THE IDEA PLACE

CLICK ART · FULLPAINT · SOLO · MACSPELL · RAGTIME · MICROSOFT WORD · SUPERPAINT · PAINTCUTTER · MACDRAFT · RAGTIME · MACWRITE · DESKTOP PUBLISHING · ARTISTO · MACDRAW · MACPAINT · MACLIGHTNING · PAGEMAKER · PORTFOLIO · READY SET GO · CRICKET DRAW

First Step:

I need another computer. Business is good; it is time to expand.

I've decided it's time for the children to learn about computers, what do I do?

My printer ate one too many labels now what?

This program just needs more memory; should I stay with Apple?

These graphs are just not good enough; wonder if there's a better program available?

This has never happened before. I didn't change anything!

What is an Apple III?

If I could sell my old computer, I could get a new Apple...

Jerry

   Becky

     Gerald

      Sharon

     Richard

   Lynn

Steve

  Betty

    Ken

     Doug

   Jay

Second Step:

Who's got the best prices? A good warranty?

What are the schools using? Where should I start?

Wonder if I could take this roller out? If I take it in, they'll charge me an arm and a leg.
I've heard the 3rd party upgrades are better than Apples and less money.

Who would know which is best for my needs? I don't have time to try... I need the right one now!

Who do I call? Will they know what I'm talking about?

Is there any software available?

What would I ask for it? Who would buy it?

Third Step:

How is the warranty backed? Six months. Apple Care sounds good.

Do you work with schools? Yes, 28 districts. Well mannered installers.

Most labor charges: $10.00 to $25.00 You can watch too.

Apple upgrades for less, no labor fee for installations. If it's not broke no; cost.
Those folks sure helped me out before. They know their stuff! Ask anyone.

They support HAAUG, local ball clubs, and stand behind their people and products.
If it is worth using, its worth stocking.

Have you considerd a trade-in or upgrade?

Williams Computer Center
25770 I-45 North
Spring, Texas 77386 367-1600
Texas Talker BBS
367-8206 500+ Members
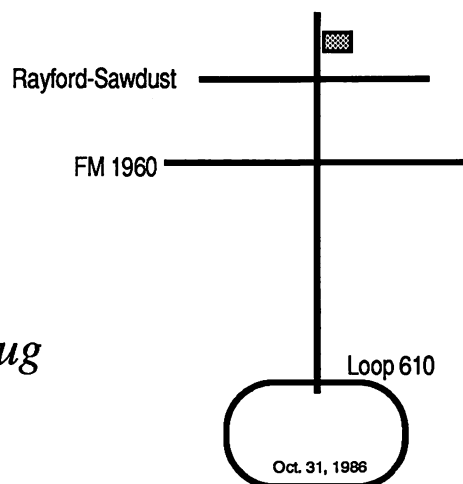
Rayford-Sawdust

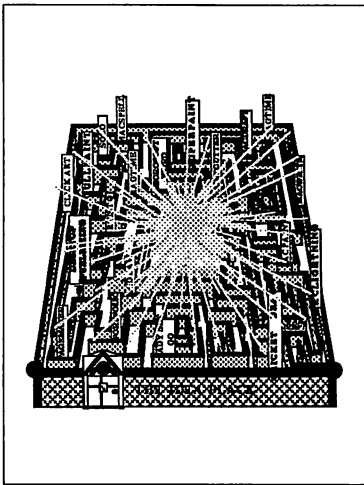FM 1960

Loop 610

Oct. 31, 1986

Apple LaserWriter $3895.00

"7" different hard drives.

Office networks, Cad-Cam for the Macintosh. $719.00

Houston Instrument plotters $629.10-$4925.00.

AppleWorks classes weekly.

About the Cover

Billy Jacobs used *MacDraw* to create this months' cover art. Containing over 800 individual objects and consuming 8 percent of a Macintosh Plus's memory, it has two objects that are repeated. Can you find them?

Editors

Tom Engle
Chris Flick

Contents

Volume 9 Number 5   September/October 1986

Advertisers Index

Overview

The Houston Area Apple Users Group provides a forum for the exchange of ideas and information about Apple Computer products and their associated hardware and software. Meetings, newsletters, and special interest groups are just a few of the ways this forum is maintained. Below are listings of the club officials and descriptions of the groups' functions.

Club Officials

President
Robin Cox   778-1635
First Vice President
Tom Dillon   376-6502
Second Vice President
Michael Conway   495-2292
Secretary
John Marek   782-3898
Treasurer
Jack Cowart   467-4215
Directors
Steve Bass   847-4407
Jeanne Boucher   723-9519
Dick Fairman   723-9619
Mike Kramer   358-6687
Dick Lee   821-1298
Bill Muhlhausen   668-3963
Senior Advisor
Rudge Allen   622-3979
Membership
Neal Scott   890-0532
Apple Barrel **Editors**
Tom Engle   774-9055
Chris Flick   363-3153
Publicity Coordinator
Dennis Bilbe   HotLine
IAC Representative
Ina Levinson   270-0137
Vendor Coordinator
Jeanne Boucher   723-9519
Program Coordinators
Mike Kramer   358-6687
Steve Bass   847-4407

The club officials meet on the first Thursday of the month at the Marriott Astrodome Hotel at 7 pm. These meetings are open to the membership. Members interested in the inner workings of the group are encouraged to attend.

Membership Information

A one-year membership to HAAUG is available for thirty dollars and includes the Starter Kit, a subscription to the *Apple Barrel,* and a membership card giving access to all the club's services. Memberships are available at the Answer Desk during the meeting, through brochures located at local dealers, or by calling the HAAUG HotLine and having a brochure mailed to you. Renewals are twenty dollars per year.

Meeting Dates

The HAAUG monthly meetings take place at the Marriott-Astrodome Hotel, 2100 South Braeswood at Greenbriar. The meetings are scheduled for the third Saturday of the month and begin at 11 am. All new members should attend the New Member Orientation.

Meeting Schedule

Time	Event
11:00	New Member Orientation
12:00	HAAUG Club Business
12:30	Main Presentations
1:30	Special Interest Groups
2:30	Special Interest Groups
3:30	Special Interest Groups

HAAUG HotLine (713) 522-2179

The HAAUG Hot Line provides a means for **members** to learn of meeting topics, news, etc, and to obtain answers to computer-related questions. If you want a return call, leave your name, telephone number, and membership number. If the question pertains to a particular commercial product, please check the list of volunteer specialists in this issue.

Special Interest Groups

SIGs focus on one topic of common interest during these meetings within a meeting. Meeting locations and times are contained in the meeting handout.

SIG Leaders

Apple // SIG Coordinator
Tom Dillon   376-6502

Appleworks
John Slack   491-1747
Advanced Topics
Bill Hensley   980-4993
Jim Record   353-1119
Basic Programming-Advanced
Richard Lemay   749-3135
Best of the Public Domain
Robin Cox   778-1635
Education
Ann Petrillo   489-7535
Games
Thomas Carson   861-5425
John Newell   531-9139
Green Apples
Richard Goss   463-0640
Hardware
Dave Simek
Home Use
Mike Stoops
David Jaschke   461-2450

Mac SIG Coordinator
Steve Bass   847-4407

Communications
Mike Conway   495-2292
Desktop Publishing
Cleland Early   941-7247
Tom Engle   774-9055
Developers
John Pence   TBA
Games
Bob Lewis   242-0990
Helix
Larry Stage   358-1105
Omnis
Buddy Jacks   299-1555
Programming
Mike Martin   333-6405
Tips & Techniques
Steve Bass   847-4407

Public Domain Software Libraries

The Public Domain Libraries are divided into two sections: Macintosh and Apple //.

Macintosh Library

The Macintosh library contains over ninety disks of public domain software. Copies may be purchased at the meeting for five dollars apiece. In addition, there are two directory disks which may be copied for free. These disks contain the listings of what is on the other disks. Also, the latest disk submissions are available for copying.

Macintosh Librarian
Larry Saphier   496-0507

Apple // Library

The Apple // library may be copied at the meeting for one dollar per side. Books are available at the meeting describing what is available in the library.

Apple // Librarian
John Marek   782-3898

Hardcopy Library

The Hardcopy Library maintains an archive of back issues of the *Apple Barrel*, issues of other magazines, and other paper material. It is open to the membership and is available at the main meeting. At present, no provisions for loaning the material have been made.

Hardcopy Librarian
Les Brown   795-2741

"HAAUG Heaven" Bulletin Board System

HAAUG Heaven is HAAUG's computerized Bulletin Board System. The BBS's telephone number is (713) 664-9873 and operates 24 hours a day. The recommended protocal setting is 8-N-1-Full.

Writer's Guidelines

The HAAUG *Apple Barrel* exists to provide the membership with timely information about their Apple computers. We at the *Barrel* hope that readers will write articles on subjects such as reviews, computer techniques, and casual observations of computerdom. For those of you interested in writing for the *Barrel* (hopefully most of you), we have established helpful guidelines to follow when writing and submitting articles.

1. Try to include some form of graphic or table which helps to explain points made in the article. The saying that a picture is worth a thousand words certainly applies here.

2. For those including graphics and tables, please make references to them within your text. This will help both the reader understand what you are saying and assist us in the paste-up of the article.

3. Include references to other sources of information if possible. By doing so, the reader is directed to other useful material beyond that of the actual text.

4. For Apple // users, we will accept text created either in *AppleWriter* or in the *AppleWorks* word processor. We prefer ProDOS, but can handle DOS 3.3 as well.

5. For Macintosh users, we accept text created in either *MacWrite* or *Word*. Please do not include graphics within the actual text. Graphics accompanying the articles should be saved as *MacPaint*, *MacDraw*, or *Chart* files.

6. Articles should be provided on disk as well as printed hardcopy. The hardcopy is primarily used to allow us to see where you as the author place emphasis. (A backup copy for yourself would be advisable)

7. Article submissions can be given to either of the editors or left at the Apple Answer Desk. Include both your name and phone number with the article to help us communicate with you if problems arise.

We hope that these guidelines will not only help you but also make it easier for us to provide a more consistent and classy newsletter.

Advertising

Rates

Inside Covers	$100
Full Page	$80
Half Page	$60
Third Page	$40
Business Card	$15
Other	$5 per column inch

Ads are to be prepaid, camera-ready, and received at least one week before the first of the month in which the ad is to appear. Contact either of the Editors for further information.

NEXT MONTH

- Cleland Early returns with a new set of Random Thoughts and a review of *Wizardry* for the Macintosh

- William Linsley describes What the User Gets

- Arthur S. Cranston II, delves further into the workings of the II in BASIC Under ProDOS

- Rick Light takes a look at compilers and related software

- Plus the usual SIG News and Specialists columns.

At the Expo

Just having returned from the October MacWorld Exposition in Dallas, I wanted to mention a few things that I saw. I could have written an editorial with a single theme. However, with so much happening, a departure into randomness might be tolerated. Although several writers are much better than I at general discussions, I felt I could take the soubriquet for this month of "Entropy Poster Child."

Big Screens on Small Macs

Of the many new products at the show, the full page display screens were most in evidence. Almost everyone exhibiting had one of these jewels capable of displaying a complete page or two of Macintosh graphics. It's amazing how nice it is to work with *PageMaker* or *MacDraw* and to ignore the scrollbars. Sheer pleasure!

There were three different displays being used. First, the E-Machine from E-Machines Inc. was at the majority of the booths. This monitor can display almost two pages of Macintosh graphics with the same resolution as the Mac's own internal display. One nice example was the ability to work in *PageMaker* on two pages at once. The term "page setup" brought on a whole new meaning. Unfortunately, you forfeit the Macintosh's display when using the E-Machine — although with healthy returns on that loss.

The MegaScreen monitor from MicroGraphic Images Corporation was another of these large two page monitors. With a 19" screen, this display also provides a nice field of view although at the loss of the Mac's own display.

The last of the screens will probably be getting the most publicity owing more to its software and hardware creators than to its software and hardware. Although Radius' Full Page Display or FPD only displays a single page of graphics, it was created in part by both Andy Hertzfeld and Burrell Smith of Macintosh fame. Andy Hertzfeld, the programmer guru and creator of *Switcher*, *ThunderScan* software, and much of the Macintosh's own code, wrote the software that gives the FPD its nice features. Burrell Smith, another member of the original Macintosh development team and a major figure in the creation of the Mac's own hardware, designed the actual display. The Radius Full Page display allows the user to use both Mac and FPD screens simultaneously (keeping a page on the FPD and open DA's on the Macintosh's own screen), to change both cursor and text size, and to get several other advantages. (See *Mac-User*, November 1986)

I'll go this one *Solo*

A new release into the desktop publishing market was shown at the show. Called *Solo*, this program integrates features of *PageMaker*, *MacDraw*, *MacPaint*, *MacWrite*, the *Desktop* and other programs not yet written. This single program should be a welcome replacement for the numerous programs that the user normally has to purchase and try to use while at the same time keeping a perspective on the complete document. Besides what you would expect of such a super page set up program, *Solo* allows the user to place a graphic of a random outline onto the page and flow text around it conforming to that outline (WOW!), kern the document, send PostScript commands directly to the printer, and many other features too numerous to mention here. I hope the programmer much luck with his product; priced at $300, this should be a real winner.

Portable Mac

Probably the biggest hit at the show came from the booth of a fellow HAAUG member. The product is a portable Macintosh with every feature you might want. Sold as the Dynamac, the top-of-the-line machine sports 4 megabytes of RAM, an internal 800K drive, an internal 1200 Baud modem, a 40 megabyte internal hard disk, an electroluminescent display screen with the same quality as the Mac's own screen, and a set of connectors that would confuse a cable salesman. Interestingly, there is even a port for an E-Machine to be attached directly.

For those hoping to purchase one, it would probably be wise to check your bank balance. The top-of-the-line model runs $7000 but for those needing a Mac on the run, this is the machine for you.

Not at the Expo

Although not related to the Expo, this message is to everyone in HAAUG. Remember to read the club bylaws in this issue and get ready for yet another election next month. Enjoy the rest of the issue.

— CHRIS FLICK

Houston Area Apple Users Group Bylaws
November 24, 1986

Editor's Note: The following updated bylaws are to be voted upon at the January meeting pursuant to Article 11.

ARTICLE 1 - NAME

The name of the organization is the HOUSTON AREA APPLE USERS GROUP, INC., hereinafter referred to as HAAUG. Its official publication is the *Apple Barrel.*

ARTICLE 2 - PURPOSES

The objectives of HAAUG are:

A. To provide a forum for the dissemination of information concerning Apple© computers and related products.

B. To share the knowledge of members at large concerning their talents and information pertaining to computers and Apple Computers specifically.

ARTICLE 3 - BASIC POLICIES

The name of the organization or the name of any elected or appointed officers or directors of the organization in their official capacities shall not be used in connection with a commercial activity or with any partisan interest or for any purpose not appropriately related to promotion of the purposes of the organization.

The copying of any copyrighted material at any HAAUG function is strictly prohibited.

No part of the net income of the organization shall inure to the benefit of or be distributable to its members, directors, officers, or any private persons, except that the organization shall be authorized and empowered to pay reasonable compensation for services rendered and to make payments and distributions in furtherance of the purposes set forth in ARTICLE 2. No member of the organization shall be financially interested, directly or indirectly, in any agreement related to the operations conducted by the organization for compensation, unless the fact of such interest be known to the Board of Directors.

ARTICLE 4 - MEMBERSHIP

Section 1. Classes of Membership. There shall be two classes of membership: Household and Institutional. Members of each class shall have equal voting powers. Any member of the immediate family holding household membership shall be entitled to all privileges of membership, except the household membership is entitled to a single *Apple Barrel* and a single vote. Institutional members shall exercise the privileges of membership through a designated individual who shall receive the *Apple Barrel* and cast a single vote.

Section 2. Eligibility. Membership in the organization shall be available without regard to race, age, color, creed, gender, or national origin. Any individual who subscribes to the purposes and basic policies of the organization may become a member of the organization.

Section 3. Dues. The annual dues for membership shall be established by the Board of Directors.

Section 4. Limitation of Authority. No member shall act in the name of the organization or take any action which would bind the organization except as otherwise specified in these Bylaws.

Section 5. Expulsion. A member may be expelled from membership in the organization for (a) failure to pay the annual dues for more than 30 days after the due date, or (b) engaging in activities prohibited by the Bylaws or any rule or practice adopted by the Board of Directors. Before any member may be expelled for engaging the prohibited activities, charges specifying the alleged cause shall be served by the President on the member charged, and he shall be given the opportunity to correct such conduct. If such member continues to engage in prohibited activities, he may be expelled at any meeting of the Board of Directors by a majority of two-thirds of the Directors attending such meeting.

ARTICLE 5 - CAPITAL STOCK

No capital stock will be issued.

ARTICLE 6 - MEMBERSHIP MEETINGS

Section 1. Regular Meetings. Regular Membership meeting shall be held monthly as determined by the Board of Directors. Notice of these meetings, stating the date, time and place shall be published in the *Apple Barrel.*

Section 2. Special Meetings. Special Membership meetings may be called by: the President, the Board of Directors, or any signed petition of fifty members. Notice of the Special meeting shall be published in the *Apple Barrel* or by a mailing to all members and shall state the purpose of the meeting. No business other than that stated in the notice shall be transacted at a Special meeting.

Section 3. Quorum. Eighty members in good standing must be present at a Membership meeting in order to conduct business. Members in good

standing shall be those members who abide by the organization's Bylaws and whose dues are current as defined in ARTICLE 4, Section 5.

ARTICLE 7 - BOARD OF DIRECTORS

Section 1. Powers. The business and affairs of HAAUG shall be managed by its Board of Directors, which may exercise all powers of the organization and perform all lawful acts which are not in violation of these Bylaws.

Section 2. Composition. The Board of Directors shall consist of a maximum of 15 members: The President, First Vice-President, Second Vice-President, Secretary, Immediate Past-President, Senior Advisor, six Directors-at-Large, and the Chairman of the Membership Committee, the Treasurer, and the Editor of the *Apple Barrel* if not elected as Directors-at-Large. The Directors-at-Large shall be elected annually pursuant to ARTICLE 8. If the Immediate Past-President is not eligible for office or is holding another elected office, then the position shall remain vacant.

Section 3. Term. Directors and Officers shall serve for a one-year term beginning on the first Regular meeting in January, or until their successors have been duly elected.

Section 4. Removal. A Director or Officer may be removed at any time, without cause, either at a Regular Membership meeting or at a Special Membership meeting by a two-thirds vote of the members present.

Section 5. Vacancies. Upon the resignation or removal of a Director or Officer, the Board of Directors shall, by a majority vote, appoint a member to serve for the remainder of the term.

Section 6. Regular Meetings. Regular meetings of the Board of Directors shall be held monthly, unless otherwise determined by the Board of Directors. Regular meetings shall be held at such time and place as may be determined by the Board of Directors without further notice.

Section 7. Special Meetings. Special meetings of the Board of Directors may be called by the President, a Vice-President or three of the Directors. Seven days' notice shall be given to each member of the Board stating the purpose of the Special meeting, and no business may be conducted other than that stated in such notice.

Section 8. Conduct of Meetings. The President shall preside at meetings of the Board of Directors. A majority of the Directors currently in office shall constitute a quorum. Meetings of the Board of Directors shall be open to all members, except that upon majority vote, the Board of Directors may enter executive session to transact business which the Board determines to be confidential.

ARTICLE 8 - ELECTION OF OFFICERS AND DIRECTORS

Section 1. Nominations. Nominations for President, First Vice-President, Second Vice-President, Secretary, and Directors-at-Large must be received at the organization's address prior to the November Membership meeting or made from the floor during the November meeting. A Nominating Committee shall be appointed by the Board of Directors for the purpose of nominating candidates to the general membership. All candidates will be announced at the November meeting.

Section 2. Eligibility. Only members in good standing may be candidates for elective office. Any dispute regarding the eligibility of a candidate shall be resolved by the Board of Directors.

Section 3. Balloting. An election shall be completed during the month of December. Only members in good standing may vote in the election. A ballot shall be mailed or hand delivered to each member in good standing. Each member may cast one vote for each office, except that up to six votes may be cast for Directors-at-Large. Prior to each annual election, the Board of Directors shall establish a deadline for the receipt of ballots.

Section 4. Results. The six eligible candidates for Director-at-Large with the greatest number of votes shall be elected. The election tellers, as appointed by the Board of Directors, shall certify the results of the election to the general membership at the first Regular meeting in January and in the *Apple Barrel*. In the event of a tie, the winner shall be determined by lot.

ARTICLE 9 - OFFICERS

Section 1. Designation. The officers of the organization shall be the President, First Vice-President, Second Vice-President, Secretary and Treasurer. They shall hold office for one year or until their successors have been elected pursuant to ARTICLE 8.

Section 2. President. The President shall be the Executive Officer of the organization, and shall have general responsibility for the business affairs and property of the organization and shall have general supervision over the other officers. The President shall preside at all Membership and Board of Directors meetings and shall see that all resolutions of the Board of Directors are carried into effect. The President shall have the general powers and duties of supervision and management usually vested in the office of president of an organization. The President shall submit an annual report to the membership describing the past year's activities upon completion of a term of office.

Section 3. First Vice-President. The First Vice-President shall, in the absence of the President, perform the duties and exercise the powers of the President. The First Vice-President shall also perform such other duties as may be designated by the President or the Board of Directors.

Section 4. Second Vice-President. The Second Vice-President shall, in the absence of the President and First Vice-President, perform the duties and exercise the powers of the President. The Second Vice-President shall also perform such other duties as may be designated by the President or the Board of Directors.

Section 5. Secretary. The Secretary shall give, or cause to be given, notice of Special Membership meetings and of Special Board of Director meetings. The Secretary shall keep the minutes of all meetings and make them available to the general membership and shall perform such other duties as may be designated by the President or the Board of Directors.

Section 6. Treasurer. The Treasurer shall have custody of the organization's funds, maintain an inventory control procedure and shall keep full and accurate accounts of receipts and disbursements in books belonging to the organization and shall deposit all monies and other valuable effects in the name of and to the credit of the organization in such depositories as may be designated by the Board of Directors. The Treasurer shall have the authority to disburse checks as provided in ARTICLE 10. The Treasurer shall maintain a system of internal fiscal controls and shall report regularly to the Board of Directors on the expenses and financial condition of the organization. The Treasurer shall also perform such other duties as may be designated by the President or the Board of Directors. The Treasurer shall be appointed by the Board.

ARTICLE 10 - FISCAL CONTROL

Section 1. Disbursements. Treasury disbursements shall be made by check. Any disbursement over $10.00 shall be supported by voucher or receipt. All checks, drafts, notes and evidence of indebtedness of the organization shall be signed by the Treasurer and by either the President or First Vice-President; however, the Treasurer shall have the authority to sign checks in an amount not to exceed $500.00 without a co-signature. No person with authority to sign checks may sign a check payable to themselves without a co-signature.

Section 2. Annual Budget. Prior to the commencement of each fiscal year, the Board of Directors shall prepare a proposed budget for the following fiscal year. This budget shall be presented at a Regular Membership meeting for the

approval by a majority of those members present.

Section 3. Fiscal Year. The fiscal year shall be from January 1st to December 31st.

Section 4. Capital Expenditures. Capital expenditures in excess of $1,000.00 must be approved by a majority of the members present at a Regular Membership meeting. Approval of a line item in the annual budget shall constitute membership approval of a capital expenditure.

Section 5. Audit Committee. An audit committee comprised of three members in good standing shall be appointed by the Board of Directors in November to audit the accounts of the organization. An Audit Report shall be submitted at the March Board of Directors meeting.

ARTICLE 11 - AMENDMENTS

These Bylaws may be amended by a two-thirds majority of the Directors of the organization currently holding office and by a two-thirds majority of the members present at a Regular Membership meeting or at a Special Membership meeting called for this purpose.

ARTICLE 12 - DISSOLUTION

Section 1. Dissolution. This organization may be dissolved by a three-fourths majority of the Directors holding office and by a three-fourths majority of the members present at a Regular Membership meeting or a Special Membership meeting called for this purpose, provided that thirty days' notice shall precede such membership meeting by a mailing to all members or by publication in the *Apple Barrel*. Upon affirmative vote to dissolve, proper action shall be taken by the Board of Directors to wind up the affairs of the organization including surrendering the organization's Corporate Charter to the Secretary of State of the State of Texas or other appropriate authority.

Section 2. Assets. Upon dissolution, all of the organization's net assets shall be distributed to bonafide educational institutions as directed by the Board of Directors.

Empirical Fumblings and Arcane Language

Caring for the new family member

The Macintosh and the Apple II series are air cooled by gravity (convection current). They are designed to not need a fan. Those cute little slots in any microcomputer are vital to its survival. Block them and you will shortly need service. Make sure there is at least an inch all around to allow air circulation. The Mac is particularly vulnerable to misuse in this area. The top is so handy a place for a manual that it's almost irresistible. In front under the screen, hidden from view because it's parallel with the base, is a vent section. That's one that most people block by pushing their keyboards up against the front of the machine. The external disk drive looks best snuggled alongside, blocking lower intake vents. I read one article that mentioned using a Mac on its back. Death came within an hour.

You can put a disk drive or modem on top of a Mac if you center it between the vent sections so none are blocked.

Several companies are out to sell cooling fans. There is an unresolved debate in the micro industry about add–on fans. Fans have both the low frequency vibration that you can feel, and also a high frequency vibration that can generally only be detected with instruments. The general feeling is that the vibration of the fan does enough damage to offset the benefits of the extra cooling. It's a no-win situation. With the Apple II series, and after a lot of debate with factory and private service people, we used cooling fans only when there were five or more cards installed. That is, disk drive controller, printer controller, 80 column, and CP/M cards were considered the base load not needing a fan. Going beyond that with 128K RAM cards and internal modems (modulator-demodulator for data transmission on phone lines) seemed to be the situation where the benefits of

cooling outweighed the damage of vibration. Keep in mind that the production cost of including a fan and air guides is slight. If there were a positive advantage, a manufacturer would be happy to stick it in and say "Nyah Nyah, my product has it and the other guy's doesn't."

That brings us to experience in industry with service requirements. In business it is customary to turn the machine on for the day the first time you need it, and leave it on until evening office shut-down. Office machines required less service over a period of years than home machines owned by the same users. Surge current (cold start-up current) and thermal shock (warming and cooling with expansion and contraction) are believed to be ultimately more damaging than continued operation. That is, home machines are switched on when needed, and off when not in use as in coffee break, dinner, telephone and door bell time. Office machines are left on for the entire day. Eight hours of time on an office machine may represent one or less off-on cycles, since some are left running all night if processing is in progress at departure time. Home machines may have twenty or more off-on cycles for identical operating time. This observation is not limited to Apple's products. The same increased service need applied to home MS-DOS machines.

One reason a Macintosh has the programmers switch is to give the user the ability to reboot without flipping the power off–on switch. Same reason that Apple added the control-apple-reset reboot sequence to the IIe and IIc. If you're flipping the off–on switch to reboot, you're preparing to contribute to the early retirement fund of your local micro power supply dealer.

After several years office machines did have definite burn-in on the monitor screens. There were borders and such

permanently etched on the monitor. On some tubes one could clearly see a *Visicalc* type screen even when unplugged. Most computers have no automatic defense, other than the undesirable off and on switching. It is up to the user to dim the screen when not in use, and of course no one does. There are several software desk accessories for the Mac such as *Fade to Black* that will take care of that for you and dim the screen after a user-selectable period of inactivity. The screen stays dim until you press the mouse, while a little icon or something floats around the screen in a random pattern to let you know the machine is still on, since the Mac has no pilot light. The problem with these programs is user inactivity is defined as keyboard or mousebutton input, and they shut down system activity, not just the screen. My screen fade will activate itself on schedule and shut down the system in the middle of a long printout. A press of the mouse button and it starts up right where it left off of course. Set a reasonable time interval or this can become a nuisance.

Everyone has heard about debugging software. Have you considered debugging a machine? Insect life can move in. Spray around equipment. Don't spray into it directly. Some sprays leave an insulating film. Some leave a conductive film. I won't debate which is worse inside a very sensitive piece of electronic equipment. Either of course is destructive in a disk drive, affecting not only the head and pressure pad, but the drive belts found on many designs. If you have a problem, old fashioned moth balls are effective. Wrap some in a clean cloth, open the machine, and find a spot where they won't affect cooling flow. Obviously you a) need to know how to open the machine and b) need to be able to stand the odor. Insect traps and baits hidden behind also help.

The Terrible Twos and other Problems of Infancy

If you look inside a computer CPU there is a invariably at least one or more boards with chips plugged in and a rectangular can that protects the power supply from amateurs like me. Perhaps it also protects me from it. Anyway, the chips are of different sizes, about the size of a sugar cube and black in color. We've all heard about growing silicon crystals by chemical means on circuit designs made by photographing a drawing and reducing it. Now we need some arcane terms for understanding this item.

A Byte is one character. We could call it a character but that would make it too easy. Actually we include things that we don't ordinarily think of as characters because they aren't printable characters. Examples are spaces, carriage returns, line feeds, screen commands, codes that put the printers into condensed and expanded modes, and more. Hence a new word, Byte.

A Bit is a binary digit. That is the smallest storage unit and represents a one or a zero, a yes or a no, an off or an on. A bit is analogous to a switch. It has two conditions. One switch gives us two possible combinations. Everything internal to computers is by two and powers of two. Two switches give us four combinations. Keep going by powers of two. Eight bits give us 256 possible combinations and that pretty much covers the current range of need. We use eight bits to represent a character. It lets us represent the alphabet upper and lower case sets, numbers, punctuation, and invisible stuff too. The standard computer keyboards, with uppercase, lowercase, control/option/escape generate well over 200 different characters.

A bit once took a vacuum tube, then a transistor, now it's a transistor equivalent. A bunch of these equivalents on a common strata is called a chip. Each bit consists of several layers of chemicals with at least three equivalent electrical connections. Just like a switch there is a wire in, a wire out, and a third wire that is the toggle or handle. So for 8 bits, or 1 byte, we need a minimum of 24 electrical connections. There are other internal connections in a chip that connect these wires to various bus wires and ultimately to a few external pins. One simple RAM chip in an old Apple II has 16,000 bits requiring 48,000 connections. To connect all those connections to the bus wire connections that come out to the pins we see, more than doubled the number of internal connections, so 96,000 is a understated number. That machine usually had 24 of those RAM chips, a batch of ROM chips, and other specialized chips that control disk drives, screen output, sound, and the internal housekeeping of the CPU. Five million connections is a fair minimum estimate. ROM by the way means read only memory, or that part of memory that has programming built in at the hardware level such as BASIC, and is not meant to be altered by a user. RAM is random access memory where we store and execute our programs and must be affected by the user. For an enhanced Mac with 512K of RAM and 128K of ROM there are millions of internal electrical connections at a microscopic level of size. One sugar–cube–sized memory chip alone accounts for over 1,500,000 connections. The RAM chips alone theoretically require over 24 million connections, and that accounts for only part of the machine. You can be sure these connections are physically pretty small.

Those little chips were once several wall sized drawings with black lines for wires, dots for transistor bases, and very little white space left. These drawing were photographed and reduced, and reduced again, until they came down to the size of the chip you see. Then photo etching, chemical depositing, and crystallography are all used to produce a final product.

The previous is a long winded way of expressing my continuing astonishment that a microcomputer can be made to work at all, much less continue to do so for a protracted period of time. Separation or incomplete connection at any one of about 50 million points can stop a modern micro dead, and yet it rarely happens. It does happen however, and when it does, it can be expensive and time consuming to locate and repair. Part of owning a micro is knowing that while it is not fragile, it ought not be used as a doorstop or nightstand, until you are ready to cease using it as a computer, and make it into an ornamental lamp. Extremes of temperature, and rapid swings of temperature that makes chips and mother boards expand and contract are hazardous, connections separate or short. Shocks and vibration will also damage connections.

A parked car in the sun gets very hot. In the winter it gets very cold. You are a good deal tougher than a micro, but do you like riding in an unpadded auto trunk? Has anyone dropped you lately? Nothing mysterious here; if you don't like the conditions, neither will your micro. The reverse does not apply. You may like swimming, but your CPU won't.

Considering the physical size of the components of each bit, the currents that may pass through without damage are at microamp level. Very low. Voltage surges and sags can cause damage, although most machines today have excellent and well regulated power supplies. The worst transients (short term major electrical pulses on a power line) are caused by lightning strikes to our local power system, followed by those that occur when power is coming back on after it has been off for a few cycles. If your power goes out, and your machine is on, run, do not walk, to flip the off switch before the power comes on again. If you are in an area where the power frequently goes off for a few seconds during an electrical storm, and then comes back, or just flickers, save your work and shut down during a storm. You have nothing to lose, but your micro may undergo an accidental lobotomy.

With understanding comes expensive panic

That brings us to an accessory called the power or line filter. The power supplies used in the Apple line of computers will handle nicely fluctuations between 90 volts and 135 volts. This device usually is not needed. Usually doesn't mean always. In newer neighborhoods where construction is going on, it's likely that you will have what is known as dirty power. Same device helps if you're going to work during thunderstorms. Serious damage is done by very short-

lived transient spikes. These are very brief periods of extreme excess high voltage, sort of like a static electricity discharge coming over your electric wiring.

The device may be a good investment, provided it's a good device. They all seem to work — once! The $24.95 and under devices, and the $50 power strips with built in surge suppression, generally use a protective device much like a fuse. One really good transient and it's gone, and there is no way to know that by physically observing the unit. So if you need it, expect to buy a sizable box (around the size of a disk drive) for around $100. This is not a standby power supply which will cost $300 and up and does not provide the same function. Nor is it an Uninterruptible Power Supply, better known as a UPS, which is still more costly.

The UPS takes power from the line, converts it to battery voltage to keep its cells charged, and then converts the battery voltage back to line power and feeds it to the computer. The computer never sees line power directly. This type of device is the most effective at all functions. Expect it to cost over $700; more likely $1,000 and up. The true UPS should produce the proper sine wave output your computer needs to stay happy, but it's best to check the specifications if you're buying one. Avoid those that mention square wave or saw tooth output. There are exceptions to every generalization.

The UPS protects the work in the machine from being lost if the power goes out by providing temporary power from a self contained battery, and is therefore protective of the work. It does not infer protection from transients, although most of the true UPS devices currently on the market offer excellent surge and transient protection as a feature. A UPS is primarily intended to protect your work, not your machine. The time they can power a microcomputer is measured in minutes, but it's ample to finish a paragraph, save your work, and shut down in orderly fashion.

Most so called UPS devices under a thousand dollars or so are not really a true UPS. They are standby power devices which connect between the power line and your computer and normally feed the power directly from the line directly to your computer while keeping an internal battery charged. When they detect an interrupt in the power line their internal switching circuit unhooks from the power line, connects the battery to a converter that changes the battery power to 110 volt AC. It then connects that to the computer. Hopefully it does it all before the computer memory has a chance to become scrambled. These are the $300 to $600 devices most commonly and erroneously sold as UPSs. It's probably a good thing that these are not powering your computer full time because most provide electrical power of the right voltage and frequency, but provide it in nearly square wave form. Computer power supplies are designed for a wave form that is smoothly curved, called a sine wave. When they receive a square wave, it's rough on them, and they age prematurely. It's like eating junk food. Now and then it's tolerable, but if it were a steady diet one wouldn't last too long without repairs.

The type of repair that a good line filter or UPS might bypass would be $150 to $300. Plugging in expensive protection equipment to avoid one such type of repair every two years might be uneconomical. Keep in mind that batteries die, even nicads, and so do diode circuits. A standby power supply or UPS will itself eventually require service. It might be cheaper to just make repairs. How important is working during lightning storms? What is the value of work you could lose in a sudden power interrupt? What does not having a machine for a few days cost you? It's your judgement call.

— DAVID SCHEUER
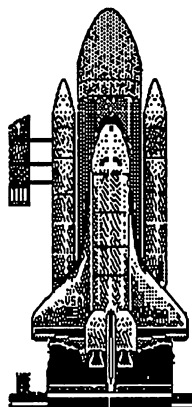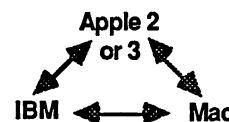
What the Author Gets

Let the user beware! Any computer software written since January 1, 1978, is automatically protected by copyright even without bearing notice to this effect. Although trade secret, patent, and contract law can be applicable, the elective procedure of copyright as endorsed in the U.S. Constitution is most commonly used. Software only enters the public domain and becomes available for indiscriminate use if the creator intends such or declines or fails to register the software within five years from the date of creation. Congress designed the currently applicable 1976 Copyright Act to provide greater protection for those unwary or neglectful creators who are encouraged under the law to concentrate on the **original means by which to express** ideas (not necessarily their own) rather than to anticipate the diverse ways others might appropriate their style, emphasis, or manner of expressing pre-existing computer code or spreadsheet ideas.

The Copyright Act of 1976 at first gave only indirect attention to protecting software from infringement when it first acknowledged the exclusive rights of those who create artistic works (17 USC 106) and then indicated that owners of copyrights in computer works have no greater or lesser rights with respect to their work than those dealing in the literary, musical, dramatic, or choreographic (17 USC 117). Public Law 96-517 in 1980 amended the Copyright Act to define a computer program as a set of "statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." This definition was preliminary to a provision which lessened the exclusive control granted in Sec. 106 and made it no infringement *for the owner of a copy of a computer program to make or authorize the making of another copy or adaptation of that computer program provided:*

(1) that such a new copy or adaptation is created as an essential step in the utilization of the computer program in conjunction with a machine and that it is used in no other manner, or (2) that such new copy or adaptation is for archival purposes only and that all archival copies are destroyed in the event that continued possession of the computer program should cease to be rightful.

The amendment concluded that *any exact copies prepared in accordance with the provisions of this section may be leased, sold, or otherwise transferred along with the copy from which such copies were prepared, only as part of the lease, sale, or other transfer of all rights in the program. Adaptations so prepared may be transferred only with the authorization of the copyright owner.*

For an author (i.e. computer programmer) to create something that the law enables the author to control in the public arena the created work must be original, fixed in some tangible form of expression, and subject to being perceived, reproduced, or otherwise communicated with or without the aid of a machine or device for more than a transitory period of time.

The author of a computer program owns the copyright. Although this usually is the person who writes the software, it could be the employer for whom the program was written as part of the programmer's job. Whether the work is produced for hire or through self-employment, the copyright holder gets five exclusive rights: (1) to reproduce his work in copy form; (2) to prepare derivative works based on the original; (3) to perform the work publicly; (4) to display the work publicly; and (5) to distribute the work to the public. Since these rights are severable and divisible, they can be sold or assigned individually to others as the author-owner may choose. Thus the author can sell program marketing rights to one or

more software distributors and at the same time exercise the reserved right to personally make and sell copies of the same work. It should also be noted that the right to distribute a program carries with it the right to control how and to whom the program gets distributed, and thus free software even downloaded by modem from a public database may still be an infringement on the copyright holder's legal control over the manner and means of distribution. Therefore, because a work is publicly available and seemingly free may really mean that the author rather than having entered the work into the public domain has had his right to control distribution infringed. The author's intent to exercise exclusive control rights and the effort to manifest public notice of this intent raise the barriers to the unauthorized use of software which can result in penalties ranging up to $10,000 and/or one year imprisonment.

The federal copyright office has available free circulars which outline federal copyright regulations and procedures. Circular R61 pertains to the registration of computer programs. Circulars can be ordered from the: Copyright Office, Library of Congress, Information and Publications Section, LM-455, Washington D.C. 20559.

Copyright application forms can be ordered by calling (202) 287-9100. Assistance is available at (202) 287-8700.

Next Month: What the User Gets
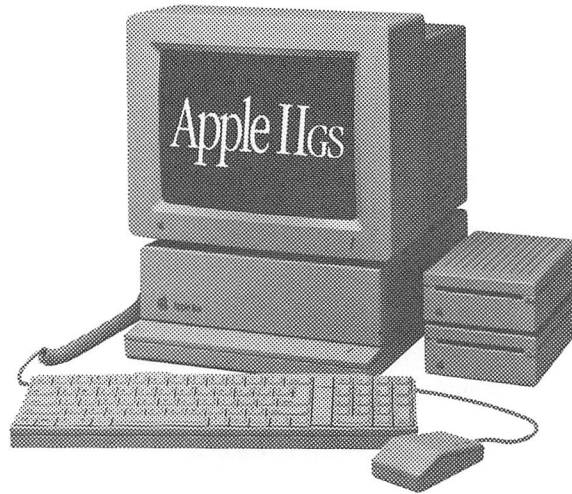
—WILLIAM LINSLEY

Programming the Apple IIGS

With the introduction of the new Apple IIGS, one can be sure that many programmers and would-be coders are feverish about getting into the inner workings of this new machine. The color capabilities of the machine and the fantastic sound afforded by the Ensoniq chip augmented by the familiar user interface of Macintosh fame are sure to inspire new applications reaching exciting new bounds. If you are a II programmer coming to the new machine, a Mac programmer wanting to create something taking advantage of the new capabilities, or just a hobbyist wishing to play, it becomes necessary to find the information needed to undertake such a task.

Unfortunately, like the lack of information with the Macintosh's introduction, many programmer's will feel they will have to wait until companies start to produce programming systems and books begin to be written. However, I feel that some information is already available and programming systems are already at or just around the corner.

To take advantage of the new software architecture of the Apple IIGS, it will be necessary to learn new programming principles. The best source for this information is from other programmers or at least from books. One such book is *Apple IIGS Toolbox Revealed* by Danny Goodman (published by Bantam Books). The book covers the theory behind drawing windows, managing menus, drawing graphics, dealing with sound, and undertaking other tasks.

Apple also will be releasing two books entitled *Apple IIGS Toolbox Reference, Volumes 1 & 2*. Like the infamous *Inside Macintosh* of Macintosh circles, these volumes will more than likely become the definitive sources of information on the Toolbox routines. However, other books will probably be necessary to help decipher or at least to enhance these volumes.

After you get reading material, the next inevitable step is getting a system in which to program this new beast. However, at present I only know of two systems even though neither has been released yet (as of writing this, not your reading this). It is interesting because a company whose product I have used extensively on the Macintosh has had a hand in both products. That company is Megamax whose C compiler is considered a very good value for the Macintosh.

For the IIGS, the first product is known as the *Apple IIGS Programmer's Workshop*, or *APW* for short. For those familiar with the ways of Mac programming this is equivalent to the *Macintosh Programmer's Workshop*, or *MPW*. Created by Apple Computer, the *Programmer's Workshop* is a complete programming environment which consists of a shell, editor, linker, debugger, utilities, and a 65816 assembler. To this you can add a C language compiler. It is this compiler that was written by the guys at Megamax. For creating applications for the GS on the GS it will probably be hard to beat this combination. December is the tentative date for release of *APW*.

However, Megamax has another system aimed at serious developers. The system requires an Apple IIGS, a Macintosh, and a product of Megamax's. This product is a C cross-compiler which allows the programmer to do the actual editing on a Macintosh where he has stable tools, compile it there where the speed of the Macintosh is welcome, and then download it to the GS where it is run. Furthermore, it is debugged from the Mac's screen. Available in November should be the Mac-to-GS C compiler and in December a Pascal equivalent should be available. Priced at $500 per module this product will probably appeal most to professional developers; however, hobbyists might want to make the investment.

At this point, the only information that I have has been gathered from the literature, other people, and the October MacWorld Exposition. If anyone gets any other information feel free to share it with the rest of the readers. Although my prime interest is in the Macintosh programming arena, I would welcome the opportunity to see the new machine's software and to get a series of articles on programming the machine. I feel that having such a series could be exciting, inspiring, and could result in some excellent products coming from programmers from the ranks of HAAUG.

— *CHRIS FLICK*

ASCII and Ye Shall Receive

(or, Access to *Access //* and Riding the *ASCII Express*)

Okay, I admit it. I play with modems. Years ago, I knew something about mainframe data communications — how to spell SOH, CLS, ACK, and NAK, and even what they meant. However, I took six years "off" when the only meaning "asynch" had was a place to put dishes and am now only "back" part-time. We have an Apple //e because of our kids. As the North Harris County PC Users Group newsletter editor (this is a mostly IBM-type group that also has an Apple SIG), I use my Apple to write/edit articles. I needed a way to receive articles from other computer types so I began to learn about Apple telecommunications.

As a HAAUG Heaven BBS participant I quickly learned that a program called *ASCII Express Pro (AE Pro)* was **THE** telecommunications program for the Apple //, as it had the most capabilities. I also learned that it is a really difficult program to set up. Maybe there was a simpler solution? I have a 300 Baud internal Networker modem (circa 1983 with no Auto Anything), and pre-*Netmaster* software. This means I can receive ASCII (text) files and communicate in Terminal Mode (type directly to other computers, including non-Apples), but cannot transmit files. I needed to upgrade. SuperTec Technology Center in Humble loaned me their *Apple Access //* package to help with the production of the newsletter, and I would like to pass on what I learned.

Access // is a data communications program that lets the Apple //e or //c Communicate with information services such as Dow Jones, CompuServe, and The Source; exchange data, files, electronic mail, and programs with another computer over telephone lines; act as a dedicated terminal hardwired to a host computer. Note: *Apple Access //* will not work on the Apple // or //+, because the software requires the Delete and Open-Apple keys of the "enhanced

keyboard." *Access //* is not copy-protected, and retails for about $75.

The package comes with a training disk. The "tutorial" is a simulation, and is more of a demo disk that does some teaching than a tutorial that does some demoing. It glosses over the complexity of telecommunications, but does explain some basic terminology.

The format of the manual is good. When so much information must be discussed, its presentation is important. It is broken up in small "bytes" with lots of visuals: screen layouts, program menu logic flowcharts, and grey-shaded "By the Way" boxes of info, hints, and warnings.

The "Setting Up the Equipment" section was sketchy. It describes an external modem setup and completely ignores internal modems (an internal modem — also called a direct connect modem — is a circuit board which is placed in a slot on the motherboard of your computer; an external modem is a circuit board that is encased in a box which is outside your computer, and is attached through a port in the back to another circuit board which is placed in a slot on the motherboard). That is because *Apple Access //* DOES NOT SUPPORT INTERNAL MODEMS! To use this software, you MUST have a Super Serial Card configured for an external modem, and it MUST be in Slot 2. And no, you cannot just put the Super Serial Card that you bought as your printer interface in Slot 2, and just readdress its Slot number in the software you run (like *AppleWorks*). The card has to have its dip switches set differently, and there is a chip that must be taken off and turned around to be used with an external modem. Don't care to do that? Then you purchase an additional Super Serial card.

This put a real damper on my experiment, as I was unable to actually use the software! But let me go on.

The menus of *Access //*, although

foreign to the very first time user, were pretty straightforward. The Main Menu gives you the following choices:

1. Dial a Service
2. Terminal Mode
3. Set up Communications
4. Transmit a File
5. Receive a File
6. Utilities
7. Help
8. Quit

These choices are further subdivided. For example, when you select #3, The Set Up Menu displays:

1. Terminal Characteristics
2. Set Speed
3. Set Parity
4. Answerback
5. Tab Stop
6. Enter Autodial Numbers
7. Close the Recording File
8. Save Access

Configuration

So setting up a telecommunication environment is not simple! The *Access // manual* (195 pages) does a nice job of explaining lots of basic concepts, but sometimes omits important information. For instance, to get one computer to understand the digital language of another computer, a common language or "protocol" must be established. The manual describes Christensen Protocol as: "a technique for data transmission and error checking that was developed by Ward Christensen of Chicago. The protocol divides the data to be transmitted into blocks of 128 bytes and adds a three-byte header and a one-byte checksum to each block." It does not check for control characters, and any kind of data can be sent: ASCII, binary, etc. This protocol functions at any baud rate, and uses 8 data bits, no parity, and one stop bit (8-Null-1). Well, that's wonderful,

but part of setting up the software includes deciding what protocol you need, and it never defines what type of terminals (the computers on-line to each other) use it. As usual, the Sysops (System Operators) of HAAUG Heaven, Clark Johnson and Rick Oshlo, were quick and efficient with their explanations. From Clark: "Christensen protocol is really just another name for Xmodem. Actually, it was the first name, as Ward Christensen is the developer of Xmodem. (He placed the protocol in the public domain — that's why it has become the standard.) Xmodem is easier to spell than Christensen." Xmodem, by the way, is a commonly used error-checking protocol used by many kinds of computers. The manual should have included this.

The manual does explain online techniques so basic that pros forget to mention them. One was why you would want to "Transmit" a long text file rather than type it in "Terminal Mode." Any baud rate is faster than "type" rate, and if you reduce connect time, it means reducing the cost.

Intuitively obvious, you say? Not when you are new.

Access // provides the usual "command file" capabilities or "macros" (a keystroke-saving method). Like in *AppleWorks*, these commands are easy to use because they are "pull down menu options" or, in "Terminal Mode," are Open-Apple-Letter combinations, including the familiar Open-Apple-? for HELP!

Access // supports Auto Dial if your modem does. You can list phone numbers, type of dialing (pulse or touchtone), and Command file macros (to reduce keystrokes and time consuming logon). Sample command files are included on the program disk which provide auto logon (up to the ID and password, which you must provide) for some of the popular information services.

AE Pro has a feature that many other communication packages do not have, which is the ability to recognize "file type" when transmitting and receiving. *Access //* apparently has it, too: it can "receive" transmissions from other computers in two ways: 1) It can

"Record" using the "Open-Apple R" command. This saves the transmitted files as ASCII (text), and is used when the other computer is either sending an ASCII (text) file, or is not an Apple // computer. Or, 2) It can "Receive A File" by Christensen Protocol (XMODEM). Receive is for various file types: BIN (binary), SYS (system), AWP (*AppleWorks* Word Processor), ASP (*AppleWorks* Spread Sheet), ADB (*AppleWorks* Data Base), etc. This is important, because, as Rick Oshlo put it: "one of the problems with Apple II file transfers is that there is no standard way to transfer the file type information. It appears that Access // has a 'built in' function to set the file type before you receive the file. Evidently, TXT, AWP, ASP, & ADB files can be selected from a list but all the other ProDOS file types must be designated by entering their 3 character code." Most other telecommunications packages can only save files to disk in ASCII (text) format. For these, you must use yet another program to convert these text files back to their "true" type. On HAAUG Heaven, if

your software can only handle text files, you convert a "downloaded file" (a received file) to its true type by using Rick's SET.FILE.TYPE program. To use this program, you must know the starting address of the downloaded file!

So what doesn't *Access II* do? HAAUG Heaven taught me that not everyone is satisfied with this program. Although I know several people who are (one is a writer who uses it about seven hours a day), *Access II* does not have all of the features of *AE Pro*: *Access II* does not support internal modems (*AE Pro* does); does not allow back scrolling; only has limited room in the autodial directory; does not allow direct printing while on-line (*AE Pro* does); does not support terminal emulation other than ANSI VT100 and DEC VT52 (*AE Pro* supports 11 types automatically, and you can set it up to emulate others too); only has a data buffer size of 3,000 characters (3K — compare to *AE Pro*'s 18K buffer). These are just a few, and the pros could tell you about more.

A friend loaned me the *ASCII Express Pro* package for a few days. It's manual is over 350 pages, and at first glance seems impossible to understand. However, it supports a wide variety of modem types, including my outdated internal modem (when emulating the Apple Communications Card), and I was able to try it out! Actually, once you get *AE Pro* working properly, it isn't so hard to use.

My conclusion, then? If it worked with my internal modem, *Access II* would be great for me. It does everything else I really need it to do — for right now. *Access II* is a nice package for the new user, as it is relatively easy to install and use. *Access II* could be limiting (if not useless), however, depending on your telecommunications needs. If so, you can borrow or rent an expert to help set up *ASCII Express Pro*. Since I will have my internal modem for a while, and since I like to experiment, the only solution for me is to buy and ride the *ASCII Express*!

— Ginny Morley

A man wakes up in the morning after sleeping on an advertised blanket, in advertised pajamas. He will bathe in an advertised tub, wash with an advertised soap, shave with an advertised razor, drink advertised coffee after his advertised juice, cereal & toast (toasted in his advertised toaster), put on advertised clothes and an advertised hat. He will ride to work in an advertised car, sit at an advertised desk, write with an advertised pen, and smoke advertised cigarettes. Yet this man hesitates to advertise, saying that advertising does not pay. Finally, his unadvertised business goes under. He will then advertise it "For Sale."

Imperium Galactum:
To Conquer and Colonize

This game is set in the future when faster than light speeds are normal. The color map shown on the screen contains fifty star systems of various types. For those of you who do not have a color monitor a map is printed on the back of the player's manual which shows the type, name and position of each star system. Each star type is given a different color to denote each of the five star types.

The manual explanation is adequate to inform players about each star type. From one to four players may play this game. You may play the computer solo or add computer played powers when playing with other people. The object of this game is to increase the population of your empire. The player with the highest population wins the game. The computer does not declare a winner at a fixed point in the game. It is therefore necessary if you play with friends to decide how many turns will constitute a game. Since each turn amounts to five years game time, a game of 40 to 80 turns would very likely give all concerned a challenge. This game has four skill levels and the manual's comments regarding the highest and lowest are very accurate.

Some of the features of this offering are the ability to design your own ships, to colonize new worlds, to conclude treaties with independent star systems, to raise your technological level, to modify a worlds atmosphere, to control worlds' economies, and to bargin with other human and computer players as to their attitude toward you. Three diplomatic stances are allowed. Hostile players may attack at any time they choose to do so. If you are neutral toward a player you may only attack if you are attacked. If you are friendly toward a player you may not attack and are not allowed to defend yourself for the first round of combat.

To some extent this game has the flavor of *Diplomacy* on the computer.

There is a negotiations phase each turn and the diplomatic stance of the player with which you are negotiating is shown as of the prior turn. The computer determines the result of negotiations with computer players and can do so for human players as well — if that is what they want. You can try more than once in this phase to alter another's stance toward you. To play the game you must explore the galaxy and try to colonize and also be willing to conquer occupied star systems. Both a standard and a random map set up are offered as playing options.

For those of you who decide to add this offering to your library I offer these playing tips. Redesign one of the small ship classes so that it has no armor or offensive ability. What you want in the first two to four turns is speed. Build about ten of these ships with a speed of 16 and use them to explore. You will need to conclude treaties with independent star systems and the first ship to

reach such a system gains the advantage in treaty negotiations. After the galaxy has been explored, redesign these ships to war ships. I found it useful to give them a planetary bombardment factor of 5 and an energy or missile ability of 5. If you play the computer I suggest you start at the lowest level and learn to use the various options against only one player. Try to conquer the galaxy.

Plan to start colonization on the first turn. To colonize successfully you must have surplus food, transports, and traders. It takes one transport to carry a colonist. The number of traders needed is shown on the production phase screen. On the first turn adjust your planets population (pop) and build five or more transports. WPP is defined as the total number of people needed to work your industries, mines, and farms. Pay close attention to this example of how to create surplus food. The hardest thing to learn was how to do this.
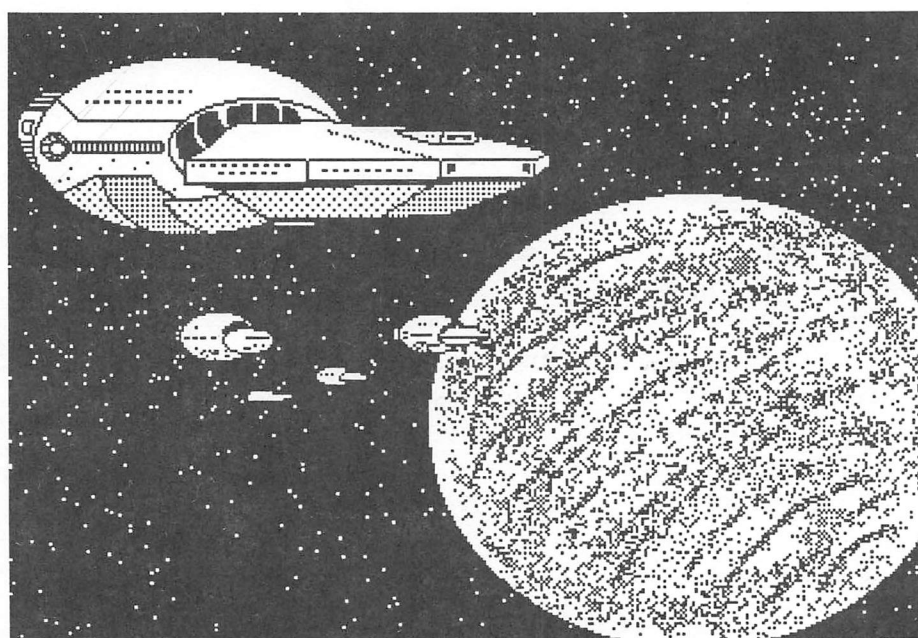
A mission to colonize or conquer — *Imperium Galactum* lets you decide!

When you start, your pop, which is your current population is equal to your working pop you have no surplus food. Lets suppose that your pop equals 140, working pop equals 140, and you choose to build 10 transports. On the first turn adjust your population to 149. Do not add any industry, mines or agriculture in the production phase. On the next adjustment phase set your population to 139. In the movement phase of the turn 2 load your transports and send them on their way. Had you built only 5 transports you would set pop on turn 1 to 144 and on turn 2 to 139. Once you get to the planet to be colonized unload 1 colonist on the second movement turn to claim the planet. In the production phase of the following turn be sure to position the cursor over that star system and buy agriculture and mines for your colonists. If you can buy but one of these items then buy agriculture. On the first movement turn of the round you bought for your new world, land your remaining colonists. Be sure to adjust the population of this world before going to the production screen.

Once you have an agricultural base going you will not need to worry about surplus food with which to feed more colonies unless that agricultural base is destroyed and there are no others to fill in for it. When exploring go to class G and class K star systems before exploring other star types. Always negotiate with independent star systems. Spend some money building up allies' defenses and use their shipyards to build up your navy more quickly. You are allowed to build ships on every friendly world allied to you. Be sure to move the cursor and check each one in the production phase. Although ships on garrison have a chance of intercepting raiders while ships on repair missions do not, I found that raiders always attacked me either way and if I was on repair mission all my ships started at 100% effectiveness while the enemy ships needed repair to some degree. Since most class M systems have no planets, they may be used as a stopping point where combat is unlikely to occur. This is useful in both attacking and retreating where you don't want to weaken your attack force before arriving at your target or you want to survive to

fight again. Create 1 or 2 task forces of transports only to free up your war ships for both faster movement and use in another place. Read rule 6.2 concerning planet economics until you understand it well enough to be able to figure WPP needed. Build up your tech level as fast as you can. It adds to your ship's speed and fighting power.

In my opinion this is not a game for young children. Do not plan to complete playing it in one or two sessions unless you are a marathon game player. Use the "save the game" feature which worked fine for us. This is truly a campaign type game for those who enjoy long, somewhat detailed gaming. Those who want an evening romp or whose attention span is short probably will not enjoy this game. I found that playing multiple computer players required repeated key pounding at the diplomatic stance table to survive to expand. Since at the hardest and next to hardest level the enemy gains large ships from the start and you have to build up to them, you either keep them from being hostile with key strokes or you end up fighting battleships with cruisers and smaller ships. Having to use around 25 key strokes and up to possibly 100 each round fast became tedious and dull for me. Those of you who enjoy the *Diplomacy* board game will most likely want to try this one. It seems to me that it plays much better human versus human than it does computer versus human.

—BILL MUHLHAUSEN

This product is priced at $39.95 and was donated to us for review by The Software Place on Westheimer.

Number Formatting In Applesoft BASIC

A few days ago, Mike Sechrist, a fellow HAAUG member, relatively new Apple // owner, and programmer of mainframes, was lamenting the fact that there is no built-in number formatter in Applesoft BASIC. I found a generalized FORTRAN style formatter subroutine on a disk in my "play room" and made him a copy. I had discovered the routine several years ago buried in a program in an exchange newsletter. I don't remember what the main program did, but I have found the formatter to be very useful and have, in fact, used it extensively in my *Grade Master* package. I have no idea who wrote it or what user group he belonged to so I can't give him/her credit.

The formatter routine accepts a floating point number (one whose variable name is not followed with a % sign) and a string that represents the desired numeric format (integer, floating point, or exponential) and returns the formatted number in a string. A listing of the subroutine and a sample program that repeatedly calls it with the same number but different formats is included below, together with the output that results if you run it. The output fields are all printed to the same starting columns (1, 16, and 28) to permit easy comparison of output alignment. The following paragraphs explain how to use the formatter routine, but do not explain how it works. As they used to say in school, the details are left as an exercise for the student. As is often the case, I use code I recognize as useful and often modify it to correct bugs or make it work better. When it is as complex as the formatter is, I rarely try to understand its inner workings.

For those not familiar with POKEs, the POKE 36,n positions the cursor (or print head) at character position n+1 and works with the 40 or 80 column screen and essentially every printer interface produced these days. As you may have found out already, HTAB does not always give the expected result and SPC(n) cannot be used easily to align information.

Integer Format

Lines 280 through 310 set up outputs with integer format. Note that the string CF$ starts with the letter "I" (for integer) and ends with a number representing the number of characters in the output field. First the number is rounded and converted to an integer. If the resulting number has fewer characters than the field width, it will be right justified in the field and have leading spaces. If it is too large for the field width specified, the cell will be filled with asterisks as shown in the

```
100  REM   ----------------------------
110  REM
120  REM   FORTRAN STYLE FORMATTER
130  REM
140  REM            FOUND BY
150  REM          MIKE KRAMER
160  REM
170  REM   IN ANOTHER USERS' GROUP'S
180  REM   NEWSLETTER YEARS AGO AND
190  REM   ADAPTED TO HIS OWN USES.
200  REM   ORIGINAL AUTHOR UNKNOWN.
210  REM
220  REM          JUNE 1986
230  REM
240  REM   ----------------------------
250  REM
260  HOME : INVERSE
270  PRINT "OLD VALUE";: POKE 36,15: PRINT "FORMAT";: POKE 36,27: PRINT "NEW
     VALUE": PRINT
280 C = 1000.35:CF$ = "I3": GOSUB 430
290 C = 1000.35:CF$ = "I4": GOSUB 430
300 C = 1000.35:CF$ = "I5": GOSUB 430
310 C = 1000.35:CF$ = "I6": GOSUB 430
320 C = 1000.35:CF$ = "F4.3": GOSUB 430
330 C = 1000.35:CF$ = "F8.3": GOSUB 430
340 C = 1000.35:CF$ = "F7.2": GOSUB 430
350 C = 1000.35:CF$ = "F7.1": GOSUB 430
360 C = 1000.35:CF$ = "F10.2": GOSUB 430
370 C = 1000.35:CF$ = "E3.2": GOSUB 430
380 C = 1000.35:CF$ = "E9.2": GOSUB 430
390 C = 1000.35:CF$ = "E9.4": GOSUB 430
400 C = 1000.35:CF$ = "E9.5": GOSUB 430
410 C = 1000.35:CF$ = "X9.5": GOSUB 430
420 NORMAL : END
430 GOSUB 560: PRINT C;: POKE 36,15: PRINT CF$;: POKE 36,27: PRINT C$: RETURN
440  REM
450  REM   ----------------------------
460  REM   FORMATTING ROUTINE FOR
470  REM   NUMBERS. ENTER WITH
480  REM   NUMBER IN C, FORMATTED
490  REM   STRING C$ RETURNED.
500  REM   CF$ CONTAINS FORMAT
510  REM   SPECIFICATION AS PER
520  REM   STANDARD FORTRAN.
530  REM     F10.3, E9.2, I5
540  REM   ----------------------
550  REM
560 CG$ =  LEFT$ (CF$,1): REM   GET FORMAT LETTER
570 C1 =  VAL ( RIGHT$ (CF$, LEN (CF$) - 1))
580  REM   CS=#DIGITS, CD=#DECIMAL PLACES
```

first example.

Floating Point Format

Lines 320 through 360 set up five floating point examples. For those not familiar with FORTRAN formatting, the letter "F" stands for floating point

(clever), the number to the left of the decimal specifies the number of character positions in the field including the decimal, and the number to the right of the decimal specifies the number of decimal places.

In the first floating point example,

```
590 CS = INT (C1):CD = INT (10 * (C1 - CS) + .4)
600 IF CG$ = "F" THEN 670
610 IF CG$ = "E" THEN 810
620 IF CG$ = "I" THEN 950
630 C$ = "F-ERR": RETURN
640 REM    ----------------------
650 REM    FLOATING POINT
660 REM    ----------------------
670 CC = C
680 C1 = INT (.5 + CC * (10 ^ CD))
690 IF C1 > = 1E11 THEN 1010: REM   OVERFLOW
700 CG$ = STR$ (C1)
710 IF CG$ = "0" THEN CG$ = LEFT$ (00000000000",CD+1)
720 IF CD = 0 THEN 760
730 IF LEN (CG$) = CD THEN CG$ = "0" + CG$
740 C$ = LEFT$ (CG$, LEN (CG$) - CD) + "." + RIGHT$ (CG$,CD)
750 GOTO 960: REM RIGHT JUSTIFY STRING
760 C$ = LEFT$ (CG$, LEN (CG$) - CD) + "."
770 GOTO 960
780 REM    ------------------------
790 REM    EXPONENTIAL NOTATION
800 REM    ------------------------
810 IF C = 0 THEN CE = 0: GOTO 830
820 CE = INT ( LOG ( ABS (C)) / LOG (10))
830 CC = C / (10 ^ CE): REM  NORMALIZE C
840 CE$ = "E" + STR$ (CE): REM   EXPONENT
850 CS = CS - LEN (CE$)
860 GOSUB 680: REM DO MANTISSA AS FLOATING POINT
870 C$ = C$ + CE$: REM   ADD EXPONENT
880 RETURN
890 REM    --------------------------
900 REM    INTEGER NOTATION
910 REM
920 REM    --------------------------
930 REM    RIGHT JUSTIFY STRING C$
940 REM    --------------------------
950 C$ = STR$ ( INT (C + .5))
960 IF LEN (C$) > CS THEN 1010
970 IF LEN (C$) = CS THEN 1000
980 CG$ = "                    "
990 C$ = LEFT$ (CG$,CS - LEN (C$)) + C$
1000 RETURN
1010 C$ = LEFT$ ("********************",CS)
1020 RETURN
1030 END
```

OLD VALUE	FORMAT	NEW VALUE
1000.35	I3	***
1000.35	I4	1000
1000.35	I5	1000
1000.35	I6	1000
1000.35	F4.3	****
1000.35	F8.3	1000.350
1000.35	F7.2	1000.35
1000.35	F7.1	1000.3
1000.35	F10.2	1000.35
1000.35	E3.2	*E3
1000.35	E9.2	1.00E3
1000.35	E9.4	1.0003E3
1000.35	E9.5	1.00035E3
1000.35	X9.5	F-ERR

Figure 1. Sample Output

the field is too narrow so the field is filled with asterisks. In the second example, more decimal places are necessary, so an extra zero is appended to the end. The third example is just right. The fourth example truncates the second decimal place since only one was specified and, since the resulting number has fewer characters than the field width, it is right justified in the field. You can figure the fifth one out yourself.

Exponential Format

Lines 370 through 400 set up examples of exponential formatting. Exponential format is similar in setup to floating point. The number to the left of the decimal specifies the field width, including the decimal, the letter "E", and the power of ten. The resulting output is a number followed by the letter "E" and the appropriate power of ten.

In the first example, the field width specified is too narrow. Rather than a string of asterisks, an asterisk followed by an "E" and the specified power of ten is displayed. I did not try a field width of 1 or 2, so I'm not sure what the result would be. As with floating point format, the number is truncated if the number of decimal places specified is less than the number has to start with.

Error Format

Line 410 is an example of an invalid format. If a format other than "I", "F", or "E" is specified in the variable CF$, the result is an error message as shown in the last example.
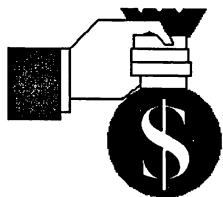
— MIKE KRAMER

ANNOUNCING

TAX PRO 87

INCORPORATING THE NEW TAX CODE

What will my income tax be next year?? Find out by using the new TAX PRO 87 spreadsheet template with AppleWorks and discover on a daily basis what your tax situation will be in 1987 under the new Tax Reform Act!! Determine the amount of money to be owed to or refunded by the IRS when your 1987 tax return is filed in 1988. Don't be surprised on April 15th!! Plan ahead!!

TAX PRO 87 is a spreadsheet template designed for use with AppleWorks which permits the user to keep track of his current tax status throughout the year. It is for the average taxpayer who has multiple sources of income ie.,wages, spouse wages, interest, dividends etc and who itemizes his deductions. The tax tables and calculations contained in TAX PRO 87 are those that will be in effect for 1987.

TAX PRO 87 may be enhanced by the user as required to accomodate those transactions that occur which are not normally encountered by the average taxpayer. The intent of TAX PRO 87 is to simplify your tax situation in an easy to use manner so that you are not surprised when you file your tax return.

Like FINANCIAL PRO, TAX PRO 87 is easy to use. Once the financial data is entered, the results are immediately available. By using the powerful AppleWorks spreadsheet, "What If?" scenarios can be presented for evaluation.

For a limited time you can buy TAX PRO 87 for only $39.95. To order your copy of TAX PRO 87 send a check or money order with the following coupon.

John B. Slack
1179 Bournewood
Sugar Land, TX 77478
Please send me _____ copies of TAX PRO 87 @ $39.95 each.

NAME_____

ADDRESS_____

CITY_____STATE_____ZIP_____

Dazzle Draw & Fantavision

Introduction

Brøderbund, the people that brought you *Print Shop*, now have two other graphics packages that are a pleasure to use. They are *Dazzle Draw* and *Fantavision*.

Dazzle Draw

Dazzle Draw is a double hi-res color graphics sketchpad. It requires an Apple //c or a 128K Apple //e (version B) with an extended 80 column card, disk drive, color TV or monitor, and either a mouse, a graphics tablet, or a joystick. The mouse or graphics tablet are recommended for greater control of the cursor.

The palettes contain 16 colors and 30 patterns. There are four brush sizes and six brush shapes. Other features are spray paint, flood fill, zoom, text, shapes, lines, capture, cut, copy, paste, grid, and mirror. All features are accessible through pull-down menus.

Dazzle Draw allows you to create slide shows to show off your creations. The slide show disk is self-running and does not require *Dazzle Draw* to run. This capability allows you to share your creations with friends. You can dump to an Apple Scribe, Apple DMP, Apple Imagewriter I, C.Itoh Prowriter, Epson JX, FX, RX, MX, Gemini 10X, 15X, Delta, and Radix printers. The newer version will dump to the Imagewriter II. The printer setup is simple. However, there is a limitation on my version that will allow one to select only the Apple Super Serial Card as the printer interface. I cannot comment on the ability to use other interface cards since I have a //c and an Imagewriter I.

Documentation is good. It is a step-by-step tutorial so it should not take long to get the hang of things.

Fantavision

Fantavision is simple to use and can keep you entertained for hours. Just a few minutes of playing with this hi-res animation package will produce results that will amaze you.

Fantavision requires an Apple // with a minimum 64K, disk drive, color TV or monitor, and either a mouse, a graphics tablet, or a joystick. The mouse or graphics tablet are again recommended for better cursor control.

The frames are constructed in a similar fashion to *Dazzle Draw*, except that you are limited to eight animated objects per frame and each object can not contain more than thirty-two points. The features are accessible via pull down menus and border control icons that are similar to those in *Mouse Paint*.

Fun, entertaining and creative. Brøderbund should be commended for two more great additions to their line of software.

The joy comes with experimentation. Simple examples illustrate the power of this program as well as complex examples. In frame 1 select to place a size 9 (largest) dot on the upper left corner of the screen. Clone frame 1, move to frame 2 and the dot will be copied there. Capture the dot using the capture icon, and move the dot to the bottom center of the screen. Now, clone frame 2, move to frame 3, capture the dot and move to the upper right corner. All you have to do now is select GO and watch as the ball bounces. The program will automatically fill in up to sixty-four intermediate frames between the frames that you have created to produce a smooth animated sceen. This effect is called "tweening" and it is the heart of *Fantavision*.

Transformations are also a lot of fun. Draw a face of a man in frame 1 and a face of an elephant in frame 2 then press GO to see your man transform into the elephant.

The back of the disk is a demo movie. There are several examples of the power of *Fantavision* that should fire up the creativity of anyone with the smallest amount of imagination.

You can create movie disks that are self-running and do not require *Fantavision* to run. That way you can share your creativity with your friends. As an added bonus, you can submit movie disks to Brøderbund and receive $100 if they use your movie as a demo on the back of *Fantavision* in future versions.

Documentation is good. There is a step-by-step tutorial that should have you off and having a blast in no time. *Fantavision* was also the editor's choice in the February 1986 issue of *InCider*.

All in all, both of these packages are excellent in almost every way. Fun, entertaining, and creative. Brøderbund should be commended for two more great additions to their line of software.

— LARRY STEHLING

Apple II

Apple Access	Patrick Hennessey	342-0643
AppleWorks	Jack Cowart	467-4215
	John Slack	491-1747
AppleWriter II	Robin Cox	778-1635
AppleWriter III	Shorty Hirsekron	697-9797
CAD-Apple	Dick Fairman	723-9619
dBase II	H. Helm	529-7405
Data Factory	Dick Fairman	723-9619
Dollars and Sense	Leon Cesazini	467-3650
DowJones Market		
Analyzer	Ed Hurley	497-8877
Home Accountant	Rick Oshlo	578-8530
IAC Calc	Chris Kravits	320-0381
Letter Perfect	Bill Muhlhausen	668-3963
List Handler	Rick Kravits	320-0381
MCI Mail	Patrick Hennessey	342-0643
Multiplan	Mackie Brown	688-8655
	Jack Bailie	440-5963
	Glen Ray	359-2103
MultiScribe	Henry Bernstein	728-0081
Palantir	Re Kramer	358-6687
PFS	Jack Cowart	467-4215
	Re Kramer	358-6687
	Ed Haymes	522-1863
Pie Writer	Ray Fielding	358-4217
Quickfile	Jack Cowart	467-4215
ScreenWriter II	Dick Fairman	723-9619
Space Eggs	T. Sloan Engle	774-9055
Time Is Money	Dick Fairman	723-9619
VersaForm III	Shorty Hirsekron	697-9797
Wordstar	Ed Haymes	522-1863
Word Handler	Rick Kravits	320-0381

Would you like to be a specialist? Leave your name and phone number at the Answer Desk during the meeting or mail it to the *Apple Barrel*.

Macintosh

Aztec C	Mike Martin	486-9169
DB Master	Mike Conway	495-2292
Dollars & 'Sense	Gerald L. Penn	498-0079
	Greg New	568-0290
Filevision	Mike Martin	486-9169
Helix	H. Helm	529-7405
Mac-SIG	Steve Bass	847-4407
MacDraw	Billy Jacobs	480-4410
MacDraft	Billy Jacobs	480-4410
MacPaint	Bill Hailey	682-5147
MacWrite	Bill Hailey	682-5147
Mac 3-D	Larry Stage	358-1105
Megamax C	Chris Flick	363-3153
MGMS CAD	Steve Bass	847-4407
Microsoft BASIC	Bill Hailey	682-5147
Microsoft Chart	Mike Martin	486-9169
Microsoft File	Gerald L. Penn	498-0079
	Greg New	568-0290
	Mike Martin	486-9169
Microsoft Multiplan	Mike Conway	495-2292
	Peter Lemettais	523-1390
Microsoft Word	Virginia Conway	495-2292
OverVue	H. Helm	529-7405
TML Pascal	Grady Beaird	529-8420
Tekalike	Mike Martin	486-9169
XLisp	Mike Martin	486-9169
ZBasic	Tom Dillon	376-6502

Hardware

Abatron Optical		
Character Reader	H. Helm	529-7405
Applied Engineering		
Upgrade Cards	Bruce Sprague	360-1000
Sider Hard Disk Drives	Robin Cox	778-1635
Mac XL	Mike Conway	495-2292

Membership Chairman's Report

As you may know, Midge Kocen, who has served as HAAUG's Membership Chairman for the past several years, recently moved to a small island in the Caribbean (tough job but someone had to do it) and I accepted the open position. Midge did an outstanding job during her tenure. She's a tough act to follow, but I plan to strive to do as good a job as she.

We are instituting several changes in the membership area. First off, we are completely revamping the new member's "starter kits" to make them more informative about the purpose and structure of HAAUG, make the sample software disks more compatible with the various Apple computers, and to better encourage new members to take an active part in the club.

A second change we are making is doing away with the ten-dollar penalty for late renewals. Times are tough and this change was made so we wouldn't discourage members from rejoining after their membership had lapsed. The change affects renewals only and the new member fee remains at $30.00 for the first year.

Thirdly, we will soon be doing away with the laminated plastic membership cards and changing to a heavy paper card. The new card will be good for only one year and then replaced with each renewal. The advantages include a lower cost per card; less bulge in your wallet; and the renewal stickers won't end up on your VISA or Foley's card instead of your HAAUG card. And more importantly, I won't have to hand laminate every single card. Hey, we're a computer club, we ought to be letting the computer do the work, right?!

If your address changes, please let us know as soon as possible so that we can keep our records current. You can drop us a card; leave a note at the Apple Answer desk at a meeting; leave a message to me on HAAUG Heaven, our electronic bulletin board; or call me and I'll make the change immediately.

One final item...the success of a volunteer organization such as HAAUG is based on the vitality that comes with continued growth and an inflow of new members. Our purpose is to share with and help each other to become better-informed Apple computer users. To do that we must continue to build our membership. As a member of HAAUG, YOU are our best medium of advertising simply by inviting your friends, coworkers, or others you may know who own or use an Apple computer to attend a meeting and join HAAUG.

— *NEAL SCOTT*

For those of you that were wondering, this is not a grey box

Removing Protection From *Multiplan* Worksheets

Introduction

Protecting a worksheet can be very useful. If you are absent-minded, you might accidentally type over a formula without protection. More importantly, in a protected worksheet, pressing <ENTER> automatically moves you from one unprotected cell directly to the next one, regardless of where it is. This is especially useful in the use of templates, where everything remains the same except for a few values which change each time the spreadsheet is used.

The Problem

There are, of course, drawbacks to protecting a worksheet. You cannot cut and paste in a protected worksheet (you can copy, but it is not much use without being able to paste). Also, you cannot sort the worksheet or edit your formulae. To perform any of these actions, you must choose Unprotect from the Options menu and type in the password used to protect the worksheet. Therein lies the problem with protecting a spreadsheet: while it keeps unauthorized persons from making changes, whether accidental or intentional, it can also do the same to you if you forget the password.

On pages 75 and 123 of the Macintosh *Multiplan* manual you will find Microsoft's discussion of passwords. On page 75, it states: "If you forget a password, you will not be able to remove protection from the worksheet." This means that if you have forgotten your password, or if you are making alterations to someone else's worksheet, you are locked out and must start from scratch. Personally, I have a penchant for making typos when I enter a password. This makes it a rather difficult to unprotect the document later, as shown in Figure 1.

The Solution

On page 123, Microsoft reiterates:

"If you lose your password, there is no way to reverse protection on the worksheet. " Fortunately, this is not true. Microsoft assumed that we would not take the trouble to wade through the section on SYLK, because it is possible to unprotect a document using a SYLK file. A SYLK file is an ASCII text file, which is nothing but a standardized representation of all characters by numbers, with no proprietary protection formats. This means that you can use *MacWrite* to read an existing one or write a new one. It is an simple task to create a file such as we just discussed. You can use *MacWrite* to create a file
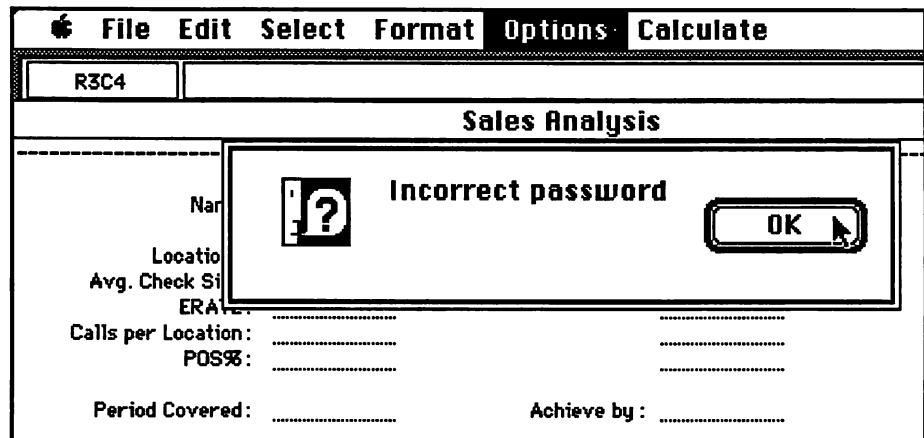

Figure 1. A possibly alarming dialog box.

which, when opened from within a protected worksheet, will unprotect it without otherwise altering it. Here's how:

1. First, enter *MacWrite*. If you need to swap disks to do so, you can clear the clipboard and save a disk swap. To do this, hold down the command and option keys while you double-click the program's icon. Now, when you open *MacWrite*, you will not need to swap disks again. Normally, the Mac would ask you to swap disks so it could transfer the clipboard to

the new startup disk. If you want to keep what is on the clipboard (or have two drives), ignore this tip.

2. From *MacWrite*, type the following two lines:
   ID;PMP <RETURN>
   E <RETURN>

3. Next, choose SAVE AS... from the File menu and when the dialog box appears, click TEXT ONLY and type in your filename (I use *The Unprotector* so I can remember it). Eject the *MacWrite* disk, insert your *Multiplan* disk, and click

Save. Always keep your SYLK files on your *Multiplan* startup disk; otherwise you will need to swap disks to read each line in the SYLK file. A one page *Multiplan* document saved in SYLK format can have over 1500 lines! Imagine swapping disks 1500 times and you will see the logic in this. *The Unprotector* only has two lines, of course, so this is not a problem.

Using *The Unprotector*

To unprotect a spreadsheet, simply choose Open... from the File menu, select *The Unprotector*, and click Open.

You might be asked if you want to save changes before opening. Respond NO as in Figure 2.

The Unprotector will not replace your spreadsheet, only _add to_ it. As _The Unprotector_ loads, you will see a dialog box informing you of your progress (Figure 3). Again, _The Unprotector_ has only two lines (see step two above).

Note that your worksheet assumes the name _The Unprotector_ after opening. Be **sure** to use Save As... to get your original name back (and keep from

SYLK, or symbolic link format, is Microsoft's way of communicating data from one program, say, BASIC, to another, say, _Multiplan_, and is analogous to DIF. Writing and using a full-sized SYLK file is extremely difficult because of the large number of options available. Fortunately, it is not necessary to understand SYLK to use the technique we are covering. All you need to know is that a SYLK record is a series of lines, each with its own **record type,** and that when it is

identify itself and an **E** line to end the transfer, otherwise, _Multiplan_ will report an error in the SYLK file and stop reading.

For our purposes, the only record types we need are **E** and **ID**. **E** is for end of file, and appears, naturally, at the end. The **ID** line must always appear first. It tells _Multiplan_ what program created the file and on what type of computer. Usually, it will say it was created by _Multiplan_ on the Macintosh. This is important because if it were not, that would change the protection scheme. The ID line as written by the Macintosh version of _Multiplan_ is: **ID;PMP;N** — meaning "ID line; Program is MultiPlan; New protection plan".

To trick _Multiplan_ into removing the protection from a worksheet, all we need to do is create a SYLK file without an **N** in the **ID** line. That way, when it is loaded, it will tell _Multiplan_ to use the **old** protection scheme; that is, ignore the global protection flag and protect a cell only if there is a **P** in the cell descriptor (in SYLK, the **C** line, one we are not concerned with). This is the **old** protection scheme used by _Multiplan_ on other machines, as well as some other Microsoft software.

In Macintosh _Multiplan_, however, the only possibilities for cell protection in the **C** line are: ;N, meaning the cell is not protected (remove cell protection option) or nothing at all (restore cell protection option). Since a cell can never have a **P** in the cell descriptor, no cell will be protected. Since this means the global protection flag will not be set, the worksheet will be unprotected. (If there is a ;P in the **O** line, the global protection flag will be set, but that can only occur from within Macintosh _Multiplan_ if you saved a **protected** worksheet in the SYLK format, which is impossible. This option is provided for users of other Microsoft software who want to create protected _Multiplan_ worksheets from within their application, say a database or accounting program.)

Conclusion

You now have a tool for unprotecting worksheets you might have thought inaccessible. Remember, next time you protect a worksheet, you
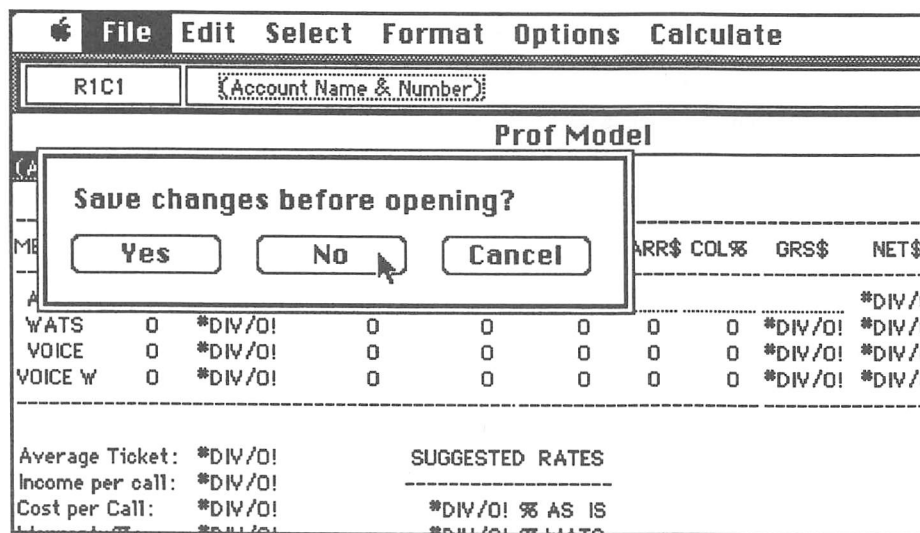
Figure 2. Opening _The Unprotector_.
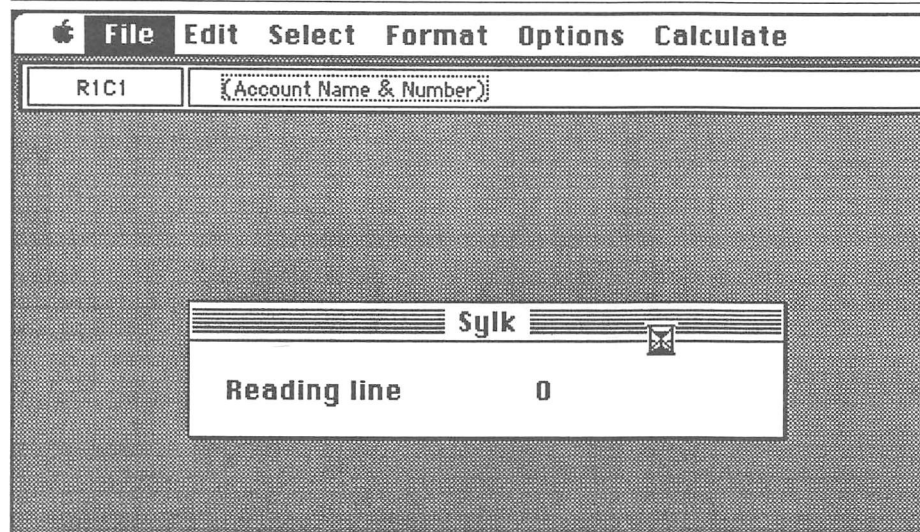
Figure 3. Screen view as _The Unprotector_ loads.

destroying _The Unprotector_) when you resave the worksheet. Actually, this technique does not really unprotect the worksheet, it creates a new, unprotected worksheet. The original worksheet's password still will not be known.

How It Works

opened, its contents are **added to** the current worksheet instead of replacing it. The **record type** tells _Multiplan_ what kind of information to find in the line. This information is added to the worksheet one line at a time. Each SYLK file must have an **ID** line to

do not need to use a password. When the Protect Document dialog box appears, just click OK or press ENTER. Then when you choose Unprotect, the worksheet will just be unprotected without any further prompts.

Also, if the only reason you protect worksheets is to provide for easy data entry using ENTER, consider this: you can select up to 20 noncontiguous cells or cell groups simultaneously. For instance, to select both R1C1 and R9C9 and use ENTER to switch between them, click in R1C1, then move the cursor to R9C9 and COMMAND-click (that is, hold down the command key and click in R9C9). Both cells will be selected simultaneously, and pressing ENTER will switch the entry area from one cell to the other. Pressing RETURN or TAB will deselect both cells. The

COMMAND-click option of selecting cells is explained, albeit poorly, on page 34 of the *Multiplan* manual. This selection method can be very useful for data entry, creating ranges of data using Fill Right or Fill Down, and for naming and formatting irregularly shaped cell groups.

I hope this discussion will prove useful to you. There is a lot left out of all the Macintosh manuals. The original Apple II manuals were very comprehensive and where they leave off, thousands of enthusiastic Apple users fill the gaps. We see the beginnings of that today with the Macintosh, but Apple designed the Mac and priced the accessory manuals in such a way that users are severely discouraged. I find this deplorable, since users are a source of so much useful information.

In my experience with the Mac, I have found evidence of poor design and even outright bugs in a large number of programs. I can say very little for the support I have received from Apple, my dealer, and Microsoft. In fact, I felt when talking to the Microsoft representative that he hardly even knew how to use *Multiplan* at all! The only useful support I have received was from users, so I hope to see more Macintosh tips in magazines such as this one, and a little less secrecy and a little more competent help from Apple and its dealers.

In the meantime, keep talking to users, read your *Apple Barrel*, and don't hold your breath!

— PETER LEMETTAIS

So You Want to Program the Macintosh...

Many people buy a computer, purchase application software, and go home and put their investment to work. These same people find that the software offerings are suitable to their needs. If the software is not, time will provide it. Other people take their computer home with the same intentions, decide what they need is not available, and decide to do something about it. This is what provides the motivation for people to tap the power of their computer through that magic known as "programming." For many, the Macintosh invites discovery of and control over its capabilities. Just starting out, these people are often greeted with the less than inspiring statement, "So you want to program the Macintosh..."

If you are truly interested in using your Macintosh to its utmost or just in playing with its graphics or sounds, don't hesitate. Begin programming now!

The next inevitable question is "How do I program the Macintosh? Everyone tells me that the Mac is too complex and only professionals can do anything with it." In part there is truth in this statement. The Macintosh is a more complex machine than the PC and more dedication is necessary to understand how the parts go together. However, the rewards from creating something that runs on the Mac are also more satisfying. Concerning the second part of the statement, anyone with enough determination can program the Macintosh. I have been programming the Macintosh for about a year. However, in that same time I have had some invaluable experiences and would like to pass on some advice on getting started.

I must say at the outset that I program in C and am partial to the freedom that it gives me. However, if you are familiar with or would just like to learn another language, there is probably a programming system for you. Generally these languages are sold as part of a "Development System." All this means is that there is usually a compiler, editor, and other necessary software components to create an executable (double clickable) application. The advice that I will give is still applicable regardless of the language you decide to use. Again, however, being a C programmer, I will be slightly biased.

Books and Reference Material

To begin programming the Macintosh you need without a doubt the now famed *Inside Macintosh*. Anyone considering investing in programming the Mac should purchase this book first. *Inside Macintosh* is to the Mac programmer as the dictionnary is to the writer. Don't be frightened off if at first it seems overpowering as there are plenty of supplementary texts to help you decipher its inner meaning. I will discuss some of these texts in a moment but first where do you get *Inside Macintosh* (referred to as IM also). Inside Macintosh comes in at least three versions. There is the older phonebook version which was printed on cheap paper and was the thickness of a typical phonebook. These are no longer printed and do have inaccuracies, but if you can find one they are usually inexpensive and will suffice. The other two versions are printed by Addison-Wesley. There is a three volume set as well as a hard-bound, one volume version. Both these versions are up to date and are usually available in bookstores such as Waldenbooks, B. Dalton's, and the Bookstop. If you are faced with the choice of getting one or the other version of the Addison-Wesley IM, I would recommend the three version set. Buy the hard-bound version later. With many a late night programming binge ahead of you, you will be less concerned about getting potato chip greased fingerprints on the less expensive version.

After Inside Macintosh, you will want to get a subscription to *MacTutor*. This is a monthly magazine packed with information concerning programming the Macintosh in just about any language you might choose. If you have a question concerning programming the Macintosh, there is probably an article addressing that subject. Back issues are available and I have included *MacTutor's* address at the end of this article.

Another set of books that has become incredibly popular and for good reason are the two volumes by Stephen Chernicoff. The first is entitled *Macintosh Revealed — Unlocking the Toolbox*. It covers such important subjects as the Quickdraw routines, memory management, and something called the Resource Manager. Do not be concerned if these don't make sense now, a few nights with this book will help considerably. Volume two, entitled *Macintosh Revealed — Programming with the Toolbox*, covers such subjects as windows, menus, dialog boxes, and a plethora of other subjects. Furthermore, Chernicoff constructs a text editor in the book. That editor is a valuable source of information and programming tactics. The programs and descriptions of routines are written in Pascal. Helpful information is also provided for assembly language programmer. For those programming in C, the translation from Pascal to C is a trivial matter. These are two books which should be in any Mac programmer's library.

Now, since I program in C and my source code is in C, I will recommend a book on the C language for those wishing to learn that language. *Programming in C* by Stephen G. Kochan is an excellent introduction to the language. The book is written with no particular machine in mind, but it

provides the necessary information to program in that language. Organized well, it is also a valuable reference tool. Many other excellent C tutorials exist on the market; invest in one of them.

Software Tools

Before I discuss a particular development system, I will recommend a few programs to have on hand. You will probably want — if not need — a resource editor. *ResEdit* (now vers 1.0 alpha) and *REdit* (unknown) are both very useful tools in their own rights. With them you can manipulate menus, windows, as well as many other parts of the Macintosh environment. *RMaker*, or resource maker, is also a piece of valuable if not necessary software. A disk editor like *FEdit* can be quite useful at times. Debuggers are a must for anyone wanting to do serious programming. Several exist. Talk to someone who has used one of them. After getting some of these pieces of software (many are in the public domain) you are ready to begin using your development system.

The Development System

Originally, this article was going to be a review of the Megamax C compiler. However, this has changed. If you are using a C compiler or other system, let me recommend that you continue to use it. If you are looking for a programming system to use, I recommend Megamax C.

I use Megamax C almost exclusively and have found it quite satisfactory for everything I have wanted to do. Furthermore, the system has quite a big following as evidenced by the numerous Megamax C programs found in technical magazines. The development system provides the user with very many nice features.

The compiler is quite fast. It creates object code which is both efficient and compact. For those occasional forays into assembly language, it provides an inline assembly facility. Inline assembly simply means that in the middle of C instructions, the programmer can include direct assembly language instructions for the Mac's microprocessor) This provides both more control and more speed to the C

programmer. Data types include the short, char, int, long, float, and double as per the C standard. Also supported are certain of the unique Macintosh data types. Full support for pointers is provided. Structures (Records in Pascal) are of course available. For those experienced in the C language, it also includes bit fields, a useful tool often absent from other compilers. The compiler can compile individual functions to be put into your program or to be placed in a library of often used routines. One nice touch is that as soon as it is finished compiling your code, it displays a dialog allowing you to either transfer to another program or to rerun the compiler thus allowing you to

---

The Macintosh is a more complex machine than the PC and more dedication is necessary to understand how the parts go together.

---

compile multiple files of source code.

The linker is also quite a nice one. A linker takes the original object code created by the C compiler and puts it together and creates a ready to run application. One of the features of the Megamax linker is that it is what is called a smart linker. This means that when it puts a program together, it will only retrieve the code that is actually referenced in your program. Therefore, if you include a function called *foo()* in your source but never explicity call it, that code will not be placed in your program. Other linkers (some only until recently) would include that code regardless. This smart capability insures smaller final executable programs. Megamax also made it quite easy for the programmer by providing special support for the creation of special types of code like that for desk accessories, windows, menus, etc. Again, like the compiler, a transfer dialog appears after the linking process.

Other parts of the system include a

good editor, the program *RMaker*, a librarian, a disassembler, a code improver, a profiler, a lot of example programs, and a batch facility. For those unfamiliar with batch programs on another computer, Megamax provides the programmer the capability to write little C-like programs. These can be written in such a way that they compile and link all of you source code, return to the editor if an error occurs, or run the program if successful. However, it is much more powerful than this because it allows you to write batch programs that might give you the status of all your files on the disk or some other useful task. Let me say now that the disassembler does not disassemble code created by other systems but only disassembles code created by the C compiler. This can, however, be quite useful when looking through code for certain details. (Disassemblers are available; *MacNosy* is probably the best known and is reasonably inexpensive).

The description that I have just given concerns version 2.1 of their system. According to Megamax version 3.0 was released. Supposedly this will prove to be faster. It will also provide full support for the new Macintosh ROMs like those in the Mac Plus and Mac 512K Enhanced. Also, they will be releasing a symbolic debugger that looks to be hot. This debugger which will only work with the Megamax system will allow the programmer to see the contents of registers and variables in his C program in a window type environment. More than this, however, is its capability to display the actual source code where an error might have occurred. Therefore, the programmer can switch between the running application and this debugger environment and see how the code is executing. This should make Megamax an unbeatable system. Also being priced at $299.00, the Megamax system is at a good price.

New Horizons

Originally, this was only to be a single article reviewing the Megamax C Development System. If I receive any encouragement this might be the first in a series of articles on how to program the Macintosh in C. I feel that this could be an interesting undertaking and

resting undertaking and I would enjoy doing it. So until next month let me just include what I would consider a good example of a person's first program for the Macintosh. I do not feel that "Hello World" programs are valid on the Macintosh. Any computer can print "Hello World." The Mac provides so much more.

— *CHRIS FLICK*

References

Programming In C
Stephen G. Kochan
Hayden Book Company, © 1983

Macintosh Revealed-Volume One
Unlocking the Toolbox and
Macintosh Revealed-Volume Two
Programming with the Toolbox
Stephen Chernicoff
Hayden Book Company, © 1985.

MacTutor
P.O. Box 400
Placentia, CA. 92670

(714) 630-3730
$30 per year, US third class
$45 per year, US first class

HAVE YOU HEARD?

Microsoft has announced that it will ship all new releases of Macintosh business software without copy protection.

Phoenix Technologies, publishers of IBM compatability software used by AT&T and others, has announced software that will allow 68000 based machines such as the Mac, Amiga, and Atari ST to run MS-DOS software without hardware modifications. This product will be available to OEMs in early 1987.

```c
#include <qd.h>
#include <qdvars.h>
#include <event.h>
#include <menu.h>
#include <win.h>

#define TRUE -1
#define FALSE0
#define NULL 0L

/*
        FIRST.C
 -A First Macintosh Application

 by C.L. Flick
 written in Megamax C
 July 1986

 The following example program is an example of
 a first Macintosh application.  Instead of an
 uninteresting "Hello World" program, I felt that
 a first program in the spirit of the Macintosh
 was in order
*/

eventrecord theevent; /* Event to be processed by program */
windowptr thewindow;  /* The window created by the program */
menuhandle themenu;      /* A menu containing QUIT */

main()
{
  windowptr whichwin; /* Which window was mouse pressed in? */
  char notdone;       /* Flag tells if we are finished */
  rect windowrect;      /* Rectangle is size of the window less
                          title bar */

  initgraf(&theport);    /* Get Quickdraw ready */
  initfonts();           /* Initialize fonts */
  initwindows();       /* and windows */
  initmenus();           /* and menus */

  notdone=TRUE;      /* Must say we are not finished yet*/
```

The First.C program listing

About First.C

The program First.C is a C program that represents in a small way the spirit of Macintosh programming. It is not meant to be a thorough and capable program and I hope it will not be criticized as such. If you are looking for a powerful spreadsheet or that super database, continue your search. It does however represent a very important concept or two about Macintosh programming.

All of the window management, menus, drawing and other activities concerned with the interface are maintained in ROM in what are called "managers." The managers consist of various routines that the programmer can call to get something accomplished. For instance, when I create a window and have it displayed on the screen, I call the Window Manager. Likewise, I interact with the Menu Manager to create the single File menu with the Quit option. Through understanding the routines and how to use them in a program, a great deal of work is handled for you. All of these routines are described in Inside Macintosh.

The second important concept embodied in the program is that the Macintosh is an **event driven computer**. Most programming on other machines is done in a procedure oriented fashion. The programmer prompts the user for something, acts upon the information when furnished and then returns with another request for the user. However, on the Macintosh, keypresses, mouse clicks, disk insertions, and the like are automatically collected in a list according to when they occurred. The Macintosh program must then retrieve these events (as they are properly known) from the list and act upon each in turn. Since the user decides when to insert a disk or when and where to click the mouse, this

```
/* Create a single menu with QUIT in it */
themenu=newmenu(255,"File");/* Create a menu called "File" */
appendmenu(themenu,"Quit"); /* Add only a QUIT item */
insertmenu(themenu,0);       /* Insert this menu in the menubar */

drawmenubar();               /* Draw the menubar */

/* Create a window and have Mac draw it on the screen */
setrect(&windowrect,40,40,472,302); /* Bad programming practice */
thewindow=newwindow(NULL,&windowrect,"My First Program", TRUE,
         documentproc,-1L,FALSE,NULL);
setport(thewindow);          /* Tell Quickdraw we will draw in
                              this window */

/* Draw some text in the window */
textfont(geneva);            /* Use Geneva font */
textsize(12);        /* at 12 points */

moveto(40,40);       /* Move pen to starting location for text */
drawstring("I am a Real Macintosh Application"); /* Draw some text */


/* Now process any user actions */
while(notdone){
    if(getnextevent(everyevent,&theevent)) {/* Did the user do something */
        switch(theevent.what){       /* What did he do? */
            case mousedown:          /* Pressed mouse button! */
                switch(findwindow(&theevent.where,&whichwin)){ /* Where? */
                    case incontent: /* Inside a window? */
                        sysbeep(3);  /* Just beep if he did */
                        break;
                    case inmenubar: /* Inside a menu? */
                        if(loword(menuselect(&theevent.where))) /*Check which menu */
                            notdone=FALSE;   /* Finished */
                        break;
                    }  /* end switch(find... */
                break; /* mousedown */

            /* Other types of events could be processed here */

            }  /* end switch(theevent... */
    }  /* End if */
}  /* End while loop */
}
```

affords much more free-dom to that user. A little more programming thought and time is required initially to write a Mac program but the finished program is usually more friendly and usable.

Since I have not really begun a series of articles on programming the Macintosh in C, I will not say too much about the associated program. It illustrates the initialization of a few of the managers responsible for windows, menus and events. Then, I create a single menu called "File" and add a single item, "Quit." A window is needed; so Presto, I create one. In the comments I state that creating a window like that in the program is in bad Macintosh programming practice. The point I was making is that since Macintosh screen size is likely to change in the future, the user should never fix that size in the program. Better programming style later. At this point, I go ahead and draw some text at the location 40,40 in the window. Feel free to change either the location or the text.

The rest of the program consists of an **event loop**. Here, the program requests any event the Macintosh might have intercepted. If a the mouse button was pressed ('mousedown' in the program), I ask where the mouse was pressed. If it was pressed in the menu, I see if it was pressed and released in the File menu. If it was, I set the variable notdone to FALSE and exit the loop and from the program. If the mouse was pressed in the window I just beep. Of course, anything could be done here. Pressing the mouse anywhere else is ignored as are all other types of events. Nevertheless, for a substantial program, the programmer would consider quite a few types of events that could occur.

— *CHRIS FLICK*

Using the Hard Disk 20 with *Turbo Maccountant*

For some time Jack Cowart, the current HAAUG Treasurer and my father-in-law, has been looking for an accounting package for the Macintosh that was truly professional quality. After trying most of the well known packages, he has decided that *Turbo Maccountant*, a $500 package by Digital, Etc., is the best from an accountant's viewpoint and is using it for keeping the HAAUG books.

First understand that Jack, although having worked with mainframes for as many years as I have known him (since 1960 or so) and with *AppleWorks* on the Apple // for the last two years, is just beginning to feel comfortable with the Mac user interface and is still learning the terminology. He is finding it difficult to work with a friendly machine for a change.

Since he has an Apple HD20, Jack naturally wanted to save his accounting data on it. The *Turbo Maccountant* manual stated that the package would work with most hard disks, and even included a brief paragraph on how to use the HD20. Since the paragraph could just as well have been written in Farsi, Jack decided to forge ahead and use the program. The problems started when Jack tried to save data. No matter what he did, *Turbo Maccountant* kept telling him to insert the Master Data Disk in the floppy drive. So much for the friendly user interface!

When I heard about his problem, I tried to help Jack over the phone. At first I thought his lack of familiarity with the Mac was the problem and kept telling him to use the Save As... command to save on the hard disk. He kept insisting there was no such command and I kept insisting there had to be since all programs that saved to disk had that command.

Following the June HAAUG meeting, I followed Jack home to show him how easy it was to save on the hard disk. I tried to save and it asked for the Master Data Disk! I then read the paragraph in the manual and could not make much sense out of it since I don't read Farsi. So I decided to use the old trial and error method to determine what procedure to use. After three hours of heavy effort and several false starts, I was finally successful in saving *Turbo Maccountant* data on the HD20. What a pity that a good accounting package was made difficult for want of clear, step-by-step instructions for using the HD20. And since many accountants may not have the experience, time, or patience to work through all the possible steps, they will probably give up and blame the poor Mac for their problems.

The following is a summary of the procedure I worked for storing data on an Apple HD20. They MUST be followed to the letter in order to set up a new *Turbo Maccountant* account on an Apple HD20. Steps 6 through 9 should be followed to work with an existing account.

1. Copy the Company Data folder from the *Turbo Maccountant* Master Data Disk to the hard disk by dragging its folder icon from the floppy disk icon to the hard disk icon. This will copy the Company Data folder and all its files to the hard disk.
2. Rename the Company Data folder MASTER FOLDER.
3. Create a new empty folder and call it Company Data. Move the MASTER FOLDER folder into the new Company Data folder. Note that there should be only one Company Data folder.
4. To set up a new company's books, duplicate the MASTER FOLDER folder. It will be automatically named Copy of MASTER FOLDER. Rename the duplicate folder with the new company's name.

5. Place this and all company folders in the Company Data folder.
6. To work with a particular company, first move its folder to the desk top. This is done by
   - Opening the Company Data folder,
   - Pointing the cursor at the company's folder,
   - Pressing the mouse button,
   - Dragging the folder to the right side of the screen.
7. Close the Company Data folder by pointing to the close box on the Company Data folder window and pressing the mouse button.
8. Double click on the *Turbo Maccountant* icon to run the program. It will automatically open the books for the company whose folder was placed on the desk top. Be sure that the Company Data folder is closed and that there is only one company folder on the desk top before running *Turbo Maccountant.*
9. When finished with the company's books, exit *Turbo Maccountant*, return to the desk top, and replace the company's folder in the Company Data folder.
10. To work with another company's books, you will have to repeat Steps 6 through 9.

If you are asked to insert the *Turbo Maccountant* Master Data Disk at any time, you didn't do it right. Have patience and start over with Step 1.

Happy accounting!

— MIKE KRAMER

Using *Silicon Press* and *File* Together

The Macintosh along with a few popular software programs makes an excellent tool for preparing customer databases as would be used in a small neighborhood video store. This database can be used with a word processor and a label preparation program as a way to produce advertisements which are both effective and inexpensive.

Although most any database or word processor would do just as well; I used Microsoft *File* and *Word*. Additionally I used *MacDraw* and Silicon Beach's *Silicon Press* for this simple project.

At the video store where I work part-time there are about 750 customers with rental memberships. Each customer has a security deposit along with address, phone, and identification numbers stored alphabetically in an index card file. This file can become unwieldy to search for a particular customers record during peak rental periods. The solution which I suggested to the store owner was a customer database with printouts both alphabetically by last name and numerically by membership card number. After I produced several sample *File* printouts from the Mac for his approval, a financial arrangement was quickly reached to prepare the entire database.

Using *File*, the only information which is required initially are the names of the pertinent fields (such as last name, first name, card number, phone number, address, etc). These fields are then inserted into a new datafile window. The fields can then be formatted individually to your suiting (note - phone number fields are best left as text fields since *File* doesn't allow dashes in number fields). At this stage it is important to assign the fields which will most likely be used for sorting the data as index fields. This allows *File* to store their information in a special index record, from which a unique algorithm will sort the records

very quickly. The next step is a quick review of your input data. Using the form design window, a data entry form can be quickly designed whereby each of the prescribed fields are arranged in a manner which is most consistent with the form containing the data you will be entering. If necessary, static text items can be added to each record which can serve as prompts for data entry.

Entering large amounts of data is never fun but it is as painless as is possible with *File*. Each successive entry field within a record can be reached by tabbing to it; shift-tab will back up one field at a time. After the last field a tab or a return will advance *File* to the next record.

Copying high quality ImageWriter printouts usually improves their quality somewhat as the copying process tends to add a smooth tone to the print.

After entering the customer names into separate records of the database, I find a printout the most convenient means to check the data for entry errors. In the days of keypunch machines and computer cards this was know as "verification." Using the Print Records command from the file menu you can print the entire database very quickly on the ImageWriter in draft mode. This will leave out the fancy fonts, check boxes, or underlining but is otherwise quite legible. At this stage it is a good idea to make a backup copy of the disk with your database just in case a disk gremlin strikes. Having entered and checked the database you can again use

the form design window to create a unique form for a final printout of the records. At this point it is easy to add special formatting such as underlining, bolding, or to change font type and size. Also new fields can be added to every record for static text, check boxes, or special comments. *File* allows you to store and name these forms with your database.

VIDEO TODAY
Rent 1 Movie get 1
FREE
Exp. 5-31-86

Figure 1. Coupon Example

The next major step in my project was creation of a discount coupon for the advertisement mailers in *MacDraw*. This coupon consisted of a simple graphic with text describing the discount and an expiration date (see Fig. 1). This coupon could have been created as easily with *MacPaint* however *MacDraw* stores it's images as quickdraw instructions which are more compact on disk, prints faster, and with higher quality when pasted into a word processor document.

I created a general advertisement letter with *Word* and the coupon was pasted in via the clipboard as a hook for the customers. This letter could have been used as a mail merge document however the amount of time required to print out 750 individual letters in high quality would be immense not to mention the expense in ribbons. Instead one generic member letter was printed repeatedly in high quality until we were satisfied with the text positioning and content. This was then copied on a pastel colored paper inexpensively at a copy center.

Copying high quality ImageWriter printouts usually improves their quality somewhat as the copying process tends to add a smooth tone to the print. Of course, the best way to go would be copying a LaserWriter printout.

Printing a mail merge with records from a File database is actually a lot less complicated than it may sound–here's an example.

1. First using the form design window and the list helper option of

required header record for the merge.

3. Open your *Silicon Press* address label file (or other target mail — merge document) and give it the appropriate instructions for finding the text file you just created (file name and location disk). In Silicon Press if your datafile is 25k or less then the data will appear in the data window and you may choose to review or edit the data before starting to print (see Fig. 3). Text files which are greater than 25k can be opened with most word pro– cessors and edited. However be sure to save it as text only file when your're done.

4. Set up the label stock in your printer, choose the Test print option of Silicon Press to properly align the labels. Then print the mail-merge with the Print option.

For best results use a clear legible font and choose the high quality printing option.

I have found that this form of advertisement is very cost effective with many of the customers redeeming of their coupons. The most expensive part of the process is U. S. postage. The envelopes can be stuffed and labels attached during slow periods at the store by clerks.
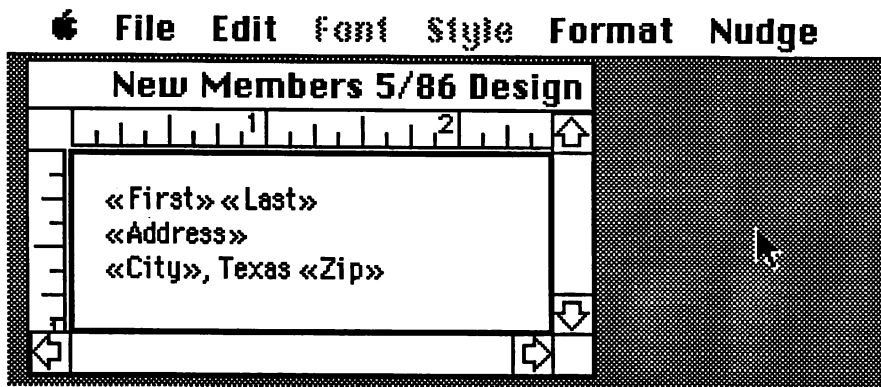

Figure 2. Address label from *Silicon Press*

The last major step was designing return address and customer address labels in *Silicon Press* to be printed as a mail- merge document. The return address label could be omitted by using printed envelope stationary. The customer address labels were created with fields for first and last name,

File, arrange only the fields containing data that you wish to merge in order from left to right on the new form. Note they should be ordered as you will call for them in the merge. Move any fields containing data that you don't need (check boxes, text, or whatever) off the form so that they won't be shown.
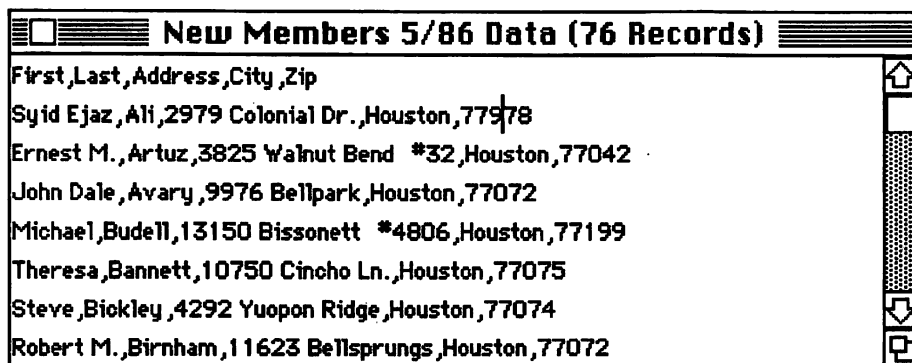
— *GREG NEW*

Figure 3. Mail-merge text file.

address, and zip code. The state was entered as text on the label of the same font as the other entries in the appropriate location (see Fig. 2). Each mail merge field must be surrounded by special international quotation marks (<<, >>) which tell the mail-merge application (*Word* works the same way) that these characters are a field name and will be replaced in the printout with other data from a specified text document.

2. Choose Save records as from the file menu and two dialogs boxes will follow: the first dialog will ask if you want to save the current form with the database. The second dialog box will ask: the file name and disk to save it on; plus to save the file as a text file (Microsoft Print-Merge) or as a File database? Choose the print-merge option and your file will be saved as text only with commas separating each field. File also automatically creates the

Meeting Dates
1986-87

———————•———————

December 20, 1986
January 17, 1987
February 21, 1987
March 21, 1987
April 18, 1987
May 16,1987
June 20,1987
July 18,1987