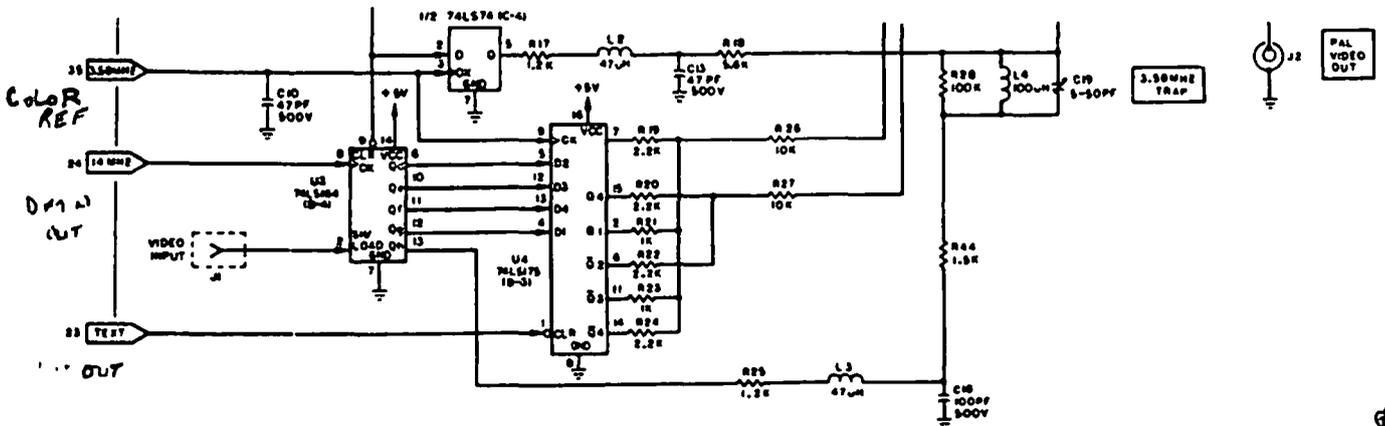
THE COMPUSERVE CONNECTION:
Accessing the Apple SIG
by Bart Cable

This discussion applies to all Apple users who want to access the Micronet Apple Users' Group (MAUG) on CompuServe. In my earlier article, "The Apple /// and CompuServe" (printed here as "Calling CompuServe with an Apple ///), I suggested that it would be possible to read messages daily on the Apple /// Section of MAUG for as little as $.25 a day. While my main interest is to rally support for the Apple ///, the shortcuts I will describe here are equally as useful for those who want to access other Sections of MAUG.

When you first subscribe to CompuServe and join MAUG, it takes a little while to become familiar with the different features available, and you will want full menu prompts. But soon you'll be able to get by without the menus, especially if you make up a reference sheet with all the necessary commands on it. I use a telecommunications package (Apple's Access ///) that allows me to record everything on disk, and I have edited and printed a copy of the various MAUG help files for easy reference.

Once you are ready to bypass the MAUG menus you can set up your options file for some shortcuts. Under the "OP" command at the function level, select the following (all commands are followed by a RETURN):

NS - to not stop between messages
BR - to set brief mode, suppressing options display
MWM - to mark messages waiting for you (as opposed to printing them out automatically)
DS7 - to set default Section for MAUG login (Section 7 here is for Apple ///)
P - to make these selections permanent

You can also select "RNS" to skip any messages you have sent or "RNT" to type these messages. If you later want to go back to receiving menus, select "MEN" and "NB" (menus and no-brief); also, "ST" will cause the system to stop between messages. Note, however, that if you have forgotten the choices in a menu and don't want to dig out your reference sheet, you can type "?" and RETURN to get a quick look at the choices. Other options are "LL" to set your terminal line length, and "PC" to change the prompt character.

Thus, you can set it up so that each time you enter MAUG you can quickly examine the latest messages in your chosen Section. If you type "RN", you can read all the new messages in that Section that you haven't seen yet. If you then want to access another Section, simply type "SS" and the Section number. "RN" will again work in the new Section. In this way you can record the messages on disk and exit without burning up a lot of connect time. Naturally you can tailor this selection of options to your own specific needs, using different combinations from the ones I have suggested here.

Shortcuts are also possible in the general CompuServe area. You can choose to reset your logon actions from the DEFALT menu on page CIS-9. To save some time you can select abbreviated prompts, once again avoiding the full menus. In addition, you can set up your own personal menu that will come up when you log on to CompuServe. With it you can list the most common areas that you visit frequently. Mine currently looks like this:

```
DEFALT Settings
+++++++++++++++

Actions When Logging On:
1 If Email waiting,  Announce it
2 You Now Receive  Short Prompts
3 Automatically run  No PROGRAM
4 First service is Personal menu
5 Create/Edit your Personal menu

1 MAUG                    PCS-51
2 MUSUS                   PCS-55
3 CREATIVE C.             PCS-22
4 Email                   EMA-1
5 Billing                 BIL-5
6 Password                CIS-175
7 DEFALT                  CIS-9
8 Top                     CIS-1
```

At logon I can go immediately to MAUG by selecting "1". But even if I want to do something else first, like change my password, I can still use a shortcut to get to MAUG when I'm done: "G MAUG" (G for "go"). To create a personal menu and select short prompts, go to page CIS-9 ("G CIS-9") and follow the instructions.

Using the Apple /// Section of MAUG as an example, all of this comes together when I want to read the latest messages in a short time. I usually do this daily, since MAUG can only hold a few hundred messages at a time (all Sections) and new ones push old ones off. A typical message may only be available for two or three days. (For important messages to individuals, you can use the E-Mail portion of CompuServe where a message remains until the person reads it.) The Apple /// Section usually contains from 3 to 10 new messages each day (sometimes more). I can get in and out fast by using 1200 baud. I set Access /// to record mode, dial up CompuServe, go to MAUG, and give the "RN" command. When all messages are read, I give the "OFF" command to log out. At 1200 baud this typically takes 1 minute, costing only $.20. At 300 baud it can be 3 to 4 minutes, costing $.30 to $.40. (These figures are based on the CompuServe rates between 6 p.m. and 8 a.m. and on weekends. The rates are higher from 8 a.m. to 6 p.m. weekdays.) Of course, if I want to send some messages, that will take more time. If I am going to compose on-line, I use 300 baud because of the extra "composition" time. But I usually create the messages ahead of time and then use Access /// to transmit them from disk at 1200 baud.

The bottom line is that MAUG provides a useful and interesting exchange of information among Apple users. The Apple /// Section has become indispensible to me, and I access it every day. Some users have complained that they cannot afford to access it that much, but with the shortcuts I've described it is possible to get in and out in minimum time. Once you get past the expense of a modem, the CompuServe Connection doesn't have to be a costly experience.

WEDNESDAY'S CHILD IS FULL OF WOE!
by Leon H. Raesly

Monday's Child is fair of face.
Tuesday's child is full of grace.
Wednesday's child is full of woe.
Thursday's child has far to go.
Friday's child is loving and giving.
Saturday's child has to work for a living.
But a child that is born on the Sabbath day,
    is fair and wise and good and gay.

                                    Anon.

Your microcomputer has surge protection, and it is safe. Right? Wrooong!! Well, you always turn it off when there are thunderstorms in the area, so you are safe, right? Wrooong! But you both use a surge protector and turn it off, so you're really safe, right? Wrooong!!

The time was early evening on a Wednesday in early August, of 84, and the setting was suburban D.C. I was on the balcony of my apartment home, about 15 miles from Silver Spring, watching a very severe thunderstorm move through the area about 5 miles south. My apartment complex is on one of the higher elevations in the area, so I had a grandstand seat to watch this storm (born on a Wednesday night) move through the region.

Now I know that YOU don't believe that storms have a personality, but this one did! And it was malevolent! Believe me, I saw it! Let me share with you what I saw happen.

The Storm (I'll call it Malev, for short) was moving very slowly, and taking it's time about everything. Since it was moving so slow, Malev could look around to decide what was the best (or worst) to do.

After several lightning strikes on houses in the area (the Fire Department was kept veeery busy that night!), Malev decided to be maliciously mischievous. He noticed that one company in that office building had three microcomputers, each protected by surge protectors, and all turned off. "Now for some fun", he said, in a deep voice (if you have ever heard a thunderstorm talk, you KNOW his voice was deep!)

After a few preliminary bolts in the area to warm up (a Wednesday's Child, you can tell!), he let loose one really strong bolt at the power line outside the office building. The lightning went charging down the wire into the building! At the main switch box, it branched into several lines, blowing a few flourescent lights on the way. Then, "Eureka!", he shouted, "I found it!". THE DEDICATED LINE TO THOSE THREE COMPUTERS! (You do have dedicated lines on all your micros, to reduce spikes from other offices, don't you?)

Into the line he poured the current, racing down to the outlets in the wall. "Oh", he thought, "a surge protector. I will just arc around that. After all, it is only a 1/4 inch to jump. Nothing for an experienced lightning bolt like me!" (None of the three surge protectors were blown, but signs of arcing were in each box.)

Now on to the switch. This was even simpler child's play to ol' Malev (Wednesday's Child's play?), since the contact's distance in a typical switch are only about an eighth of an inch apart in the open (off)

position. And now ol' Malev had full play to wreak his havoc with impunity.

The end result (observed by a very confused staff the next morning, when nothing worked right, and everything seemed very wrong!): one micro had a 128K board blown, and one RAM chip on the motherboard blown. It was dead. The second unit had the timing cycle shot, and would not access certain boards in some slots, but would access other boards in the same slot. In addition, the keyboard encoder was fried, and a q would print as a [, and other wonderful typing things of this nature.

The third was where ol' Malev was at his mischievous best (or worst, from my view). A 128K board was blown, as was a 32K board. The sneaky part was that two motherboard soldered connections were melted, leaving them in "Sometimes" contact. It would come and go in operation!

It took us three days of switching parts, boards, and booting programs, etc. just to isolate the problem in two micros, and then we were out of operation on them for another three days for repairs. (Thanks to Mark Badders of HLA Computers in Beltsville, Md. that it wasn't any longer!). Several boards had to go back to their manufacturer, and we are still waiting (three weeks later) for them to be returned. And the third machine? We finally gave up on its intermittent behavior (after we got the first two back, of course) and into the shop it went. Ol' Malev was truly a Wednesday's Child!

As I watched the storm move off that night, I could swear that I saw several of the thunderheads sneer at me. Ol' Malev got in the last word, but for the last time! Now I am prepared. Every night, and whenever we see a thunderstorm approach, we not only unplug the machines, we move the power cord five feet away from the outlet! I think that I finally have ol' Malev beat, if he ever comes back.

Thanks to Ms. Levchenko of the Aspen Hill Public Library of Montgomery County, Md. for finding the above quotation for me in "Bartlett's".   &

LISA ANALYZES INSURANCE CLAIMS
by Donald W. Kornreich

The U.S. Dept. of Transportation needs to analyze over 5,200 insurance claims against a particular transportion system from 1977 through 1983. The basic data includes the name of the person making the claim, the person's labor code, the date of the accident, the date of the claim settlement (if any), the system's current/latest estimate of the probable claim settlement value, the actual settlement value, etc. It was decided to analyze this data on a Lisa (Release 1.0 of the Operating System; one megabyte of RAM, 5 megabyte hard disk) using LisaCalc and LisaGraph. The ultimate reports were primarily graphs, backed up by selected tabular data. The calculated data would be summarized by: year of the accident, months between the accident and settlement (in 6 month increments), claims by labor category, number of claims, value of claims, claims per 200,000 direct labor hours, zero value claims, etc.

We created three LisaCalc templates involving 45,000 different formulas to do the basic summarization. Up to five files per year were created because of the LisaCalc limitation of 255 rows per file (including an allowance for rows describing column headers plus column subtotals.) Three basic templates were required for processing each file because one megabyte of RAM (including allowances for Lisa's operating system as well as LisaCalc) would not accommodate 240 records per file plus the 45,000 embedded formulas. Naturally, this increased our enthusiasm for the future Lisa virtual memory operating system (e.g. one that uses the Motorola 68020 microprocessor). The subtotals from these three basic templates were then cut and pasted into a master template (featuring another 8,000 embedded formulas) in order to develop the totals and subtotals for the 50 initial graphs (usually plotted by the year of the accident). It was very tempting to include too much data on a graph in order to reduce the number of separate graphs.

This task took a total of six weeks, including one week lost while restoring the storage capacity of the Profile (5 megabyte hard disk) and the floppy disks (860K formatted each). The reformatting was required because of the buildup of garbage that is not visible from the desktop. Do not terminate an application that is running with "Apple.", nor take a floppy disk file to the wastebasket without following it with an empty folder from the same floppy disk to the wastebasket. If you do either of these you will get garbage on your Profile and floppy disks which you can't see or delete from the desktop. (Query whether the same problems will arise with the MacFinder?)

This lost time was primarily due to pushing Lisa to its limits, thereby uncovering certain software bugs. With more practical experience, we are avoiding the problem of inadvertently keeping files on the disk that we thought were in the wastebasket. It is unfortunate that Apple does not suggest some of these procedures in the documentation for its applications.

The data base is being updated periodically, with all of the files being reprocessed through all of the templates. We are looking forward to the Lisa 2/10 upgrade which should reduce our processing time per file per LisaCalc template by half. The Lisa analysis was then used to project the number and value of future insurance claims. The Apple Lisa demonstrated its ability to accomplish successfully a large and complex task.

(Ed. Note: In a "last minute" conversation with the author, we learned that during the month of October the Lisa 2/10 hardware upgrades were initiated.) ⌘

At right is a quarter of a chessboard I made with Macpaint. If readers want to duplicate (or improve upon) the design, here's how I made it:

CHESSBOARD PATTERN   .....by Raymond Hobbs

1. Choose "EDIT PATTERN" from the GOODIES menu.
2. Edit any pattern into one copy of the pattern you wish (e.g., I made one apple).
3. Draw a box.
4. Select the paint can and fill the box with your pattern.
5. Enclose the box and select "FATBITS" from the GOODIES menu.
6. Erase every other copy of your pattern in the box.
7. Leave FATBITS.
8. Enclose the box and duplicate it vertically.
9. Now enclose both boxes and duplicate them horizontally.
10. Enclose one of the four boxes.
11. Click "INVERT" from the EDIT menu.
12. Repeat for the diagonally opposed box.
13. Now duplicate the boxes first horizontally, then vertically.
14. Make sure everything is aligned, then print your chessboard.
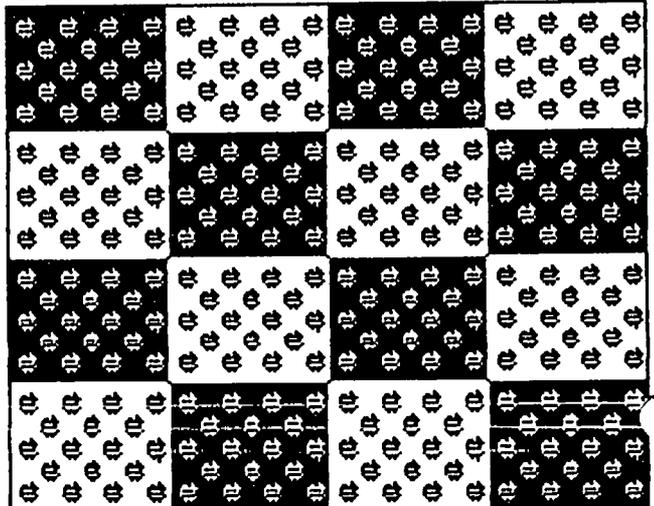
FIRE IN THE VALLEY:
A Book Review by Robert C. Platt

Fire in the Valley: the Making of the Personal Computer by Paul Freiberger and Michael Swaine, 1984 Osborne/McGraw-Hill, 288 pp.)

Of the number of recent books on the computer industry, this is one of the most enjoyable. This book stands out, not because of its exclusive focus on the microcomputer revolution, but due to its concentration on the people that created that revolution. It is well researched, well written and well balanced. These are not small feats, since other books have failed on any of these three fronts. This book carefully explains such mysteries as:

● How did user groups start?

● How did Apple Computer start and why did it succeed when others failed?

● How did the S-100 bus standard evolve?

● Who was Captain Crunch and how did he get his name?

Admittedly, these facts have little direct use outside of a Trivial Pursuit game played with fellow WAP members. But the history of the microcomputer industry (carried through the 1984 introduction of the Macintosh and PCjr) does provide insight into the role of computers in society and the role of user groups such as WAP.

The authors skillfully portray the people that made the microcomputer industry. Physical and philosophical descriptions of the key players lead the reader to relive the early conflicts in the industry. Many questions that we continue to debate today have been with us since the first days of microcomputing. For example, the debate over the role of dealers vs. mail orders; dealer support vs. manufacturer support; software piracy; and computer access to the economically disadvanted all started with the birth of the industry.

The book's coverage is excellent. It begins with the history of computing in general and an explanation of computer operations. Introductory chapters also cover the invention of the transistor, the microprocessor, and the DEC mini-computer. Next the book turns to the first personal computer, the MITS Altair, followed by the IMSAI. Other chapters turn to early computer user groups, magazines, computer shows and software companies. The history of Microsoft, Tandy/Radio Shack and Apple are also covered. The book ends with accounts of Xerox, Hewlett-Packard and IBM's entries into the industry.

I recommend the book highly. If you care about microcomputers, you will enjoy reading it.   ⌖

LETTER TO SOFTALK
October 22, 1984

Margot and Al Tommervik
Softalk Publishing
7250 Laurel Canyon Boulevard
North Hollywood, CA 91605

Dear Margot and Al,

INAPPROPRIATE TECHNOLOGY
by Jay M. Thal

I've decided to step back into the past. I'm convinced that, for certain uses, I cannot afford to spend the time using the computer and printer.

This realization came about last month as I was preparing my monthly mailing for the DisabledSIG's meeting. At that time the mailing list stood at 131 names; after the meeting I realized that it had reached 150!

So what's the problem? Doesn't word processing save time in preparing and formatting the message? Of course it does, but it is only then that the problems start.

Printing out 150 mailing labels takes about 15 minutes. Then you load in the form-feed postcard stock into the printer and the text into the computer and you're ready to run. Or should I say crawl. Each card takes about 20 seconds to print - that's 50 more minutes. You then turn the postcard stock around, type in a 10 line Basic program, and print out a return address. Those take about 10 seconds apiece - or 25 minutes.

So far I've spent and hour and a half!

Now I have to peel away the form-feed edges and separate the cards. Another 15 minutes. Attach the mailing labels - 12 minutes, and lick the stamps - another 12 minutes.

Nearly 2 1/2 hours in total and a YUKKY tongue!

Now, I'm not totally against modern technology. In fact, once I form my mailing list I can Xerox it off, on four pages of stick-on labels, in about 15 seconds. I now figure that I can mimeograph off about 150 cards in five minutes - cards that already have a stamp on them courtesy of the Post Office.

All told, I think I can save two hours a month if I give up on my computer.

DOES ANYONE HAVE A MIMEOGRAPH MACHINE?   PLEASE. ⌖

It was with considerable disappointment and sadness that we at the Washington Apple Pi learned of your need to shut down operations. We have always considered your publications of the highest quality. We had great hopes for St. Mac since our Macintosh owners are so much in need of information.

We remember the help you gave us several years ago by running for free an ad for a publication we were offering. If there is anything we can do to assist you, please feel free to call. If there is any message you would like to convey to the many WAP members who subscribed to either Softalk or St. Mac, we would gladly pass it on.

Could you tell us if you are planning to re-organize and start over or attempting to find a publisher to pick up your subscription commitments? Many of our members have asked.

Again, we hope things fair well and look forward to hearing from you.

Most Sincerely,

David Morganstein, President
Washington Apple Pi   ⌖

VIEWS AND REVIEWS
by Raymond Hobbs

Occasionally, in writing reviews of products (and also in writing about the tribulations of MAC programming), I find that I am saying to myself, "Self, you have become too jaded a critic to be useful." At such times, I need to step back from the rather hard-beaked approach a businessman normally takes to data processing, and to grab some non-business applications for my computer(s) and have some fun.

This month's review takes on two software packages and a book, and it was fun all the way through. The software is represented by two games which have been around for a little while, "Castle Wolfenstein" and "Aztec", while the book is an updated "What's Where In The Apple". To set the tone right up front, let me begin with the book.

* WHAT'S WHERE IN THE APPLE, by William F. Leubbert, (Micro Ink).

When this book first appeared on the stands, stores had a problem keeping it in stock, and with good reason - whatever mysterious things are going on in RAM and ROM are documented in What's Where. I consider the book to be an indispensable reference, not only to the programmer, but to any user who wants to find out why things work the way they do, or how to make small changes in programs to make them work better.

The new What's Where is divided into three parts. Part 1 is a programmer's guide, an overview of memory organization in the Apple, and how it is used. This part is greatly expanded over the original What's Where, and is chock-full of useful system information on how to get the most out of Parts 2 and 3, which is where all the detail is located. Parts 2 and 3 provide two different looks at memory itself. Part 2 is a memory location-indexed atlas of the Apple's memory, while Part 3 is an atlas of memory by routine or parameter name (listed alphabetically). All of the Apple][models (][,][+ and //e) are covered. So if you know the routine name and just want to find out where to access it, you use Part 3. If you know that something is happening in RAM at some location on page three, you use Part 2 to find out what it is. If you know what you want to do, but don't know what the routine is called or where it is located, chances are that you can find the information in Part 1 to steer you to the right place in Part 2 or Part 3.

But that's not all. Armed with a couple of hours and some 3 x 5 cards, you can browse the atlas and jot down useful locations for error trapping, reset disabling, DOS modifications, even monitor graphics handling (Part 1 even shows you how to incorporate ROM routines into your BASIC programs). You can make your own PEEK and POKE chart a la Beagle Bros.! Addresses are given in both decimal and hex, so What's Where is a ready reference for both Assembly Language and BASIC.

I think that as a reference, What's Where is a must. But I also believe that it's more than that. What's Where takes us into those parts of the Apple that a lot of us forgot about when we became sophisticated users of business packages and integrated software. You know, the Apple wasn't designed as a business machine. No, siree, it was made for kids - both normal sized and overgrown - who wanted a computer to have fun with. What's Where takes us back to that.

* CASTLE WOLFENSTEIN, by Silas Warner, (Muse Software).

This game has been on the market for quite a while now, and it still sells very well. It is an adventure game, in the sense that it has a beginning and an end, with a set task to accomplish in between, but it is not a text-type adventure. When you complete the task, you "win" the game; however, the game will become more difficult on the next iteration.

The scenario is this: during World War II, you have been captured behind German lines, and are presently incarcerated in Gestapo headquarters at Castle Wolfenstein, awaiting interrogation. A fellow prisoner has managed to steal a pistol with a full clip (10 bullets), which he gives to you just before he is dragged away to his fate by the armed guards. Your mission (should you choose to accept it) is to escape from the castle, which is a multi-level, multi-room matrix. The rooms are, of course, liberally supplied with guards and SS stormtroopers, waiting to recapture you or blow you away. In trying to escape, you are aided only by whatever supplies you can strip off of guards you shoot (or who surrender), or find in locked chests located throughout the castle. Supplies are bullets, grenades, uniforms and bulletproof vests. There are also ample supplies of food, drink and other paraphernalia which can be taken, but provide no assistance in escaping the castle. You can either shoot your way through the hordes of guards and SS stormtroopers or try to sneak past them. To make matters more interesting, somewhere in the umpteen rooms in the castle are hidden secret battle plans for "Operation Rheingold" which, of course, the allied high command would love to get its hands on, if you can spirit them away.

The action in Wolfenstein is quick and snappy - control is effected by keyboard, paddles or joystick. Games can be saved and resumed, and there is no score kept. The graphics are easy to follow, colorful and unobtrusive - they do not interfere with the playing of the game, unless you are given to running into walls (the screen will break up momentarily if you do this). This game, one of 3 that I have bought, has proved interesting and absorbing after literally hundreds of iterations. A classic.

* AZTEC, by Paul Stephenson, (Datamost).

Aztec is the same generic type of arcade/adventure game as Wolfenstein (I got hooked on this type of game), set in the jungles of Central or South America, with an Indiana Jones type of character roaming through a rather roomy multi-level pyramid/temple in search of a golden idol. The pyramid has been visited before by a Professor Von Foerster, who left behind numerous booby-traps and loathsome creatures to beset the unwary. Aztec's spiders, snakes, scorpions and other creepies are easily the equal of Wolfenstein's baddies. As you wander through the pyramid, you may supply yourself with extra strength by using an elixir, as well as stock up on such necessities as pistols, machetes, dynamite and bullets. These items may be located among rubble piles or in baskets scattered throughout the pyramid (you may also find Professor Von Foerster himself!).

I found it much more difficult to maneuver in Aztec than in Wolfenstein, due partly to the lack of paddle

Disks from Washington Apple Pi's Disketeria are available for purchase. This form is only for ordering disks that you want mailed to you.

5 1/4" DISKETTES: - Members $ 5.00 ea.; Non-members $ 8.00 ea.) Plus $1.00 ea. postage
3 1/2"   "   - Members $ 7.00 ea.; Non-members $ 10.00 ea.) up to a maximum of $5.00.

DOS 3.2
( ) Volume 1 Utilities I
( ) Volume 2 Utilities II
( ) Volume 3 Games I
( ) Volume 4 Games II
( ) Volume 5 Games III
( ) Volume 6 Games IV
( ) Volume 7 Games V
( ) Volume 8 Utilities III
( ) Volume 9 Educational I
( ) Volume 10 Math/Science
( ) Volume 11 Graphics I
( ) Volume 12 Games VI
( ) Volume 13 Games
( ) Volume 14 IAC Utilities IV
( ) Volume 15 Games VII
( ) Volume 16 Utilities V
( ) Volume 17 Graphics II
( ) Volume 18 Educational II
( ) Volume 20 Music
( ) Volume 21 Apple Orchard
( ) Volume 22 Utilities VI
( ) Volume 23 Games VIII
( ) Volume 24 Games IX
( ) Volume 25 Utilities VII
( ) Volume 26 Stocks/Investments
( ) Volume 27 Math
( ) Volume 28 Planetfinder
( ) Volume 29 Utilities VIII
( ) Volume 30 Games X
( ) Volume 31 Plot Utilities
( ) Volume 32 Games XI
( ) Volume 33 Accounting
( ) Volume 34 Solar Tutor
( ) Volume 35 Garden Management
( ) Volume 36 Games XII
( ) Volume 37 Utilities IX
( ) Volume 38 Games XIII
( ) Volume 39 IAC VII
( ) Volume 40 IAC VIII

DOS 3.3
( ) Volume 41 IAC 25 Mach. Lang. Util.
( ) Volume 42 One Key DOS ***
( ) Volume 43 IAC 29 Utilities H
( ) Volume 44 Utilities I
( ) Volume 45 Diversi-Copy ***
( ) Volume 70 Business/Math/Statistics
( ) Volume 71 Music
( ) Volume 72 Keyboard Games
( ) Volume 73 Text Adventure Games
( ) Volume 74 Paddle Games
( ) Volume 90 Spreadsheet C Genl. Bus.
( ) Volume 91 Spreadsheet D Investment
( ) Volume 92 Spreadsheet E Bus. Recd.
( ) Volume 93 VisiPlot and VisiTrend
( ) Volume 100 Utilities A
( ) Volume 101 Utilities B
( ) Volume 102 Games A
( ) Volume 103 Merry Christmas
( ) Volume 104 Business A
( ) Volume 105 FIG-FORTH/Utilities

DOS 3.3 contd.
( ) Volume 106 Science Engineering
( ) Volume 107 Games B
( ) Volume 108 IAC 10 (Graphics)
( ) Volume 109 IAC 11 (Applesoft Tutorial)
( ) Volume 110 Personal/Education
( ) Volume 111 Games C
( ) Volume 112 Utilities C
( ) Volume 113 Business B
( ) Volume 115 IAC 12/13 Misc.
( ) Volume 116 IAC 14 MicromodemII
( ) Volume 117 Picture Packer
( ) Volume 118 Utilities D
( ) Volume 119 IAC 15 Misc.
( ) Volume 120 IAC 16 Misc.
( ) Volume 121 WAPABBS 1.1 Doc. **
( ) Volume 122 IAC 17 Misc.
( ) Volume 123 French Vocabulary
( ) Volume 124 Utilities E
( ) Volume 125 IAC 18 Misc.
( ) Volume 126 Sights and Sounds
( ) Volume 127 Math/Science
( ) Volume 128 Games D
( ) Volume 129 GLAQ
( ) Volume 130 Diversi-DOS ***
( ) Volume 131 Personal/Educ. 2
( ) Volume 132 IAC 19 - Utilities F
( ) Volume 133 IAC 20 - Pascal & DOS 3.3
( ) Volume 134 New Members Disk
( ) Volume 135 WAPABBS 1.1 Disk 1 **
( ) Volume 136 WAPABBS 1.1 Disk 2 **
( ) Volume 137 IAC 21 Spreadsheet A
( ) Volume 138 IAC 23 Utilities G
( ) Volume 139 IAC 24 Education 3
( ) Volume 140 Education 4
( ) Volume 141 Special Data Bases
( ) Volume 142 IAC 28 Pinball Games
( ) Volume 143 Sports
( ) Volume 144 IAC 27 Applesoft Prog.
( ) Volume 145 Apple Logo Tool Kit
( ) Volume 146 Apple Logo Sample Prog.
( ) Volume 147 Logo Documentation
( ) Volume 150 EDSIG1 (Elem. Math)
( ) Volume 151 1983 Tax Template
( ) Volume 152 IAC 31 Miscellaneous
( ) Volume 153 Investments A
( ) Volume 154 Investments B
( ) Volume 155 IAC 33 Miscellaneous
( ) Volume 156 IAC 35 Applesoft-AW//e
( ) Volume 157 IAC 36 Arcade Games
( ) Volume 158 Apple Logo Programs

Eamon Series
( ) Volume 180 Dungeon Designer
( ) Volume 181 Beginners Cave
( )*Volume 182 Lair of Minotaur
( )*Volume 183 Cave of the Mind
( )*Volume 184 Zyphur Riverventure
( )*Volume 185 Castle of Doom
( )*Volume 186 Death Star
( )*Volume 187 Devil's Tomb
( )*Volume 188 Caves of Treas.Isl.

Eamon contd.
( )*Volume 189 Furioso
( )*Volume 190 The Magic Kingdom
( )*Volume 191 The Tomb of Molinar
( )*Volume 192 Lost Isl. of Apple
( )*Volume 193 Abductor's Quarters
( )*Volume 194 Quest for Trezore
( )*Volume 195 Underground City
( )*Volume 196 Merlin's Castle
( )*Volume 197 Horgrath Castle
( )*Volume 198 Deathtrap
( )*Volume 199 The Black Death
( )*Volume 200 The Temple of Ngurct
( )*Volume 201 Black Mountain
( )*Volume 202 Nuclear Nightmare
( )*Volume 203 Feast of Carroll
( )*Volume 204 The Master's Dungeon
( )*Volume 205 The Crystal Mountain
( )*Volume 206 The Lost Adventure
( )*Volume 207 The Manxome Foe

Pascal   (See also Volume 133)
( ) Volume 300 PIG0: ATTACH 1.1/BIOS
( ) Volume 301 PIG1:
( ) Volume 302 PIG2:
( ) Volume 303 PIG3:
( ) Volume 304 PIG4:
( ) Volume 305 PIG5:
( ) Volume 306 PIG6:
( ) Volume 307 PIG7:
( ) Volume 308 PIG8:
( ) Volume 309 PIG9:
( ) Volume 310 PIG10:
( ) Volume 311 PIG11:
( ) Volume 312 PIG12:
( ) Volume 313 PIG13: Guerilla Guide
( ) Volume 314 PIG14:

CP/M
( ) Volume 401
( ) Volume 402
( ) Volume 403
( ) Volume 404
( ) Volume 405
( ) Volume 406 ZCPR2 Install
( ) Volume 407 ZCPR2 Documentation
( ) Volume 408 ZCPR2 Utilities
( ) Volume 409 Modem 730

Forth   (See also Volume 105)
( ) Volume 700 Assembler/Disassembler
( ) Volume 701 Full Screen Editor
( ) Volume 702 GoForth Tutorial

Macintosh - @$7.00 (see above)
( ) SigMac Disk 1 MS-BASIC Pgms
( ) SigMac Disk 2 Atkinson's Goodies
( ) SigMac Disk 3 Fonts
( ) SigMac Disk 4 MS-BASIC Pgms
( ) SigMac Disk 5 Desk Accessories
( ) SigMac Disk 6 Mac Paintings
( ) SigMac Disk 7

* Volume 181 required with these disks.   ** Vols. 121, 135, 136 must be purchased together.
*** Use of this disk requires sending money to the author ($30 for Diversi-disks and $9 for One Key Dos.)

(NOTE: ALLOW 3 TO 4 WEEKS FOR MAILING.)   Total Order ▪ ___ disks.; postage $___ ; Total amount enclosed $

NAME   _____

ADDRESS   _____

CITY, STATE ZIP _____

TELEPHONE   _____ WAP MEMBERSHIP NO._____

Make check payable and send to:

Washington Apple Pi, Ltd.
Attn. Disketeria
8227 Woodmont Avenue, Suite 201
Bethesda, MD 20814   DATE _____

WAP TUTORIAL REGISTRATION

The following four WAP tutorials are being offered on Tuesday evenings from 7:30 to 9:00 PM, at the office, 8227 Woodmont Avenue, Room 202, Bethesda, MD. (The tutorials start promptly at 7:30; if you bring your computer please arrive 15 minutes early to set up.) You may sign up for any or all of the series. They are designed for the "beginner" and will be repeated monthly. A detailed outline of the tutorials was given in the September 1984 WAP Journal.

( )   November 6   - INTRODUCTION TO APPLE COMPUTER HARDWARE   ( ) December 4
( )   November 13 - HOW TO USE YOUR APPLE SOFTWARE   ( ) December 11
( )   November 20 - BEGINNING APPLESOFT BASIC   ( ) December 18
( )   November 27 - INTERMEDIATE APPLESOFT BASIC   ( )

The fee for each tutorial is $10.00 with an Apple, monitor and disk drive, $15.00 without (monitors available for 1st 5 registrants - call office). Please note that WAP does not have equipment for you to use; if you do not bring your own, you will have to look over someone's shoulder.

___ Tutorials at $10.00 (with equipment)
___ Tutorials at $15.00 (without equipment)

Please check the desired tutorials and return this form with fee(s) made payable to Washington Apple Pi, Ltd. to:

Washington Apple Pi, Ltd.
Attn. Tutorials
8227 Woodmont Avenue, Suite 201
Bethesda, MD 20814

Name _____

Daytime Phone _____   Evening Phone _____   Total Enclosed $ _____

WASHINGTON APPLE PI, LTD.
8227 Woodmont Avenue, Suite 201
Bethesda, MD 20814

ADDRESS CORRECTION REQUESTED