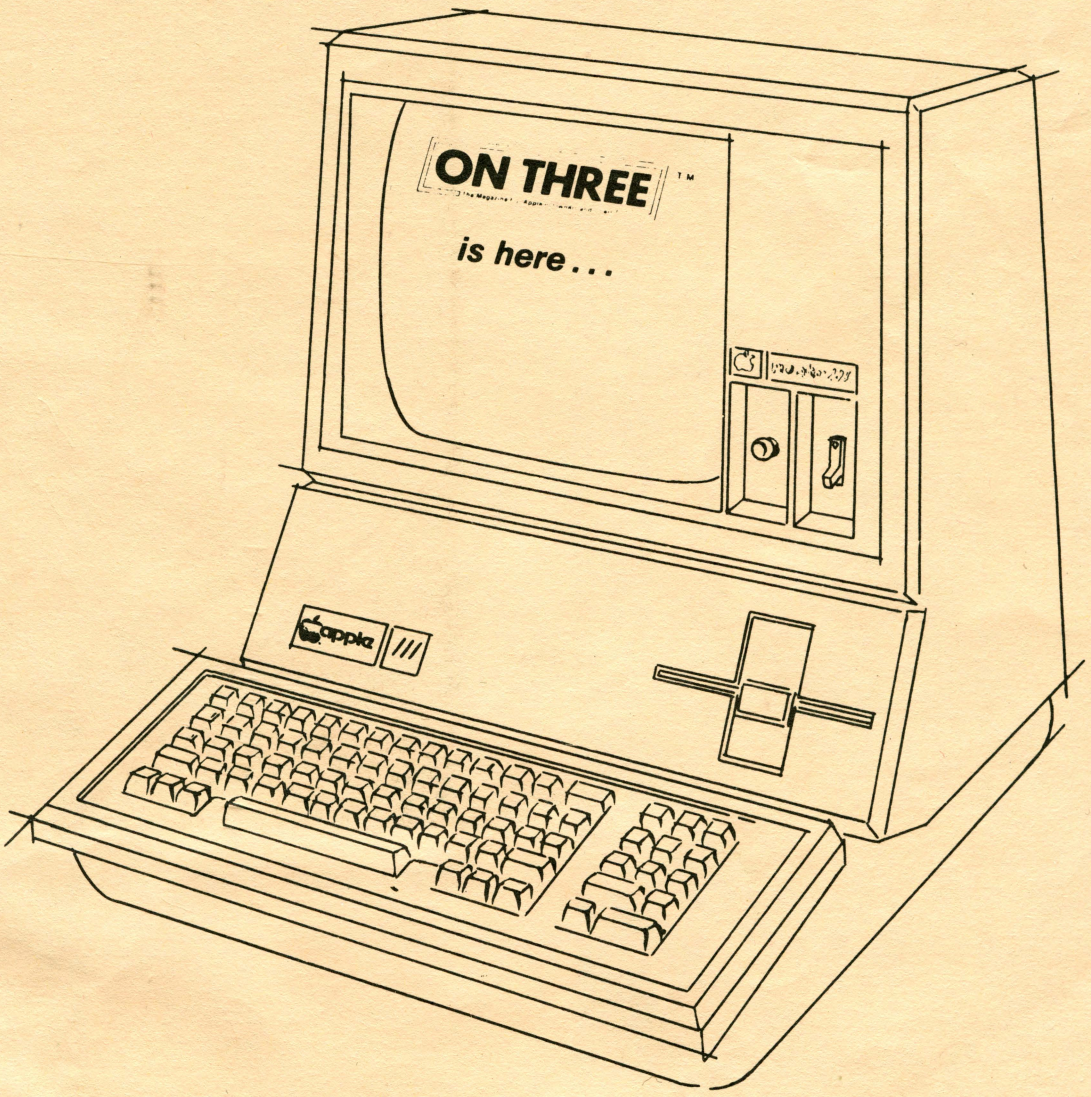
ON THREE T.M.

The Magazine For Apple III Owners and Users

Important Apple /// Information Inside . . .
Including—

- **512K Memory Upgrade Information**
- **Low Cost Floppy & Hard Disk Drives**
- **Apple /// Support Questions Answered**



ON THREE

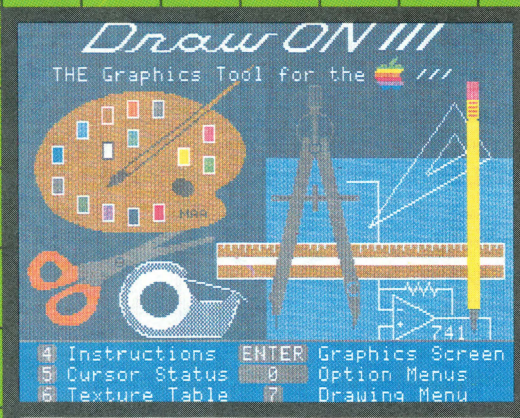
TM

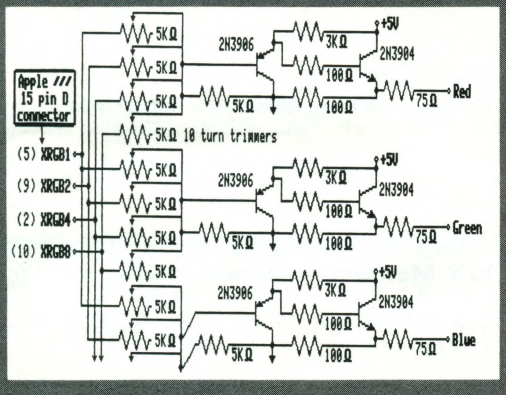
The Magazine For Apple III Owners and Users

Volume 2 - Number 2

$4.00 U.S.

- **512K Memory Upgrade Information**
- **Low Cost Floppy & Hard Disk Drives**
- **Draw ON Is Exciting!**
- **Business Basic Program Lister**
- **User Groups Revealed**
- **Reviews, plus much more!**





ON THREE Presents... 512K Memory Upgrade For Your Apple ///

For everyone who has wished for more memory in their Apple /// or Apple /// plus—here it is!

The ***ON THREE*** 512K Memory Upgrade is a simple replacement memory board, and all programs running under SOS will work with it. How would you like your *VisiCalc* (regular and Advanced version), */// E-Z Pieces, Word Juggler, Business Basic* and others to have about 450K of memory to work with?

If you have ever run out of room with a huge spreadsheet model, the **512K Memory Upgrade** is just what you need. Just think of the forecasts that you could create with your *512K Apple ///*! If you have ever filled memory while working with /// E-Z Pieces, the **512K Memory Upgrade** will give you the room you need. The extra memory combined with the compact spreadsheet that /// E-Z Pieces produces will give you room for the largest model that you can imagine.

How would you like to be able to type PRINT FRE in Apple /// Business Basic and the response that comes back is: 467452. That's over 456K of available space in Business Basic. With that much room, the Apple ///'s version of Basic is one of the most powerful on the market.

Have you ever had problems putting some program on your ProFile hard disk while running under *Catalyst*? Since Catalyst takes up a good portion of memory, there are times when there isn't enough memory to go around. Many programs will not run at all under Catalyst because of memory limitations. Some programs will run only on minimal Catalyst systems. That means no spooling, no special purpose drivers like ***ONTIME*** or the clock/calendar, and a lot of dynamic driver loading. Who needs problems like this? With the **512K Memory Upgrade**, your problems with programs not fitting while running under Catalyst will be solved. You will be able to put ***Lazarus*** and ***Draw ON*** under Catalyst and have your Draw ON pictures being printed out while you undelete a file with Lazarus or work on your word processing with Applewriter or Word Juggler.

Included in the **512K Memory Upgrade** package is a diskette that will check your memory for any errors, thus giving you confidence that your **512K Memory Upgrade** works correctly. In addition to the confidence program, we are including a new disk drive for your Apple ///—a *RamDisk* drive! This disk drive is simply a new device driver that takes a portion of your 512K Apple ///'s memory and uses it as an ultra-fast disk drive. You can use this drive to store and retrieve information at lightning fast speeds. If you are using one of the popular program switching utilities like Selector /// or Catalyst, you can use the RamDisk for quickly storing information that you need to transfer from one program to another.

The **512K Memory Upgrade** is the single most exciting thing to happen to the Apple /// in a long, long time. Using state of the art 256K memory chips, the board is very simple to install and even easier to use. The **512K Memory Upgrade** will NOT take up an expansion slot as it is a simple board swap-out. Just keep on using your existing programs—you don't have to change them! Visicalc, Advanced Visicalc, /// E-Z Pieces, Word Juggler, Business Basic, Pascal, Catalyst, Selector /// and many other programs will automatically have about 450K of memory to work with.

The ***ON THREE*** **512K Memory Upgrade** is priced at *only* $949 and we will begin shipping on or before December 15th. Orders will be shipped on a first come / first serve basis. Thus, the first ones who order will be the first ones who receive the memory upgrade.

SEND NO MONEY NOW! To order the **512K Memory Upgrade**, simply send a signed purchase order stating that you will accept a COD delivery for $999 plus $10 postage, handling and insurance. When we receive your old memory board, we will mail you back a check for $50. As soon as you receive your **512K Memory Upgrade** and install it in your Apple ///, return your old memory board to us and you will get a rebate of $50.

Send your ***ON THREE*** **512K Memory Upgrade** purchase orders to:

ON THREE
Attn: **512K Memory Upgrade**
P.O. Box 3825
Ventura, CA 93006

Quantity discounts:

1-4 memory boards:	$949 apiece
5-9 memory boards:	$909 apiece
10+ memory boards:	$869 apiece

Add $10 per memory board for shipping, handling and insurance.

The **512K Memory Upgrade** includes...

1) An **Apple /// 512K Memory Board** using state of the art 256K memory chips.
2) The **Confidence Memory Program** which will ensure that your **512K Memory Upgrade** is working correctly.
3) The .RAM ultra-fast **RamDisk** drive.
4) **Complete** instructions for installation and use.
5) A full **90-day Warranty**.

* For purchasers of 5 or more of the 512K Memory Upgrade, please write for information on our on-site installation program. If you would like, we can come to your location and install the **512K Memory Upgrade** for a nominal fee per machine.

* The ***ON THREE*** 512K Memory Upgrade can only be used on a 256K Apple /// or an Apple /// plus. If you have an old 128K Apple ///, you must first upgrade to 256K before being able to upgrade to 512K of memory.

* Per our usual policy we will give schools and government offices NET-15 terms. We cannot extend credit to individuals or companies for the **512K Memory Upgrade**.

ON THREE

Volume 2 Number 2

Editor/Publisher:
Bob Consorti

Comptroller:
Joseph Consorti
Logistics Director:
Janet Schanz
Interior Artwork:
Virginia Carol
Mailing Cover Artwork:
William V. Padula
Cranford, New Jersey

Typesetting Services:
SuperSet Graphics
Woodland Hills, California

Printing Services:
California Offset Printers, Inc.
Glendale, California

/// TABLE OF CONTENTS ///

/// FEATURES ///

/// DEPARTMENTS ///

A note to our subscribers: As of November 1, 1984 most of the prices for ***ON THREE*** products and services have changed. After October 31, 1984 no orders will be processed at the old rates. We were able to hold our prices steady for almost two straight years, but due to increased costs, some of our product prices have changed.

Attn: All current and potential *ON THREE* authors. As of November 1, 1984 the per page prices have changed to $25 per printed page, exclusive of ad copy.

ON THREE now accepts **Visa**, **MasterCard** and **American Express** orders. To place an order over the phone call (805) 644-3514 direct.

ON THREE - The Reference Source for the Apple /// is published somewhat monthly by ***ON THREE***, 5550 Telegraph Road Suite B-4, Ventura, California 93003.

For a copy of our Author Guidelines, please send a self-addressed stamped envelope (20 cents) to the above address.

Subscription information: U.S. - $40 for 12 issues. For First Class Mail, please remit an extra $10. Foreign subscription info:

Canada, Mexico, A.P.O., F.P.O $54
Outside U.S., Canada & Mexico (Air Mail) $61

All back issues have been reprinted and are available for $5.00 each. please indicate on the order form which issues you want.
Postage For Software & Hardware:
Continental U.S. rates listed with products.
Canada & Mexico—Double the listed U.S. postage rates.
Outside U.S., Canada & Mexico—Triple the listed U.S. rates.

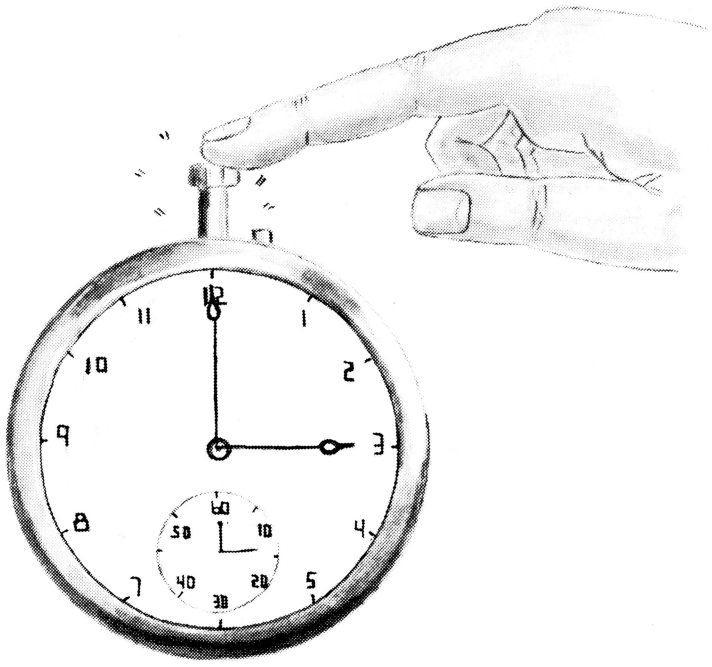
All funds MUST be remitted in U.S. dollars drawn on a U.S. bank or funds in your native currency drawn on a bank in your country.
Group purchases must have one mailing address. Please write to the above address for information on volume pricing and terms.
Return postage must accompany all manuscripts, drawings, and diskettes submitted if they are to be returned, and no responsibility can be assumed for unsolicited materials.

All letters sent to ***ON THREE*** will be treated as unconditionally assigned for publication and are subject to ***ON THREE***'s right to edit and comment editorially.

ON THREE Presents . . .

ONTIME Features:

ONTIME Clock Driver



1) Time & Date displayed whenever YOU want it!
2) Works with ALL programs!
3) Displays the Day, Month, Year Hour, Minute and Second.
4) Stop Watch/Timer Mode.
5) An unbelievable enhancement to your Apple /// system.

Calling all of you Time-Conscious professionals: Before ONTIME™ the only way you could find out the time and date from within some programs is if you looked on your watch!

ONTIME, the incredible new product from ***ON THREE*** Magazine, will conquer this problem by giving your Apple /// a great new feature: With the ONTIME clock driver installed you can display the Time & Date whenever YOU want it!

A simple keystroke combination will turn the clock display on, another will turn it off. One more turns on a stopwatch that can count down by tenths of a second! This is extremely useful in monitoring how long you've been on the telephone, or using a computer service like the Source.

The best part about ONTIME is that it operates in the 'backround' - continuously displaying the Time & Date while you are doing something else. Thus you can be working on your spreadsheet, typing a letter with your word processor or even printing out a financial report and ONTIME will continuously update the Time & Date information on the screen.

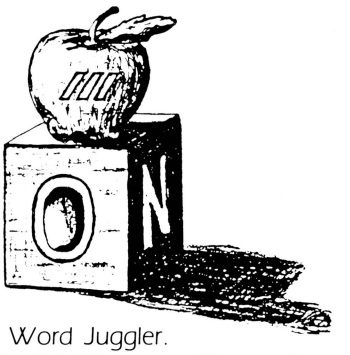
Completely compatible with all Apple /// programs that run under SOS, ONTIME is sold exclusively through ***ON THREE*** Magazine for only $39.95. Please add $2.00 for shipping and handling, California residents must also add 6% sales tax.

ONTIME requires an ON THREE O'Clock clock/calendar or equivalent to work.

Special Deal! For a limited time, you can purchase the ONTIME - ON THREE O'Clock combination for only $79.90 plus $4.00 shipping & handling. That's ten dollars off! If you have been waiting for a reason to get a clock - ONTIME is it!

The Editor's Block:

Bob Consorti



For anyone who hasn't heard—Apple Computer, Inc. cancelled future development on the Apple /// (and ///plus) in May of this year. *Future Development* is their euphemism for production and sales. **They have cancelled the Apple ///!** After you catch your breath, you will probably ask—why? To answer that one I'd have to fill you in on about five years of egotism and stupidity.

Before I go into the reasons for Apple cancelling the Apple ///, let me assure you of one thing: ***ON THREE*** will continue to support you in all of your Apple /// needs, regardless of Apple's decision. In many respects we can now do even more to help you. There are a number of products that we are just coming out with that will help you become even more productive. Before Apple's cancellation of the machine we could not release such products.

512K Of Memory

Our biggest announcement is the **512K Memory Upgrade**. For everyone who has wished for more memory in their Apple ///—here it is! A simple replacement memory board—all programs running under SOS will work with it. How would you like your *VisiCalc* (regular and Advanced version), */// E-Z Pieces*, *Word Juggler*, *Business Basic* and others to have about 450K of memory to work with?

If you have ever run out of room with a huge spreadsheet model, the **512K Memory Upgrade** is just what you need. Just think of the forecasts that you could create with your *512K Apple ///*! If you have ever filled memory while working with /// E-Z Pieces, the **512K Memory Upgrade** will give you the room you need. The extra memory combined with the compact spreadsheet that /// E-Z Pieces produces will give you room for the largest model that you can imagine.

On an Apple //e, the Appleworks program (/// E-Z Pieces for the //e) gives you only 10K of space to work with. With an optional expansion memory board, this can be raised to only 55K. On the Apple /// we have over 8 times the workspace! Which machine would you rather be working on?

How would you like to be able to type PRINT FRE in Apple /// Business Basic and the response that comes back is: 467452. That's over 456K of available space in Business Basic. With that much room, the Apple ///'s version of Basic is one of the most powerful on the market.

Have you ever had problems putting some program on your ProFile hard disk while running under *Catalyst*? Since Catalyst takes up a good portion of memory, there are times when there isn't enough memory to go around. Many programs will not run on Catalyst at all because of memory limitations. Some programs will run only on minimal Catalyst systems. That means no spooling, no special purpose drivers like ***ONTIME*** or the clock/calendar, and a lot of dynamic driver loading. Who needs problems like this? With the **512K Memory Upgrade**, your problems with programs not fitting while running under Catalyst will be over. You will be able to put ***Lazarus*** and ***Draw ON*** under Catalyst and have your Draw ON pictures print out while you undelete a file with Lazarus, or work on your word processing with Applewriter or Word Juggler.

Included in the **512K Memory Upgrade** package is a diskette that will check your memory for any errors, thus giving you confidence that your 512K Memory Upgrade works correctly. In addition to the confidence program, we are including a new disk drive for your Apple ///—a *RamDisk* drive! This disk drive is simply a new device driver that takes a portion of your 512K Apple ///'s memory and uses it as an ultra-fast disk drive. You can use this drive to store and retrieve information at lightning fast speeds. If you are using one of the popular program switching utilities like Catalyst or Selector ///, you can use the RamDisk for quickly storing information that you need to transfer from one program to another.

The **512K Memory Upgrade** is the single most exciting thing to happen to the Apple /// in a long, long time. Using state of the art 256K memory chips, the board is very simple to install and even easier to use. The memory upgrade will NOT take up an expansion slot, as it is a simple board swap-out. Just keep on using your existing programs-you don't have to change them! The 512K Memory Upgrade is priced at *only* $949 and we will begin shipping on or before December 15. Please see the inside front cover for information on ordering this exciting new product.

Selector ///

Question: What works better and easier than Catalyst, looks like Lazarus, and costs less than Catalyst?

Answer: ***Selector ///***, the new program switching utility from ***ON THREE***.

For those of you who are dissatisfied with the way Catalyst works, ***ON THREE*** will shortly have available a new utility program for the Apple ///. ***Selector ///*** will allow you to place most all of your programs on your hard disk (or other disk drive) and switch between them by choosing them from a menu. It will even allow you to place your copy-protected programs such as *Applewriter* and *Visicalc* on your hard disk. The best part? It doesn't destroy your original disks! You won't have to worry about working without any back-up copies as Selector does not destroy your original copy-protected program disks.

Selector follows the *Pop-Up* menu user-interface of ***Lazarus ///***. For those of you who like this way of showing menus, Selector is just one of many upcoming products from ***ON THREE*** that will follow this way of showing you information. If you ever have had problems putting some programs on your hard disk under Catalyst, Selector will wash them away! You will now be able to put large programs such as Draw ON and Lazarus on your hard disk and switch between the two with no problems. Selector makes putting and *using* programs on your hard disk much easier.

This utility will be priced at only $99 and will work with any disk drive that you can attach to your Apple ///. You can store

your programs on your Micro-Sci A143, an Apple ProFile, or our line of hard disk drives. Selector doesn't care about what kind of disk it is working on—it just works! We should start shipping this product by the time you get the next issue, so look for the ads in the next issue of ***ON THREE***.

Hard Disk Drives

For the last two years we have been getting letters from people asking for good quality, low cost hard disk drives for their Apple ///'s. We have finally found a line of hard disk drives that meets these expectations. If you have filled your 5 megabyte ProFile and are looking for more storage space, how about a 10 megabyte drive? If that isn't enough we can supply you with a 20, 30 or 40 megabyte drive. With that much storage you are going to need a method for reliable backups. A streaming tape backup unit will also be available that can backup the hard disks in less than 10 minutes.

All of these drives (and the tape backup unit) are great quality items. Following our commitment to fantastic value for the money, these drives will be priced very aggressively. For less than the price of the 5 megabyte Apple ProFile you will be able to get our 10 megabyte drive. Our higher capacity drives will have similar low prices.

Using one of the popular program switching utilities like Selector ///, you can place all of your applications programs on any of our hard disk drives and switch between the programs without restarting your Apple ///. Faster than the ProFile, these drives offer the best ratio of storage space per dollar spent. If you have been thinking about getting a hard disk drive for your Apple ///, or if you already have a ProFile 5 Megabyte drive and have filled it, our line of hard disk drives are the ones for you. By the time you get this issue we should be shipping these powerful & low cost drives, so please call or write for information on placing your order.

Draw ON /// & Mouse

In response to popular demand, ***Draw ON ///*** now supports the mouse. Draw ON is the first program for the Apple /// that works with the mouse. Easier and faster than the joystick, the mouse makes working with Draw ON even simpler. However, since Apple decided to cancel the Apple /// there will not be an Apple /// mouse. Don't despair, we have already worked around the problem! Draw ON will work with the *Apple //e mouse*. If you simply purchase the Apple //e mouse, we supply the instructions for enabling you to use it with Draw ON ///.

Just before the Apple /// was cancelled, we were working with Apple to bundle the Apple mouse with Draw ON. The interface card and mouse device were going to be the same as the ones in the Apple //e mouse, so there is no need to worry about compatibility problems. All you have to do is follow our simple instructions and you will be able to have the mouse control Draw ON.

For those of you who already have a parallel interface card and who do not want to buy the *Pkaso*, we have a version of Draw ON coming out for you! We are doing our best to support as many printers and interfaces as possible. If you have an *Apple Dot Matrix Printer* (DMP), or a C.Itoh 8510 and any interface card (UPIC, etc.) we will soon be shipping a version of Draw ON that works with your configuration. Even if you have an *Apple Imagewriter* connected through a serial port, you can now use Draw ON.

Draw ON /// Contest

How would you like to make some money from the pictures that Draw ON creates for you? ***ON THREE*** is having a contest for your best picture, font or shape. If you would like to earn some extra cash, just send those files in! For further information please see the article *Why Is Everyone So Excited About Draw ON?* later in this issue.

Draw ON is currently the only program on the Apple /// that works with the mouse. How would you like every other program for the Apple /// to be able to use the mouse? Wouldn't /// E-Z Pieces be better with a mouse? How about Applewriter, or even VisiCalc?

ON THREE is currently developing a product that will enable you to use the mouse from within all existing programs. Simply change one driver and you can use the mouse. No complex program patches or other fixes are required, you can just start using the mouse! We will keep you informed on this product, and all other new products as they become available.

Apple /// Cancelled

In March of this year, Apple decided to cancel the Apple ///. After five years of mismanagement and internal petty rivalries, they decided to stop production and sales of the machine. Even though sales had started to take off (and possibly because of this) Apple felt that the /// was a dead weight. The line that Apple will give out to those persistent enough to coax an answer is that the Apple /// product line was losing money for Apple, thus they had no other choice but to cancel the machine.

The truth of the matter is that if the Apple /// was an independent company, it would be in the *100 million* dollar range. That's about the size of Compaq—the very successful makers of an IBM portable clone. The profit ratio was better than twice that of the rest of the company. So, while accounting for only 10% of Apple's sales, the Apple /// brought in about 20% of the profit. At the height of employment the Apple /// group only had about 15 people (including sales personnel). Do you honestly believe that Compaq consists of only 15 people!

Since the introduction of the Apple /// certain forces within Apple have tried (with considerable success) to damage the Apple ///. Product after product have been withheld or purposefully killed. Does anyone else know that there is a fantastic **FORTRAN** for the Apple ///? Full ANSI FORTRAN 77 is available on your Apple ///. This implementation is one of the best that I have ever seen. Stand-alone FORTRAN 77 programs can be used on your Apple ///. You can also create FORTRAN sub-programs and link them into your main Pascal programs. Likewise, you can create Pascal sub-programs and link them into your FORTRAN programs.

If you are in a scientific field and have been waiting for a FORTRAN package for the ///, be prepared to wait a while longer. Apple still refuses to believe that there are any *users* out there who need FORTRAN on their Apple ///. Currently, the only way to get a copy of FORTRAN is to become an Apple certified developer and beg for a copy from technical support. Just before the cancellation of the Apple /// I thought that we had closed a deal with Apple that would allow ***ON THREE*** to distribute

FORTRAN. Unfortunately, as soon as the /// was gone—so was the deal. If you need a copy of FORTRAN for your ///, write a letter to both Apple and ***ON THREE***. As we get your mail we will try and convince the rather thick heads at Apple to release this fine product.

If you ask anyone who owns or uses an Apple /// they will tell you that the /// is a fine machine. So why did Apple cancel it? Apple recently announced the Macintosh and the //c (the portable Apple //). Both of these machines are extremely inexpensive to produce and generate large profits for Apple. Apple wants people to buy their highest profit items, thus they are pushing the //c and Mac.

The industry is still having problems with the shortage of Apple //e's. This shortage is an artificial one, created by Apple so that people would have no choice but to buy //c's. About the time that people who were going to buy Mac's realized that there wasn't going to be any software for the machine for at least a year, they started buying Apple ///'s again. Sales of the /// over the last six months have been higher than at any other time.

It is my opinion that the decision makers at Apple felt that the ///, while not significantly hurting Mac sales, interfered with that machine. Because of this and the previously mentioned rivalries—the /// was dropped. One interesting note about the decision to cancel the machine: Apple says that they will continue to support those who have questions or problems with the machine!

While they have sufficient spare parts to service defective Apple ///'s for years to come, don't hold your breath on the question of support. When an Apple /// user has a problem that the average dealer can't answer (anything greater than "Where's the power switch?"), the question ends up at the Apple /// group inside Apple. Now that there isn't an Apple /// group, who's going to answer the questions? I still have not been able to get a reasonable answer to that one!

As the decision of Apple's cancellation of the /// became public knowledge, we began getting calls from Apple /// owners all over the world. Without exception they say that they love the Apple /// and will stick with it. Because you are going to continue to use your Apple ///, we will continue to support you. ***ON THREE*** will keep on providing you with the best products at the lowest possible cost.

While I could go on for a few dozen pages about the cancellation of the ///, I don't have the time or inclination to re-hash old problems. I've always been an optimist and this doesn't change that. ***ON THREE*** will continue to support Apple /// users in all of their endeavors, and in many ways we can now do even more to help you.

The discussion of Apple's cancellation of the /// will no doubt go on for some time to come. Though we will continue to publish these discussions, I've had enough for now. It's time to sit back, relax and enjoy the latest issue of ***ON THREE*** — The Reference Source For The Apple ///.

Continued from page 24.

```
16    REM
17    REM ************************************************************
18    REM
20        GOTO 5000
45    REM ************************************************************
46    REM
47    REM PRINT-A-LINE SUBROUTINE
48    REM
49    REM ************************************************************
50        PRINT TAB(LM);B$:
              LC=LC+1:
              M=0:
              IF LC>PL THEN GOSUB 6000
60        RETURN
```

```
90    REM ************************************************************
91    REM
92    REM MAIN PROGRAM BODY -- READ A LINE; DETERMINE IF IT IS A
 *  (REM) REMARK; IF NOT, THEN CHECK THE PROGRAM LINE FOR COLON
 *  (REM) SEPARATORS AND LENGTH; FINALLY, PRINT THE LINE.
93    REM
94    REM ************************************************************
100       INPUT$4;A$
110       L=LEN(A$):
              Q=0:
              IF L=0 THEN 100
120       A=INSTR(A$,S$,2):
              B$=LEFT$(A$,A):
              N=A+1
130       B$=S4$+B$:
              B$=RIGHT$(B$,6)
140       A$=B$+RIGHT$(A$,L-A):
              M=LEN(B$):
              N=M+1:
              L=LEN(A$)
150       C$=MID$(A$,N,1):
              IF C$=S$ THEN B$=B$+C$:
              N=N+1:
              GOTO 150
160       C$=C$+MID$(A$,N+1,2):
              N=N+3
170       IF C$="REM" THEN 1000
180       B$=B$+S4$+C$:
              M=LEN(B$)
190       FOR I=N TO L
200         C$=MID$(A$,I,1):
              B$=B$+C$:
              M=M+1
210         IF C$=Q$ THEN Q= NOT Q

   FILE = /PRO/A3LIST.L                1 Sep 84 11:43 AM    PAGE 2

220         IF Q THEN 240
230         IF C$=":" THEN GOSUB 50:
              B$=S8$:
              M=LEN(B$)
240         IF M<LL THEN 370
260         FOR R=LL TO PM STEP-1
270           A=INSTR(PU$,MID$(B$,R,1),1):
              IF A>0 THEN A=R:
                    :
                    :
```

Continued from page 11.

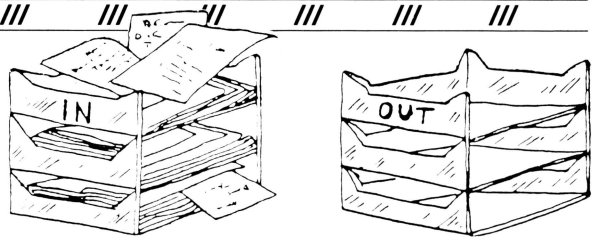
using Apple Works (Apple // version of /// E-Z Pieces) they only had about 55K of memory to work with? On the Apple ///, /// E-Z Pieces has about 180K of available space. Sitting side by side, they would have sold a heck of a lot of Apple ///'s.

Could you imagine if Apple came out with the long hoped for 512K Apple ///? A program like /// E-Z Pieces would have over 440K of available space! Some computer manufacturers would give their first born son to have the right to market such a computer—but not Apple. They are already a billion dollar a year company, but I guess they never want to become a two billion dollar company.

Since haba systems started to announce (not deliver) /// E-Z Pieces almost a year ago, Apple made the decision to cancel the Apple /// even earlier—long before the official announcement. The strategy that Apple has pursued of hurting the /// will, of course, cause many people to get angry when they become aware of the truth.

All of Apple's problems concerning the /// don't really matter anymore. With Apple out of the picture, companies developing products for the /// will have a free reign. *ON THREE* is currently developing a low cost 512K memory upgrade board and many other products for the ///. As we are ready to begin shipment of these products, we will notify everyone.

Ask THREE: (Letters to the Editor)



Dear Bob,

Congratulations on your very informative publication. I have owned a /// since the dark days of intermittent system failures (remember the drop fix?). Until now the medium for information exchange has just not been available. Your magazine will go a long way towards solving this problem.

On the subject of support, I have read many complaints about failure of Apple to properly support the ///. I must vehemently disagree. When a company completely replaces a product as Apple did for both the /// hardware and software, it shows total commitment. When was the last time you heard of an auto manufacturer completely replacing a defectively produced car?

From my experience, the major problem with the /// support is the local dealer that can not or will not properly service his customer. Many dealers can not intelligently discuss anything but the Apple //. If a problem existed with the /// or Apple software and the dealer could not help, I have always received a totally satisfactory response by contacting the Area Representative. Don't bother to call Apple, it's easy to get lost in the company's telephone maze.

I am primarily a *canned* software user, but if I can help on your efforts, feel free to call on me.

Sincerely,

Jim Edmondson
Nevada

Note: The above letter was written about a year ago, long before Apple's cancellation of the ///.

Dear Mr. Edmonson,

Now that Apple has shown their true colors with the cancellation of the Apple /// product line, your letter merits publication. If an auto maker sells defective cars that do not work, the federal government steps in and forces a recall to fix the problems. Apple did not face a mandatory recall, but if they did not replace the first 14,000 defective machines they could have faced a massive class action lawsuit. Their lawyers no doubt advised them that it would be cheaper to fix the defective computers.

Your thoughts on the question of support is very interesting. It should be noted that Apple also recently cancelled all of its agreements with local Area Rep. Firms. It is also very interesting to note that when a purchaser of an Apple /// system pays top-dollar list price you would expect some semblance of support from the dealer. When an Apple /// owner has to ask his local Area Rep. Firm for help, there's a serious problem.

When Apple introduced *ProDos* for the Apple //, they had an intensive dealer training program so that dealers would know what they were selling. Why wasn't this done for the ///? Apple never allocated the resources to do such a training program. Because of this, most dealers never understood what a powerful machine the Apple /// is—Thus, poor sales. In terms of corporate responsibility, Apple made the decision to cancel the Apple /// years ago when they refused to market the machine.

Dear Bob,

I'm trying to get an Apple /// user group started in the North Jersey area. Would you please publish a notice in your magazine asking for anyone interested in joining the group to write to me.

Your help is sincerely appreciated.

Sincerely,

Roger T. Richardson
P.O. Box 166
Irvington, New Jersey 07111

Dear Mr. Richardson,

Here's your letter, I hope you get a good number of responses. To anyone else who would like to start an Apple /// user group: *We will publish those notices!*

Dear Bob,

Somewhere in the manuals we received with our Apple /// I remember seeing a note to the effect that one cannot use SOS files in the CP/M mode, and that there is no way to convert SOS files for use in CP/M. It seems to us that in the opinion of the majority of the readers of ***ON THREE***, one of the worst things a person could do to an Apple /// is put a *Softcard* in it. However, we purchased one along with dBASE so we could get out from under the mess Quick File got us into. At this point I could spend quite a bit of time venting my frustrations about Quick File but I want this letter to be less than ten pages. We soon ran into the problem that most of our current data was lost in Quick File files and we had the onerous task of re-inputing all that data or finding a way to translate it. We chose the latter and have been quite successful.

The following is a guide through the steps to make it happen.

First: Write an ASCII data file out of the Quick File using the report generator. This will make an Applewriter data file (SOS TEXT file).

Second: Run the WPL program on the ASCII data file to clean out all the autoletter @ = @'s and < = >'s and add a space at the beginning of each line of the file (each of which is a field all to itself).

Third: Convert the file using the SOSXFER utility on the CP/M system disk, thus making it a CP/M compatible ASCII character file.

Fourth: Run the dBASE command file on the converted file to assemble the data which is one field / line into the standard data format. This format will be readable under dBASE when using a command such as "APPEND FROM FILENAME.TXT SDF" (MERGE.CMD is the filename).

Let us know if you think this set of programs would be of interest to the rest of the readers. If so we would be glad to send copies on disk to help others out of the deep abyss one can be helped into by Quick File. Also please add my name to the list of **Hot Line** people because I know from first hand experience how hard it is to find anyone who has any knowledge of the CP/M side of the ///.

Sincerely,

Ken Strauss
Ohio

Dear Mr. Strauss,

That set of programs would be very welcome, and I'm sure a number of our readers would appreciate them. We have added you to our Hot Line list, and we thank you for your consideration.

Dear Bob,

I read with great interest David Cortopassi's article in the latest issue. I then picked up the phone and ordered CROSSWORD-SCRAMBLER and some other items.

I would like to address a few of Mr. Cortopassi's statements. Mr. Cortopassi is obviously not a typical computer owner. He admits to being a mainframe programmer by profession, and an employee of Apple Computer. I am sure he is absolutely sincere when he describes the Apple /// as *user friendly* (how I hate that phrase), and the documentation *superb*. Given the above attributes, why did he go to such lengths to make CROSSWORD-SCRAMBLER a turnkey program? Might not sophisticates such as Apple /// owners consider this somewhat patronizing?

Let's view the Apple /// from the other end of the spectrum. I am probably not a typical Apple /// owner either. I have owned my Apple /// for about two years now. At the time of purchase I was 55 years old. I never went to college, and have nothing much in the way of technical background except in my own very limited field. I had no use in mind for the computer at the time of purchase. I bought it out of curiosity, I guess. I wanted to learn something about computers and thought that the best tool to use to learn about computers would be a computer. So far I have found my Apple /// to be about as friendly as a pet rattlesnake and the documentation to be nearly incomprehensible. So how would I learn? From the dealer? They were reasonably helpful at first but I soon sensed that they were beginning to consider me a nuisance. From Apple? I wrote three letters directly to Mr. Markula (**Editor's Note**: Mike Markula was Apple's last CEO), I even registered the third one to make sure it got there. The only response I ever received was the official Apple Computer Bug Letter from Chris Ryan, thanking me for my letter and saying that someone would be getting in touch with me shortly. That letter was dated 5-23-83. I'm still waiting.

Getting back to that documentation, I will give just one example that I feel is typical. On page 23 of the Business Basic manual is the command word **,chain**. I fiddled around with this on quite a few occasions and couldn't make it work. The statement as shown in the book is incorrect. In retrospect, I can't imagine why I failed to find the correct format by sheer trial and error because I used every combination of keystrokes I could think of. Where did I finally find my answer? In the B. Dalton book store, that's where. Ever since buying my computer I have been haunting that place on at least a twice weekly basis looking for a book dedicated to the Apple ///. As I am sure you know, there were none. Every other computer on the market, yes, but not the Apple ///. Finally on the ninth of March, 1984 (Oh glorious day) while scanning the shelves, my sharp eye caught the title ***BASIC KEYWORDS FOR THE APPLE ///*** by Eddie Adamis. I quickly grabbed it, there was only one copy, and ran for the cash register. I didn't want to let go of it, but the clerk insisted on putting it in a bag and stapling it shut. Leaping into my rusty Concorde I sped home. I ran into the house, ripping the bag open as I ran. I opened the book and found CHAIN. I fired up the hardware and tried the examples shown. They worked on the first try. I stuck my Basic practice disc into .D2 and started chaining programs at random. They all worked. I felt like Einstein discovering that $e = mc^2$.

Now really, I may not be as bright as the average Apple /// owner, but can you blame me for asking, if Eddie Adamis could make it so plain that I was able to make it work on the first try—why couldn't Apple Computer? If this were the only example I would be ashamed to complain at such length, but in my experience the manuals are full of such examples.

Eddie Adamis, by the way, has written another book on the Apple /// which has not yet been released by the publisher, ***Business Basic for the Apple ///***. The manager at B. Dalton, who by now is an old friend of mine, has ordered it for me and has promised to call me the moment it comes in.

Sincerely,

Dave Nelson
Wisconsin

Dear Mr. Nelson,

It's very interesting to see different peoples reactions to the same product. Mr. Cortopassi and you obviously do not see eye to eye on the subject of the Apple ///. While one who has been around computers for a while might find the Apple /// an *Easy* system, a novice to computing will undoubtedly have a tougher time learning the machine.

In reference to your attempts at getting help directly from Apple Computer, Inc: It hardly ever works. They just don't have a staff trained on Apple ///'s to answer many questions for end-users. If you are persistent you may be able to coax a better response, but don't hold your breath.

All manuals have mistakes in them, and unfortunately the Apple /// manuals have their share. Thanks for the notes on the pubs. from Edie Adamis, I'm sure many of our readers will find that interesting.

Dear Bob,

I have a ///, a Profile and a Qume, and enjoy your publication. Other than the motherboard blowing out the day after the warranty expired (it took three boards to get a good one), and some voltage fluctuation which interfered with the counting in PFS Report, thereby rendering an entire file useless, I haven't had many problems. The ones I have had might be of interest:

1. Using PFS File, sometimes a form is inaccessible when I try to retrieve it using column 1 (the 1st item on the form). However, I can get to it using any of the other headings, but obviously, this takes longer. The same problem doesn't affect printouts via PFS Report.

2. When printing, I have to use ".Sprinter" as a print option on PFS File, and ".Printer" for PFS Report. Not only that, on PFS

File, when I X the items I want printed, resulting in a single space printout with a common left margin, the first three letters of the first entry are always missing.

3. Using PFS Report, if your columns are over 40 characters wide, more often than not, letters in excess of 40 will wrap around underneath, making for a two line entry; you don't get a "too wide to print" warning. Strange...

I also use Quark's Word Juggler, but haven't been able to come up with anything quirky (or even Quarky). Their 2.6 version is the best yet, and does a better job of highlighting variables, underlining, etc.

Keep up the good work.

Sincerely,

Peter V. Taylor
California

Dear Mr. Taylor,

Regarding the PFS problems, it sounds to me like you have damaged PFS files. This usually occurs when a directory has been trashed or the diskette that you were using has been damaged in some other way. This would account for some of the problems which you experienced. Otherwise, it is possible that you have an older version of PFS for the Apple ///. In that case, call Software Publishing Corporation and make sure that you have the most recent version.

You can check the version number of the PFS programs by typing V at the main menu and then pressing ENTER. The version number will show up at the bottom line of the screen. The people at Software Publishing will undoubtedly want to know this number when you call them.

Thanks again for your letter and your comments.

Dear Mr. Consorti,

I just received Volume 2 Number 1 of ***ON THREE***. Even though I am just a user I really enjoy reading the articles and learning how others have solved their problems. I have just expanded my 128K machine to 256K so I could run /// E-Z Pieces. April is my first anniversary with my Apple ///. I learned Apple Writer, QuickFile and Visicalc. The cut and paste ability of /// E-Z Pieces is just what I have been wanting. Why does the Database portion store many decimal places from a DIF file from the spreadsheet which is formatted in 2 decimal place dollar? When I print out a database label, the left side has 12 on each line. This is very annoying at times. How can I delete this? I have an Apple Dot Matrix printer.

Over the past six months I have learned how unsupported the Apple /// really is. My goal shortly after I purchased this computer was to use the spreadsheet and database program to produce a small payroll for my anesthesia group of nine people. I didn't feel I should spend hundreds of dollars for this specialized service for so few entries. Until /// E-Z Pieces, I could not transfer spreadsheet data to and from database data. I asked around and read articles for a few months, then I started my fiasco. I decided the Personal Pearl would fit my needs. The listing stated that it would work on my Apple ///; but the company has not adapted the program for my machine yet. I could buy the CP/M version though. I decided I might learn about CP/M so I bought both. Next I wanted to calculate data within the Pearl. The program was designed to interface with Supercalc. Over Christmas when I was learning this program, Supercalc came out with a version that worked with graphics. Sounded great! I ordered it; but the company didn't have the graphics package for Apples yet. I decided to just get the spreadsheet. The first time the company sent the IBM version. That wouldn't run. The second time the software was for the Apple //. Again, Apple /// does not run Apple // emulation in CP/M. Next the company sent an 8 inch floppy disk with the Supercalc program for all the systems they had designed. I couldn't read it with my 5 & 1/4 inch disk drive. Next a downloading company put that information on a couple of appropriate sized disks. This time I went through all the monitor setups but none worked for the Apple ///. Next, with the continued help of my local dealer (who has been a jewel), I received a verbal commitment from Sorcium to send the necessary codes to my dealer who could get the necessary codes from Apple so the proper interface program could be added to the Supercalc program which still sits at my dealer's store. It's been one month with several nonproductive calls and an unanswered letter to Sorcium. My patience has run out at this point. Unfortunately I paid for the program months ago when I picked it up thinking it would run. When I asked for my money back since I had a nonfunctioning diskette, I was told to forget it. I can understand my dealer's situation with time and resources invested; however if I have a malfunction in my anesthesia practice even with good intentions I still get to experience a nasty malpractice suit most likely. All I want is to use a program that the company said would work on my Apple ///.

Another question relates to the Personal Pearl. The sign on version states it is set up for 52K. When I use up 52K, I can't store anymore. Obviously, I have more room in my 256K machine. Is there a simple solution? I have a somewhat large fill in sheet which fills the system so I can't enter any data. The manuals says I can break this down into several sheets but the interconnection lines are limited so I don't end up with all that data I want in my summary area. Has anyone else worked with this small elegant database program?

As you can tell I am limited in my computer knowledge and skills but really enjoy learning and using this means of data storage and manipulation. My need for typing services has markedly diminished since my access to word processing.

Keep up the great work. Several of my colleagues with Apple ///'s have wished they had invested in Apple ///'s after working with their systems for a few years. Unfortunately none have purchased this great unit yet.

Sincerely,

Milamari Cunningham M.D.
Missouri

Dear Dr. Cunningham,

With a little luck, someone will be able to answer your question on /// E-Z Pieces. I should have brought back our old "Horror Stories" column for your letter. It seems that everything that could go wrong has. Your dealer is to be commended for his lousy attitude regarding giving you your money back. I'd like to know his name so that other people in your area will remember the service you were given.

Regarding your question on the 52K limitation, your Apple 256K /// does NOT have any more room. When you are working in CP/M, you are limited to 64K of total memory. Some of this space is taken up by the display screen and other things, thus you have

only 52K of memory space for the program. Newer versions of CP/M allow for more than 64K of memory, but none have appeared that take advantage of the Apple ///'s large memory size.

Dear Bob,

Thank you very much for your letter and two issues of ***ON THREE***. I was delighted to read the letters from all the frustrated Apple /// users. In New Zealand there are very few Apple /// owners and the dealer I bought the machine is not interested to let the owners share their own experience. I think the few that have got the machine are thinking why the Apple /// owners are left in the dark by Apple Computer and its authorized dealers in **New Zealand**. From reading your articles and correspondence I can now understand the reasons why the Apple /// has not been popular and successful as the Apple //.

I was trying to trade in my /// for an Apple //e as the more I inquired about the /// the more I got frustrated till I read your encouraging magazines. I've enclosed my subscription for one year and hopefully I will continue using the /// with the help I get from your magazine.

I would like to make the following points for you to make some comments.

1. Why doesn't Apple Computer ensure that all their dealers have a code of practice to listen and help users with their applications and let each user contact each other for exchanging and sharing ideas.

2. I understand from the dealer that the /// is classed as a 2nd class machine because Apple themselves realize that the machine is not going to be successful and not going to be used in the wide range of applications like the //. I understand that the /// probably is going to be left as it is now and the //e will still give Apple Computer the stay in the market. This was one of the reasons I was considering to trade-in the /// for a //e.

3. It was interesting to know that there are over 100,000 Apple ///'s sold and still without any communication among owner/users. I hope you succeed with your brave attempt to reach all the users and owners.

4. Have you any knowledge of the new *ProDos* for the /// and //. If you do know, do you have any information about it and whether it is worth buying to get the use of the hundreds of programs available for the //.

5. I would like to know if it is possible to have the keyboard rearranged in a more practical way i.e., having **+** on the numeric keypad instead of the **-** key. And if there is a better layout of the keyboard it will be of advantage to everyone. I've seen some good keyboards advertised for the Apple // in *Byte* magazine. Maybe something similar for the ///?

6. I would like to order the products you have advertised but the Customs here tax 40% on the total cost which could cost three times the price from you. I'm also interested in the Gameport and for the price it is a very good buy but I will still have the same problem with sales tax and duty which ends up to about two to three times the original price. I want to know if you accept payment by credit cards such as Visa or American Express cards and if you charge extra for these services.

7. Do you know of any software house that sells Apple /// software and accessories? I want to know if there is a Z-80 card available from other manufacturers apart from Microsoft (which sells here for NZ $1000).

Looking forward to the next issue of ***ON THREE*** and awaiting your reply. I will be contacting other Apple /// users about you magazine when I find out some address of other users/owners.

Your sincerely,

Denis Win Thein
New Zealand

Dear Mr. Win Thein,

Thank you for your comments, we will continue to do our best to help you in all of your Apple /// needs. It is a shame that Apple doesn't take a stronger stand on educating it's dealers to the power of the Apple ///.

***ProDos* is a new disk operating system for the Apple // series of computers that uses the Apple ///'s disk format. Thus, you can use the same storage disk on an Apple // or ///. ProDos does NOT work on an Apple /// as it requires a 64K Apple // to work.**

You will find it very hard to rearrange the keys on your numeric keyboard as these key-codes are hard-wired into the .CONSOLE driver. You could change them, but it would require a modified .CONSOLE driver. Perhaps one of our readers will send in an article with the patches necessary.

We do accept Visa, MasterCard and American Express with no additional charges for the service. It is a shame about the customs problems, hopefully we can work out something. We would be hard pressed to give recommendations on a specific mail-order house as I don't know of any with Apple /// products. Possibly one of our readers can help you on this question also.

Dear Mr. Consorti,

Hurray for an Apple /// publication. I feel a lot less like the illegitimate stepchild! I buy magazines regularly and will soon subscribe. Let me put your readers to works.

A couple of questions regarding the Apple /// that perhaps you can answer or can guide me to answers. We run an Apple /// driving an Epson MX-100 printer. Presently we are using Applewriter /// for word processing.

1. I need to justify both right and left margins (fill justification) "and" do proportional spacing. We are looking for a word processor that will run on the Apple /// that will do both. Whether we interface to a Royal 5030 electronic typewriter or buy a daisy wheel printer capable of proportional spacing such as the Brother remains to be seen. Any suggestions?

2. We have experimented with the Super Cord interface for the Royal and can make it do almost anything we need except use the Royal's proportional spacing plus fill justification capability, and we have been unable to make the Royal accept several of the format commands such as bold and underlining. Problem, I suspect, is we don't know what commands the Royal reads. Again, have you any thoughts or directions?

In short, we are trying to produce fill justification and proportional spacing and so far have run amuck.

3. Whatever word processor we move to must also show the page on the screen prior to printing. I would like to take care of widows and orphans, but the page depiction on the screen would at least give us a chance to do this quickly prior to printing.

A by-the-way item: If the Applewriter /// book describes how to get a picture of the page on the screen, I have never found

it. **CONTROL P ?** gives the format menu. **PD.CONSOLE** will cause the program to print to the screen as formatted, though page breaks are simply shown by spaces. **CONTROL 7** will stop and start scrolling. Perhaps I stayed dumber longer than most; perhaps a bunch of Apple /// & Applewriter /// users have yet to find that out!

4. All kinds of rumors out re Apple /// enhancements coming. Can you substantiate any? These enhancements will determine whether we go with ProFile now or later and buy a Modem now or wait to see if Apple is going to put in on the inside.

5. Do you know of any 8 & 1/2 inch floppy drive that will daisy chain into the Apple /// disk drive system?

Again, great magazine—keep it that way and I'll keep on buying it.

Sincerely,

Holli Nelson
North Carolina

Dear Miss Nelson,

Since we don't have a Royal to test out, one of our readers will have to answer your questions. Since you want to do proportional spacing, I would suggest using the Word Juggler word processor instead of Applewriter ///.

Regarding Apple /// enhancements, there will be none. As soon as Apple cancelled the Apple ///, all hardware projects were also cancelled. However, as you can see from this issue *ON THREE* will shortly be announcing a few new projects for the Apple ///. Our main one will be a memory upgrade from 256K to 512K. Using state of the art memory chips, this replacement memory board will work with all existing Apple /// programs.

Dear Mr. Consorti,

In the process of trying to put together the package required to use **Draw ON ///** I contacted our Apple hardware vendor who informed me that they are no longer able to get the Cursor ///. He advised me that with the introduction of the Mouse for the Apple ///, the Cursor /// had been dropped. On that basis, will Draw ON /// make use of the new mouse for the Apple ///?

I expect that once we get all this information I will be very interested in acquiring **Draw ON** as soon as I configure it to my system and available peripherals.

Thank you very much for your response and help. Keep up the good work. I do intend to keep reading ***ON THREE***.

Sincerely,

James J. Bibby
Indiana

Dear Mr. Bibby,

The Cursor /// joystick is no longer being manufactured or sold by Apple. The reason behind this was not the release of the mouse for the Apple ///. Since Apple cancelled the ///, they will not be coming out with the mouse for the Apple ///. However, that did not deter us. We have added to Draw ON the capability to use the Apple // mouse.

If you buy an Apple // mouse, you can attach it to your Apple /// with the instructions that we provide with Draw ON. Likewise, we provide the software driver that allows Draw ON to work with the mouse. Thus, you will be able to have full mouse use within Draw ON.

Thank you for your comments, they were very much appreciated.

Dear Bob,

I've used my Apple /// for about two years now, which makes me an old timer. I've still got lots of questions, and I'm still discovering new capabilities. I'd be interested in seeing articles on modems and the required software. I'd also be willing to write such a piece, if arrangements for equipment and software loans-for-review could be made. Perhaps most users of the /// are like me—I know some programming, but do not consider myself a whiz, and because I hold a full time job (in my case, university professor), I can't spend unlimited amounts of time simply exploring. Documentation and information about specifics is often sadly deficient, even today.

How about some articles on using the /// in emulation mode? How do you get controlled printer output? How about accessing the additional memory of the ///? The standard 128K configuration in emulation is a 48K Apple //, so an expanded /// like mine, with 256K on board should have at minimum another 128K, logically. I've never seen anything discussing how that additional memory might be accessed, a key issue for running Apple Logo, for instance, or any other // programs requiring more than 48K.

How about a review of Applewriter ///? I've been reading computer magazines for a couple of years now, and Applewriter /// seems to be a big secret. Reviews typically cover Applewriter //, and assume that the version for the /// is pretty much the same thing (a massive understatement, at best). Such a review wouldn't influence me to buy, since I already own Applewriter ///, but a good tutorial on the system's advanced features and the techniques others have developed for getting maximum performance from the system would be interesting. I've written two books and about 10 articles using Applewriter ///, and I'm very pleased with it. A few complaints, however: the image on the screen doesn't show what the printed page will look like. Out of [Z] mode, this discrepancy means that what appears to be lined up on the screen (in columns, say) will not necessarily print out that way. My Applewriter /// notes SOS 1.1 incidentally.

I look forward to seeing ***ON THREE***—and to writing an article or two for you if interested.

Sincerely

Mariann Jelinek, Ph.D.
New York

Dear Dr. Jelinek,

Thanks for your letter, regarding accessing the additional memory of the Apple /// in Apple // emulation mode—it can't be done. Many of the programs that you would like to run in emulation mode have versions that run in Apple /// native mode. We did have a review of Applewriter /// in the Volume 1, Number 2 issue of *ON THREE*.

Please update your SOS from 1.1 to 1.3 if you already haven't. Simply transfer the file SOS.KERNEL

from a disk that displays the SOS 1.3 message to your Applewriter disk. This is all you will need to do. I look forward to any article contributions you may have.

Dear Mr. Consorti:

Lazarus arrived, and although it could not resurrect the file I had been hoping to bring back from the dead, I am very impressed with it. It is nicely designed, and I consider it good insurance.

I do not know enough about the technical aspects of file management to know whether it is possible to restore parts of files, but it would be nice if future versions of Lazarus could restore blocks of information even if the initial part of the file had been written over. That was the case with the file I was trying to restore. Somehow, 78 blocks of information had been written over with 2 blocks of information, and, if I understand correctly, 76 blocks of good information are still waiting to be rescued. There ought to be a way to restore those 76 blocks.

Your Editor's Block (Volume 2, Number 1), touched on some important topics for the future of the ///. For the /// to be successful, it has to become to the // what the Lisa is to the Macintosh. I know that I would find my machines (yes, I have two ///'s) more useful if I could run //e software and produce //e compatible programs. Think how nice it would be to be able to produce a Business Basic or Pascal program on the /// and be able to run it on a //.

Programs like /// E-Z Pieces will probably help, though it is probably too late for software to drive hardware sales the way VisiCalc helped sell Apple //'s. One of the reasons that some people opt for the // or an IBM is simply that a wider variety of software is available for those machines. Perhaps Apple should consider selling the /// bundled with /// E-Z Pieces or Applewriter, Advanced VisiCalc, and QuickFile (or Keystroke).

I have heard rumors that Apple will soon release two new cards for the ///—one a megabyte of memory, the other a 68000 chip (which I assume will offer Unix, the C programming language, and—with any luck—Macintosh emulation). Those features would certainly help, as will the improved versions of the 6502, assuming Apple puts them in ///'s as well as in //'s.

I have not read anywhere that the /// is a dud—only that Apple has not supported it. I have read that the IBM PC is a dud (too much money for too little computer) and that the PC jr is a dud (the worst computer value on the market). The /// is a good, solid machine, and I believe would sell if Apple could convince dealers to stock and display it.

If people who were interested in the // had a chance to compare it with a ///, I think that many would be willing to spend the few extra dollars to have all the advantages of the ///, especially if the /// were able to do everything the //e can do plus—with the addition of a card—emulate the 32-bit technology of the Macintosh as well.

The Apple][hit the market at just the right time, and Apple was lucky with sales. The /// was the first machine that it had to work to market, and Apple didn't do such a wonderful job. They are doing better with the Macintosh and Lisa, and I suspect that they will continue to improve their marketing strategies. They are doing their best to learn how to provide customer support (IBM's strong suit), and I think that they recognize the need for both product continuity and product diversity.

While we are waiting for Apple to announce its plans for the ///, keep up the good work. I am looking forward to the day that ***ON THREE*** is as thick as Softalk.

Cordially,

Joel P. Bowman
Michigan

Dear Mr. Bowman,

I'm sorry that Lazarus couldn't restore the file you deleted, if you had not stored that other file over the original file, Lazarus would have been able to do it. When you write over one file with another, more often than not, the first thing it overwrites is the information within the file structure that indicates where the blocks of that file are on that disk. When this occurs, Lazarus cannot do a partial file restoration.

About my last Editor's Block: Now that Apple has canceled the ///, it's a moot point. While Apple understands the idea of upward mobility within their 32-bit product line (Mac, Lisa)—the idea of the /// being the //'s big brother was just too hard a concept for them. With the cancellation of the Apple ///, they are probably reserving room for the future "big brother" of the //e. Whether they call it the //x or not remains to be seen.

Since you wrote your letter Apple has seen fit to bundle /// E-Z Pieces with the ///. That move, along with other new items in the Apple /// product line spurred sales over the last 6 months to new heights. Even with the increased sales (over 2,000 a month) Apple chose to drop the Apple ///. The /// is a solid machine and your (and my) hopes of Apple training dealers to sell the /// will go unfulfilled. If Apple had any commitment to the /// they would have made an Apple ///e with a built in Apple //e emulation mode plus some extras.

When you say that Apple didn't do such a wonderful job marketing the ///, you are generous. They not only did a terrible job marketing the ///, in many instances they deliberately hurt the Apple /// by not releasing products that they had developed inside of Apple. Since I am not bound by any of the confidentiality agreements that others have signed, I can tell of some of the products that Apple developed and just let die.

Ever wonder how Rupert Lissner created both Apple Works for the // and /// E-Z Pieces for the Apple ///? It seems a bit strange that Apple is selling it for the // but haba systems is selling it for the ///. You see, Apple didn't want to have the product for the ///, only the //. When Apple refused to publish the version for the ///, they gave the rights to Rupert Lissner (the author of both versions) who sold it to haba systems. We almost didn't have /// E-Z Pieces!

Sometimes I wonder how many other products like /// E-Z Pieces were developed by Apple for the ///. With a little luck we may see some others. Who knows! Maybe *ON THREE* can get a couple of them. With Apple wanting to sell Apple //'s over Apple ///'s I can see their point: Who would buy an Apple //e if when

Continued on page 5.

Why Is Everyone So Excited About Draw ON?

by Bob Consorti

Since we have begun shipping this exciting graphics package we have been getting a number of comments from ***Draw ON ///*** users. The response has been almost universal—*Draw ON* is the most powerful and exciting graphics program for the Apple ///. With this article I will explain just what Draw ON can do for you and why everyone is so excited.

Draw ON /// Unveiled

Draw ON /// is a powerful and versatile graphics tool designed exclusively for the Apple /// and Apple /// plus computers. Draw ON /// transforms your Apple /// into a combination drafting table, easel and sketch pad. Draw ON works in all of the Apple ///'s color and black/white graphics modes and brings the power of *LisaDraw* and *MacPaint* to your ///.

Draw ON is operated from either a mouse, joystick or the keyboard. The commands are simple with numerous menus and help screens to aid you in use of the program. A keypad overlay is included and help is always available to make many of Draw ON ///'s options even easier to remember and use.

Because many people requested it, we included in Draw ON the ability to work with the **mouse**. If you purchase the ***Apple //e mouse*** and follow our instructions for installing it, you can use the mouse to operate Draw ON. The //e mouse will work fine in the Apple ///. If you check this issue's *Editor's Block* you will see that we have a couple of interesting products coming out that will use the mouse. If you were thinking about buying the mouse, this might help you make the decision.

Once again, since a picture is worth a thousand words, there are a number of pictures here of just what you can create with Draw ON. From complex circuit layouts to architectural floorplans, Draw ON will make your designing work easier. Creation of charts & graphs and preparation of slides & tables for presentations are a snap with Draw ON. Using the powerful features of Draw ON will turn you into a one-person graphic arts studio.

Business Charting

Most of the popular charting programs allow you to save the chart that you have created to disk. After creating the basic chart and saving it on disk, you can use Draw ON to load in that drawing and change it to suit your tastes. Move objects around, texture bar or pie charts, or add and move titles around the drawing screen. Since Draw ON has a variety of font styles and sizes available, it is the ideal tool for making those boring charts more readable and exciting.

Figure #1 is a chart created by the popular *PFS: Graph* program. Note that all the legends are written in an upper case typestyle. The PFS program does not allow you to change the style or size of the letters you label your charts with. Using Draw ON /// you can change the style and size of the text on your chart. Simply save the PFS generated chart to disk and make the appropriate changes with Draw ON. Figure #2 is the same chart after a few minutes of changes with Draw ON. This looks quite a bit better than the PFS chart and only took a short time to do. Which would you rather have in your presentation?

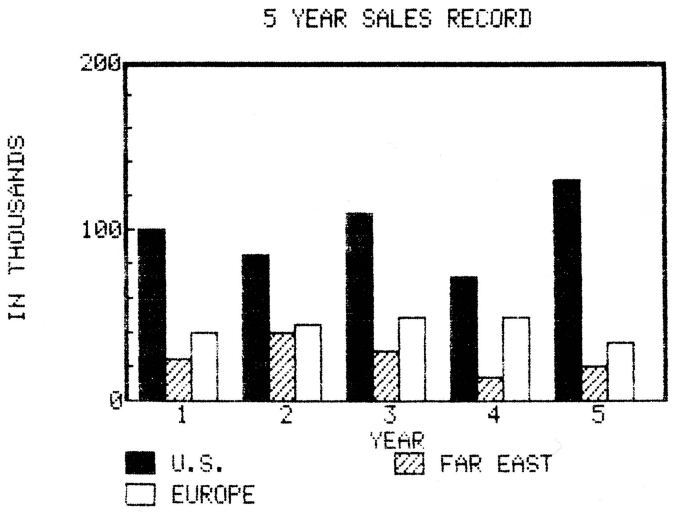


Figure #1

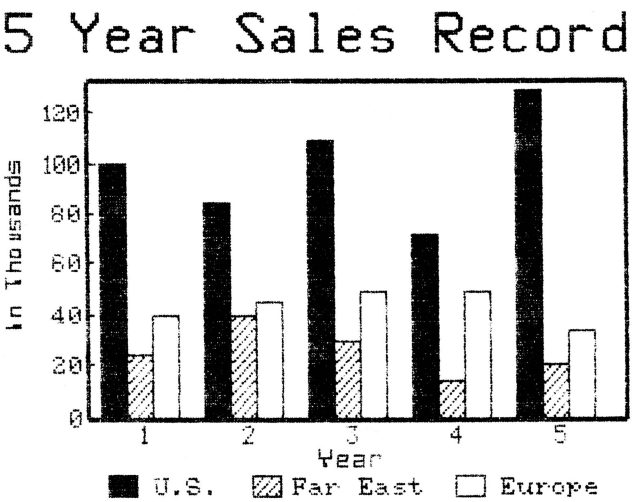


Figure #2

In addition to changing the style and size of the text on the chart, I used Draw ON to re-size the dimensions of the chart. If you will look at the chart that PFS: Graph drew, there is a good amount of the chart that isn't full. PFS: Graph incorrectly sized the chart when the program initially drew it. Since it looked a bit *off*, I expanded that portion of the chart and fixed the number legends on the left side of the chart. In just a few minutes, I made a very presentable chart out of a dull one.

You can use Draw ON and its multiple fonts and texturing capabilities to design your own business cards, forms, title pages and company logos. You might create special symbols, letterheads, calendars, etc. for your small business. Print the prototype drawings and bring them to your local printing company as examples of what you want. You can also use a copier to make your own stationery, forms or advertising flyers.

Draw ON is also an excellent tool for the graphic artist. They can use the program to speed the initial design and evaluation

of advertising materials, labels, color combinations, product wrappings or box design. Draw ON offers many of the tools of a drafting table: Pens, erasers, templates (circles, arcs, triangles, rulers etc.), scissors and tapes are all provided within Draw ON. Just as a graphic artist uses his tools to aid in finishing his artwork, you can use the tools of Draw ON to help you complete your graphic projects.

Cut & Paste

One of the most powerful features of Draw ON is the ability to *pick-up* and make copies of objects. This cut and paste facility allows you to create unlimited libraries of your own special figures and objects (such as circuit components, business logos, animation characters...). You may select these objects and move them onto, between, and around any of the available drawing screens.

If you look at Figure #3 you will see some of the uses of the cut and paste facility. After drawing the original box I made a copy of it reversed along the Y-axis. Then I drew some circles and placed some boxes in front of the circles and some behind. In just a few minutes I used Draw ON to create a set of complex figures that could not be drawn on an Apple /// before Draw ON ///.

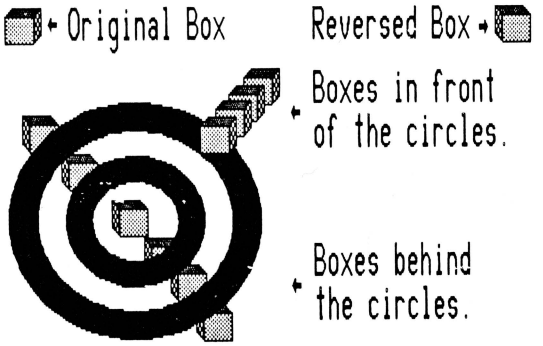


Figure #3

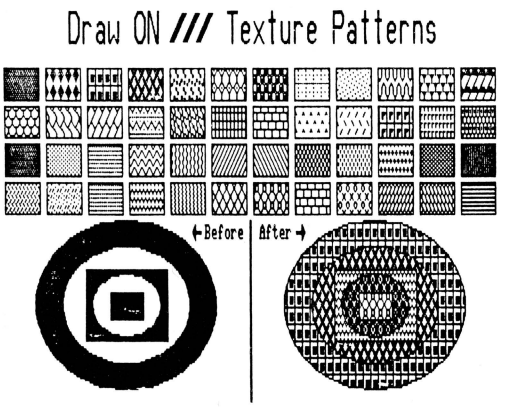


Figure #4

Textures & Patterns

You might be wondering about the patterns inside the boxes. Draw ON has the ability to fill an object (or an area of your drawing) with a specific pattern. Some of these patterns are shown in Figure #4. You can change them at any time to suit your tastes. Up to 24 may be used at any given time and you can load and save the textures to disk for later use. Figure #4 also shows how a complex object can be filled with a number of patterns.

In addition to changing the texture patterns, Draw ON allows you to create and modify the fonts that you are using. If you would like to create a script style font—you can! Draw ON makes it simple to design characters in any style or size. You can label your drawings with any of these fonts and even use them in your other programs.

Figure #5 shows one of the pictures that Draw ON created. If it looks familiar—it should! Apple has a similar picture for the MacIntosh that is in almost every ad for that machine. Before Draw ON ///, I don't think anyone believed that such a picture could be created on an Apple ///. Please don't let the picture confuse you. While we offer everything that the graphics program for the MacIntosh offers (and work in color!), we don't follow the same user interface.

Figure #5

Shape Records

One of the features that Draw ON offers is the ability to draw an object as a ***Shape***. After drawing an object as a shape, you may then scale the size of the shape or rotate the object with much greater ease, speed and precision. Shape records also require only four blocks of disk space (versus up to 33 for a picture), so objects may be stored more efficiently as shape records than as fotofiles.

After selecting *Shape Recording*, simply draw the object you want to record as a shape using any of the drawing tools at your disposal. As you draw lines, arcs and other pieces of your object, Draw ON records each movement and drawing. When you redraw the object, Draw ON simply repeats each of your movements and drawings at high speed. You can scale the size of the object up or down and rotate it to any angle. Since all of this happens on the vector (line) level it is very fast—much faster than scaling or rotating a standard object.

You can use shape records to initially describe an object, such as an electrical component. Rotate it into all of the orientations you are likely to need, and then create and save a library screen of that and other electrical components. From then on, use the ***preserve/preview/restore*** techniques to put the object on your drawing screen.

Shape records give the Draw ON user capabilities approaching those of the popular *LisaDraw* and *MacDraw* programs. While most graphics programs work on either the bit-image idea (MacPaint) or the vector (lines and curves, etc.) idea (MacDraw), Draw ON /// uses ***both!***

Printers Supported

After creating your drawing you will probably want to make a printed copy of it. Draw ON allows you to print out your drawings on a variety of dot matrix printers which are connected to your Apple /// via the *PKASO Parallel Interface Card*. Additionally the PKASO card can be used from within all of your other programs. Through the PKASO card, Draw ON supports the following printers:

- Epson MX-80, Type-II, Old MX-100
- Epson Type-III, Graftrax, Graftrax-Plus
- Epson MX-70
- Centronics 739
- Okidata, text models
- Okidata with Okigraph
- IDS Black & White Prism
- IDS Color Prism
- NEC PC8023
- C.Itoh 8510
- Apple Dot Matrix Printer

By the time this article gets to you we will have adapted Draw ON to work with a number of other printers. If you already have an Apple Dot Matrix Printer and a UPIC (or any other parallel card), we have a version of Draw ON for you! We will also soon be shipping the Apple Imagewriter version. If you have this printer and have connected it to your Apple /// via the RS-232 port on the back of the machine, you will be able to use Draw ON directly. Even if you have your Imagewriter connected via a serial card, Draw ON will work correctly.

We are also working on a version that will be compatible with the EPSON series of printers. If you have an Apple DMP or Imagewriter, or even an EPSON please call or write ***ON THREE*** for information on availability of a version of Draw ON that works with your printer.

Draw ON /// Contest

In each coming issue we will have a contest for the best picture, font or shape created with Draw ON. The first prize will be $100 and the runner up will receive $50. In addition to the $100 cash prize, the winning entry will be featured on the cover of the next issue of ***ON THREE***. So if you have created an exceptional picture, font or shape using Draw ON, submit it to us and get in on the fun. To enter, send in a diskette with your entry to:

ON THREE
Attn: Draw ON Contest
P.O. Box 3825
Ventura, CA 93003

Please send only standard 140K diskettes and sufficient return postage if you would like your diskette returned. You may also put as many entries on a diskette as you wish. Make sure that you include your name and return address. All entries will become the property of ***ON THREE*** and may be included in future issues or DOM's as examples of just what Draw ON can do. The decision of the judges will be final.

Continued from page 27.

If it sounds like I like this program, then you're getting the message. Because it is the first type of **Idea Processor** that handles a specific situation, I foresee a new rush in the computer industry to create this type of program. Channelling the power of your personal computer into helping you do specific areas of your job will be the most important part of the computer industry in the next few years. And I'm not talking about word processing, spreadsheets, or filing systems - This is a completely different aspect to personal business computing.

The program is fairly bug free. I haven't been able to make it crash yet! The only possible shortcoming is the length of time the program goes away to **think**. At certain parts of the program it appears that the program is **hung-up** for 30 seconds or more. A message indicating that the program was thinking would be nice. This usually occurs when you have a lot of issues and objectives that the program is working on. Aside from this small fault, I've found no major problems with the program.

Every mid-level manager on up should have a copy of The Art Of Negotiating Computer Preparation Program. No, let me widen the audience a bit, anyone who does anything resembling negotiating should have a copy of this program. In the past few months I've used the program twice to help me in a negotiation. What were the results? Well, I can't talk about specifics, but let me say that the results were worth much more than the list price of the program.

The Art Of Negotiating Computer Preparation Program is priced at $495 and is available from Experience In Software, 2039 Shattuck Avenue, Suite 401, Berkeley, California 94704. (415) 644-0694.

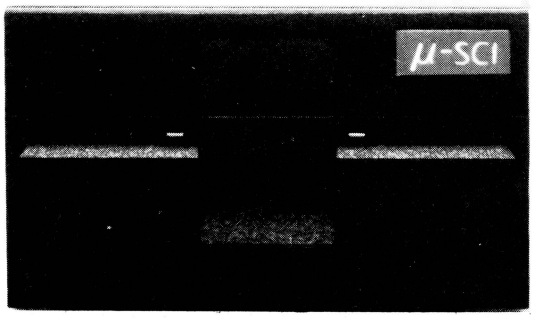
A Look At The Micro-Sci A3 & A143

by Bob Consorti

After using the *Micro-Sci* **A3** and **A143** for a few years, I would like to share my experiences with everyone considering purchasing one of these low cost disk drives. For those of you not familiar with the *Micro-Sci* line of drives for the Apple ///, this article will explain the findings of our rigorous tests.

Micro-Sci A3

The **A3** disk drive is a direct replacement for the Apple Disk /// and stores 140K of information. Just as with all of the other *Micro-Sci* drives for the Apple ///, the **A3** plugs directly into the back of the /// or it can daisy chain from the back of your Disk ///. The drive doesn't need a power cord as it draws all the power it needs from your Apple ///.



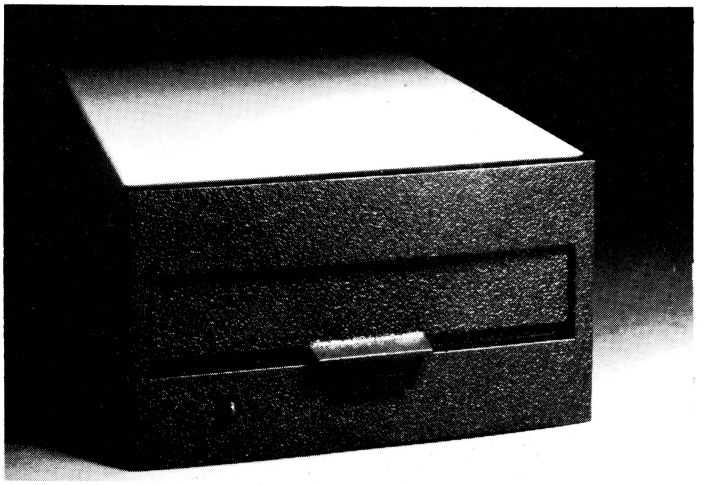
Up to three extra **A3** drives can be daisy chained onto your Apple ///. The **A3** works at the exact same speed as the Apple Disk ///, thus it takes the same amount of time to store information on disk. Since it is an exact replacement for the Disk ///, the **A3** uses the same diskettes as the Disk ///. All of your data that is currently on Disk /// diskettes can be used directly on your **A3**.

Programs that are set-up to store information on a standard Disk /// will find no problems with the **A3**. Because it looks to the program as if it were an Apple Disk ///, there are no compatibility problems! All programs that use a Disk /// will work with the *Micro-Sci* **A3**. The **A3** even works in Apple][emulation mode, looking like another 140K disk drive.

At our office, we have used the **A3** successfully for a couple of years now. Every machine we have has at least one **A3** hooked up to it. Two of our machines are dedicated to disk copying and have three **A3** disk drives attached to them. After thousands and thousands of copied disks we have had no problems with our **A3** drives. *Micro-Sci* has been making disk drives for Apple Computers for a long, long time and they are among the most reliable in the industry. The special ***ON THREE*** discount price of only $299 will save you $80 from the cost of an Apple Disk ///.

Micro-Sci A143

The **A143** is an Apple /// compatible disk drive that stores four times the information of a standard 140K drive. That's 560K, or **573,440 bytes** of information on a single diskette. Up to three **A143** drives can be attached to your Apple ///, giving you up to one and three quarter megabytes of data on-line at all times!



This drive can be used with all popular Apple /// programs. Backup ///, PFS, Visicalc, Quick File, /// E-Z Pieces, Applewriter, Word Juggler and all the others will work with the **A143** drive. A standard Apple /// diskette holds 280 blocks of information and the **A143** diskettes hold 1120 blocks. You will be able to store over half a megabyte of information on the **A143** diskettes. If you have ever run out of room on a regular floppy diskette when trying to store a large Visicalc or /// E-Z Pieces file, the **A143** will meet your needs-without costing you an arm and a leg.

The **A143** plugs directly into the back of the Apple /// and does not require an interface card or power cord. You can attach up to three **A143**, **A3** or Disk /// drives to your Apple ///, in any combination. Thus, you could have one external Disk /// and two **A143** drives, two **A143**'s and one **A3**, etc. With the addition of the software driver, the **A143** is compatible with all Apple /// programs.

When you add a device such as a ProFile hard disk to your Apple ///, you have to add the appropriate device driver to your boot disks so that you can use the device with a particular program. When you add a *Micro-Sci* **A143** disk drive, you will do the same thing. Simply follow the directions in the *A143 User's Manual* for using the System Configuration Program to add the *Micro-Sci* device drivers to your boot disks and you will be able to use the **A143** with all of your programs.

You can even change the name of the drive to your choosing. Normally, the *Micro-Sci* drives will be used with the names .D2, .D3 or .D4. Some programs require that you have a ProFile hard disk drive with the name .PROFILE. Subject to space limitations, you can fool those programs into thinking that you have a ProFile hard disk drive by naming your *Micro-Sci* drive .PROFILE!

The **A143** disk drives offer four times the capacity of a standard drive by using both sides of the disk (standard drives use only one side) and by packing twice as much information on each side. Because of this, high quality diskettes must be used. For the **A143** you will need diskettes rated for double sided, double density use. These diskettes will typically run about $4 or $5 apiece. Even though you are using the same sized (5 & 1/4 inch) disks,

once you have formatted an **A143** diskette, you will only be able to use that diskette in an **A143** drive. You will not be able to use an **A143** diskette in your standard 140K drive. If you want, however, you can reformat the disk in a 140K drive and use it in that drive, but you will not be able to read an **A143** disk in a standard drive.

While a standard drive cannot use a high capacity **A143** diskette, the **A143** can read a standard diskette. Thus, you can use the **A143** to read your current diskettes. While you can read from a standard diskette in the **A143** drive, it isn't able to write to a standard diskette. This means that if you only have an external **A143** drive, you can make copies of standard 140K diskettes by putting the diskette to be copied in the **A143** and the diskette to be copied onto in the internal drive.

While the **A143** works fine with all programs running under SOS, it does not work in emulation mode. This is due to the fact that while running under SOS, the **A143** is controlled by the *Micro-Sci* device driver—but while in emulation mode, there are no device drivers and no way to control the drive. If you need to have a second disk drive in emulation mode, buy the **A3**. This will give you the second drive that some programs require while in Apple][emulation mode. You can then attach the **A143** to the back of your **A3** and have three drives on-line at once while using Apple /// programs.

You can even use the *Selector ///* or *Catalyst* program selectors with the **A143**. These utilities will allow you to place your programs on the **A143** and switch between them without restarting your Apple /// each time. Thus, you will be able to go from Apple Writer to Visicalc to the System Utilities at the touch of a button. You don't have to buy a ProFile hard disk for these capabilities, you can do it with your *Micro-Sci* **A143**!

Backing-Up Your Hard Disk

If you have a ProFile hard disk you will no doubt curse every time you have to do a back-up. Using a standard Apple drive you are in for a good two or three hours of shuffling disks in and out of the drive. Make sure that you have enough diskettes because you will need up to 40. If you have a 10 megabyte drive you will need 80 diskettes, etc. By using the *Micro-Sci* **A143** drive you will only need a maximum of 9 or 10 disks and an hour or so to back-up your hard disk.

For those unfortunate few who have had their hard disk crash and who haven't done a back-up because they didn't have the time, the **A143** is what you need. Time after time we get letters from people who neglected to keep up to date back-ups of their hard disk and then had the disk fail. The **A143** will save you precious time in the event of a problem with your hard disk. Many people have to re-enter months of work when something like this happens. Since the **A143** provides an effective means of backing-up your hard disk, everyone with a hard disk should have one.

We have used the **A143** disk drives for a couple of years in our office without any problems. Our **A143**'s have acted as both main storage devices, and backup devices for our hard disks. In the beginning we used them only to store information from orders. As our business grew we added hard disks for our main storage devices and used the **A143** drives to back up the hard disks. To date we have not had one **A143** returned to us being bad. Just as with the **A3** drive, the **A143** has a great record of reliability.

With a price of only $498, the **A143** is a versatile and attractive option for everyone who owns an Apple ///.

Where Should You Buy?

When you think about purchasing an expensive item you should always weigh the benefits of buying it from your local dealer at a high price than from a mail-order house at a much lower price. The typical argument for buying it from your local dealer is that he will offer you service and support whereas the mail-order house (because of their low price) offers no support.

On one hand, ***ON THREE*** can be classified as a mail-order house because we do sell our products through the mail. Where we differ from the standard mail-order house is on the subject of service and support. In the rare case that your disk drive requires service, will your dealer simply give you a new drive while he sends the bad drive back for repair? If you ask your dealer if he will carry a spare drive in the event your drive goes bad, he will tell you in no uncertain terms NO.

Because ***ON THREE*** offers same day shipping, we have to keep a number of drives in stock. Thus, when someone calls in saying they have a bad drive, we can immediately ship a replacement. This saves your computer from being *down* and keeps your productivity high, even during those unexpected problems.

Even if your dealer provides good support, chances are that he has never seen or even heard of the *Micro-Sci* line of drives for the Apple ///. Since ***ON THREE*** has sold hundreds of these drives and has used them extensively in-house over the past few years, we are the only ones qualified to help you when you have questions or problems with the drives. If you aren't sure about how to connect your drive or any other question, call us—we can help you with all of your Apple /// needs. Since we offer **service**, **support** and very ***low prices***, we are confident that when if comes time to buy a disk drive for your Apple ///, you will come to ***ON THREE***.

/// to the Max #4 ...And Things That Go GLEEP In The Night

by Al Evans

This month's column concerns making sounds on the Apple ///. I have spent an embarrassing number of hours investigating the subject, and have grudgingly reached the conclusion that the simplest way is the best. Theoretically, the Apple /// has fairly powerful sound-generation capabilities. But for reasons which will be explained below, it is impractical to make full use of these capabilities. Instead, we have to rely on a very basic technique.

Stated simply, you make a sound on a computer by 1) issuing some instruction which causes a loudspeaker to move, 2) waiting some amount of time, and 3) repeating these two steps for however long the sound is to be made. If the amount of time you wait in step 2) is the same for each repetition, the sound is a more-or-less pure tone with a pitch which is an inverse function of this time—longer time equals lower pitch, shorter time equals higher pitch. If the amount of time between cycles varies randomly, the sound is a noise. If the time is changed regularly between cycles, the pitch of the tone or noise sweeps up or down.

There are three ways of accessing the loudspeaker built into the Apple ///. The first is a hardware beep generator generally used to signal the user that something is amiss—I'm sure you've all heard it. It exists because it is sometimes necessary to make a noise without any delay in processing. For example, if the computer receives an ASCII BEL character (**CONTROL G**) over a phone line at high speed and stops processing long enough to beep obediently, it may well miss the next character or two. With the hardware beep generator, the character causing the beep takes no longer to process than any other. This is a very useful noisemaker, but with an extremely limited repertoire—one pitch, one duration. The normal way of accessing it (from Pascal) is to write a CHR (7) to the console. For assembly-language programmers, the **pushbutton** for this tone is at location $C040 with the I/O space enabled.

The second way of gaining access to the loudspeaker is the one used in the **.AUDIO** driver provided by Apple. This device driver makes lovely pure square-wave tones over 7 1/2 octaves at 63 different volumes. Unfortunately, that's all it does. It's nice for playing little tunes, but doesn't provide for any sound effects, and has no **on/off** switch. Since I was writing a game and needed these facilities, I decided to explore more deeply.

Deep within the bowels of the computer are two devices called **6522 Versatile Interface Adapters** (or **VIA's** for short). The VIA is a marvelous chip combining many useful functions. These two devices are responsible for much of the power of your Apple ///. One of them, the **EVIA** is used among other things, to operate the loudspeaker. Six bits of one of its output registers are connected to a simple digital-to-analog (D/A) converter which controls the voltage on the speaker. In the **.AUDIO** driver (see the **Standard Device Drivers Manual**, pp. 129-134), the voltage on the speaker is set according to the VOLUME parameter. One of the two timers in the VIA is set to the COUNT parameter, while the other is set to a constant representing the duration of one unit of the TIME parameter. Each time the first timer runs down, the voltage on the speaker is reversed. Each time the second runs down, the TIME remaining is decremented.

And that is the **SIMPLE** explanation. All of this is done on the **device driver** level, and involves mystic acts such as allocating and deallocating undocumented **System Internal Resources**; setting and clearing the interrupt-inhibit flag cleverly but illegally to keep the system happy; and strange routines to read six characters either one at a time or all at once.

I analyzed it thoroughly because with this setup, it is possible to make sounds on an **interrupt-driven** basis; i.e., to have the computer go on about its business until the time has come to change the voltage on the speaker, then change the voltage, then go back to whatever it was doing before. Obviously, this is useful in a game—the action and sound can be completely independent of each other. Furthermore, only 1 or 2% of the time in a sound making routine is spent making sounds—the rest is spent just waiting until it's time to do something. This means that an interrupt-driven sound routine would be almost **free** from the standpoint of processor time. Encouraged, I wrote an interrupt-driven **.AUDIO** device driver. After five versions, I had to admit defeat. It worked, but...

Unfortunately, according to the **SOS Device Driver Writer's Guide**, the minimum response time to call an interrupt handler is about 160 microseconds, and another 115 microseconds is required to return from the interrupt handler to whatever was happening before the interrupt occurred. So even though my interrupt handler takes only 15 microseconds to execute, the total time required to update the speaker is 290 microseconds. Since the voltage on the speaker has to be changed twice per audio cycle, a sound with a frequency of about 1700 Hz consumes ALL of the computer's processing time—not exactly an improvement.

And that brings us to sound-producing technique three, one with which Apple][programmers will be familiar. The speaker itself is mapped into the computer's memory at location $C030. Any reference to this address (with I/O enabled by the environment register) will result in a tiny **click** from the speaker. A great variety of sounds can be made by controlling the frequency of the clicks in interesting ways. For the reasons above, this is the technique I use and recommend.

I have Apple][sound routines gathered from many sources over the years. I have adapted four of my favorites for the Apple /// Pascal environment; they are included with this column.

The Routines

The first of the four assembly-language routines in **.PROC NOISE** (**Program Listing -1**) produces a **white noise** modified by a **lowpass filter** which can be swept over a range of frequencies. The white-noise generating technique was originally published in the April, 1980 **Byte** by J. O'Flaherty. he **filter** and **sweep** were added by Ray McVay in a version published in the September, 1980 **Call-A.P.P.L.E.**

The second routine produces a tone with a frequency which can be swept up or down. It is from the same **Call-A.P.P.L.E.** article. The third routine produces a tone at a volume which can be varied over 16 steps. I got it from the June, 1981 **Apple Assembly Line**; it was originally written by Mark Kriegsman and was revised by Bob Sander-Cederlof. The fourth is from the same source. It demonstrates one method of producing two tones at once.

The noisemaking routines are declared as a Pascal procedure

Procedure Noise (N__TYPE: Noise__Type;
REPS, TIME, P__OR__F, D__V__P2: Integer);

where Noise__Type = (N,T,D,V) { Noise, Tone, Dual tone, Volume tone }
REPS = number of repetitions of this noise
P__OR__F = lowpass filter (for noise) or pitch (for tones)
D__V__P2 = sweep direction and rate (for types N and T), pitch 2 (for type D), or volume (for type V)

Two additional routines, **Sound__On** and **Sound__Off**, allow control over whether the noise is actually made or not. Particularly in games, sounds are interlaced with action, and it must take just as long NOT to make a sound as it does to make it, or the game will run at different speeds with the sound on and off. These procedures set or reset a **switch** which determines whether or not the main routine will actually enable access to the speaker. If the switch is **off**, the routine runs just as usual, taking the same amount of time to execute, but doesn't **click** the speaker.

The Pascal Program

With so many possible combinations, it would take incredible luck or determination to choose parameters for these routines by a trial and error method. Use the **NOISEFINDER** program (**Program Listing -2**) to **tune in** exactly the right numbers for your next explosion.

This program lets you use the arrow keys and a joystick plugged into port B on the back of your Apple /// to vary the parameters of the Noise procedure. It has two main modes of operation. In the **make noise** mode, the joystick is inactive, the noise is produced only when a key is pressed, and the **parameters** display is updated each time the noise is made. In the **repeat** mode, the joystick is active and the display is updated only when a key is pressed. The recommended approach is to choose the type of noise desired, wiggle the joystick around until something approaching the desired sound is obtained, then press **M** for **make noise**. This locks the sweep rate (delta), second pitch, or volume and lets you fine-tune the duration, first pitch or filter, and number of repetitions from the keyboard.

When the sound is exactly right, think up a name for it and press **P** to print the parameters out on your printer.

Putting it together

Enter the Pascal program **NOISEFINDER**, save it as **NOISE.PASC**, and C(ompile it to a codefile of the same name. Enter the assembly language PROC's, save them as **NOISE.AL**, and assemble them to a codefile of the same name. Invoke the system L(inker from the main command line. Enter **NOISE.PASC** as the name of the host file. Enter **NOISE.AL** as the library file name. Press <RETURN> when asked for the next library file name. <RETURN> when asked for a map file name, and enter **NOISE. LNK** as the name of the output file. If everything goes right, you should be able to Xecute NOISE.LNK.

And into the sunset...

Somehow, I'm intuitively certain that the technique discussed in this column is NOT the best way to make sounds on the Apple ///. If you're really interested in helping build a better **.AUDIO** device driver, send me a blank disk with sufficient return postage and I'll send you what I have discovered and developed so far. Perhaps by cooperating, we can build a REAL noise generator for the ///.

Program Listing #1

```
;*************************************************************************
;*                                                                       *
;* .PROC NOISE                                  | Copyright 1984 by |  *
;* -----------                                  |  O N   T H R E E  |  *
;* by Al Evans                                  |    Volume 2, #2   |  *
;*                                                                       *
;* These assembly language routines are used by the Pascal program       *
;* NOISEFINDER to produce sounds on your Apple ///.                      *
;*                                                                       *
;* NOISE: Pascal Declaration                                             *
;*                                                                       *
;* Procedure Noise(N_TYPE: Noise_Type; REPS, TIME, P_OR_F, D_V_P2: Integer);
;*   where Noise_Type = (N,T,D,V)) (Noise, Tone, Dual tone, Volume tone) *
;*         REPS = Number of repetitions                                  *
;*         P_OR_F = Filter (for noise) or Pitch (for tone)               *
;*         D_V_P2 = Delta (for N,T), Volume (for V), or Pitch 2 (for D)  *
;*                                                                       *
;*  These Apple /// routines are based on Apple ][ routines by:          *
;*                                                                       *
;*           J. O'Flaherty, R. McVay, and M. Kriegsman                   *
;*                                                                       *
;*************************************************************************

            .MACRO      POP
            PLA
            STA         %1
            PLA
            STA         %1+1
            .ENDM

            .MACRO      PUSH
            LDA         %1+1
            PHA
            LDA         %1
            PHA
            .ENDM

;-------------------------------------------------------------------------

            .PROC       NOISE,5

;-------------------------------------------------------------------------

            .REF        N_SWITCH

;-------------------------------------------------------------------------
;Zero-page variables
;-------------------------------------------------------------------------

RETURN      .EQU        0
DELTA       .EQU        2
VOLUME      .EQU        2
PITCH2      .EQU        2
P_OR_F      .EQU        3
TIMEL       .EQU        5
TIMEH       .EQU        6
OLD_ENV     .EQU        7
REPS        .EQU        8
PITCH       .EQU        09
SHFTREGL    .EQU        09
SHFTREGH    .EQU        0A
TIME_IN     .EQU        0B
FILTER      .EQU        0C

;-------------------------------------------------------------------------
;Hardware addresses
;-------------------------------------------------------------------------
```

```
SPEAKER         .EQU        0C030
ENVRMT          .EQU        0FFDF

;-------------------------------------------------------------------------

SETUP           POP         RETURN          ;Save return address
                PLA                         ;and get parameters
                STA         DELTA
                PLA                         ;Discard hi byte
                PLA
                STA         P_OR_F
                STA         PITCH
                STA         FILTER
                PLA
                PLA
                STA         TIME_IN
                STA         TIMEH
                PLA
                PLA
                STA         REPS
                PLA
                LDA         N_SWITCH        ;Is this noise for real?
                BEQ         GET_NTYPE
SET_IO          PHP                         ;If so, set up access to
                SEI                         ;speaker
                LDA         ENVRMT
                STA         OLD_ENV         ;Save previous environment byte
                ORA         #40             ;Turn bit 6 on for I/O
                STA         ENVRMT
                PLP
GET_NTYPE       PLA                         ;Get Noise_Type parameter
                TAY
                PLA                         ;Discard hi byte
                TYA
                BEQ         NOISE1
                CMP         #01
                BNE         $1
                JMP         TONE
        $1      CMP         #02
                BNE         $2
                JMP         DTONE
        $2      JMP         VTONE           ;Assume #03

;-------------------------------------------------------------------------
;Pseudo-whitenoise routine based on an article by Ray McVay, Call-A.P.P.L.E,
;September, 1980, which in turn was based on an article by J. O'Flaherty,
;Byte, April, 1980.
;-------------------------------------------------------------------------

NOISE1          SEC
                INC         TIMEH
                LDX         FILTER          ;Minimum delay = lowpass filter
LOOP1           ROL         SHFTREGL
                ROL         SHFTREGH
                TXA
                BEQ         PASS
                DEX
PASS            BNE         NOCLICK
                BCC         NOCLICK         ;Click if carry set
CLICK           LDA         SPEAKER         ;Click speaker
                LDX         FILTER          ;and reset "Filter"
NOCLICK         ROR         A               ;The ROR's, EOR, and ASL's
                ROR         A               ;have the effect of setting
                ROR         A               ;the carry flag at random
                EOR         SHFTREGH        ;times...
                ASL         A
                ASL         A
                ASL         A
SWEEP           PHP                         ;Save carry
                LDA         DELTA           ;Change filter value?
                BEQ         NODELTA
                DEY
                BNE         NODELTA
                TAY
                BMI         DOWNIT
UPIT            LDA         FILTER          ;UPIT and DOWNIT alter minimum
                BEQ         NODELTA         ;delay according to sign of
                DEC         FILTER          ;DELTA
                TYA
                BNE         NODELTA
DOWNIT          INC         FILTER
                AND         #7F
                TAY
NODELTA         PLP                         ;Restore carry
                DEC         TIMEL
                BNE         LOOP1
                DEC         TIMEH
                BNE         LOOP1           ;Not finished yet
                DEC         REPS            ;Do it again?
                BEQ         $1              ;No, exit
                LDA         TIME_IN         ;Yes, restore original
                STA         TIME_H          ;variables
                LDA         P_OR_F
                STA         FILTER
                JMP         NOISE1
        $1      JMP         FINISH

;-------------------------------------------------------------------------
;Tone with sweep routine based on the article by R. McVay mentioned above
;-------------------------------------------------------------------------

TONE            LDA         DELTA           ;Time before next pitch change
                AND         #7F
                TAY
                LDX         PITCH           ;Basic pitch delay
                LDA         SPEAKER         ;Click speaker first time
LOOP2           DEX
                BNE         NOTYET
TONE1           LDX         PITCH
                LDA         SPEAKER
NOTYET          LDA         DELTA           ;Sweep?
                BEQ         NODELTA1        ;No...
                DEY                         ;Sweep now?
                BNE         NODELTA1        ;No...
                TAY                         ;Yes, time to change pitch
                BMI         TDOWN           ;)= $7F, sweep down
TONEUP          DEC         PITCH           ;( $7F, sweep up
                TYA
                BNE         NODELTA1
TDOWN           INC         PITCH
                AND         #7F
                TAY
NODELTA1        DEC         TIMEL           ;Finised yet?
                BNE         LOOP2           ;No, continue
                DEC         TIMEH
                BPL         LOOP2           ;No, continue
                DEC         REPS            ;Do it again??
                BEQ         $1              ;No, exit
                LDA         TIME_IN         ;Yes, reset variables
                STA         TIMEH
                LDA         P_OR_F
                STA         PITCH
                JMP         TONE
        $1      JMP         FINISH

;-------------------------------------------------------------------------
;Tone with volume routine by Mark Kriegsman as revised by Bob Sander-Cederlof,
;from Apple Assembly Line, June, 1981
;-------------------------------------------------------------------------

VTONE           LDA         VOLUME
                AND         #0F             ;Maximum volume = 15
                STA         VOLUME
                LDX         PITCH
        $1      DEX
                BNE         $5              ;Don't click yet
                LDA         SPEAKER         ;Click
                LDX         PITCH           ;Reset main delay
                LDY         VOLUME          ;Click speaker again after
        $3      NOP                         ;a variable additional delay
                NOP                         ;allowing a variable "recovery"
                DEY                         ;time.
                BPL         $3
                LDA         SPEAKER
                LDY         VOLUME          ;What counts down must count
        $4      NOP                         ;up for equal timing...
                INY
                CPY         #10
                BCC         $4
        $5      LDY         #0A             ;A brief additional delay
        $6      DEY
                BNE         $6
                DEC         TIMEL           ;Finished yet?
                BNE         $1              ;No...
                DEC         TIMEH
                BNE         $1              ;No...
                DEC         REPS            ;Do it again?
                BEQ         $7              ;No, exit
                LDA         TIME_IN         ;Yes, reset timer
                STA         TIMEH
                JMP         VTONE
        $7      JMP         FINISH

;-------------------------------------------------------------------------
;Dual tone routine from Mark Kriegsman as revised by Bob Sander-Cederlof,
;Apple Assembly Line, June, 1981.
;-------------------------------------------------------------------------

D_TONE          LDX         PITCH           ;Two timing loops in this one
                LDY         PITCH2
        $1      DEX
                BEQ         $2              ;Time to toggle speaker
                NOP                         ;Equalize timing
                NOP
                LDA         #0
                BEQ         $3              ;Always
        $2      LDA         SPEAKER
```

```
                LDX     PITCH           ;Reset loop 1
        $3      DEY                     ;Tone 2
                BEQ     $4              ;Time to toggle for Tone 2
                NOP                     ;Equal timing
                NOP
                LDA     #0
                BEQ     $5              ;Always
        $4      LDA     SPEAKER
                LDY     PITCH2          ;Reset loop 2
        $5      DEC     TIMEL           ;Finished?
                BNE     $1              ;No
                DEC     TIMEH
                BNE     $1              ;No
                DEC     REPS            ;Do it again?
                BEQ     FINISH          ;No, exit
                LDA     TIME_IN         ;Yes, reset time to orig. value
                STA     TIME_H
                JMP     DTONE

;---------------------------------------------------------------------

FINISH          LDA     N_SWITCH        ;Did we really make a noise?
                BEQ     $1              ;No, leave environment alone
                PHP                     ;Yes, restore previous
                SEI                     ;environment
                LDA     OLD_ENV
                STA     ENVRMT
                PLP
        $1      PUSH    RETURN          ;Return to Pascal
                RTS

;---------------------------------------------------------------------
;The following two procedures just turn N_SWITCH on and off
;---------------------------------------------------------------------
;Pascal Procedure Sound_On
;---------------------------------------------------------------------

                .PROC   SOUNDON

                .DEF    N_SWITCH        ;Referenced by other proc's

BEGIN           LDA     #01
                STA     N_SWITCH
                RTS

N_SWITCH        .BYTE   01              ;Default is "ON"

;---------------------------------------------------------------------
;Pascal Procedure Sound_Off
;---------------------------------------------------------------------

                .PROC   SOUNDOFF

                .REF    N_SWITCH

BEGIN           LDA     #00
                STA     N_SWITCH
                RTS


                .END
```

Program Listing #2

```
PROGRAM NOISEFINDER;

{ ************************************************************************** }
{ *                                                  ------------------- * }
{ * NOISEFINDER                                    | Copyright 1984 by | * }
{ * ----------                                    |  O N   T H R E E  | * }
{ * by Al Evans                                    |    Volume 2, #2   | * }
{ *                                                 ------------------- * }
{ * This program lets you use the arrow keys and a joystick to vary the  * }
{ * parameters to the Noise procedure.  Using this program you will be able * }
{ * to find just the noise you are looking for and print out the paramaters * }
{ * used in making that noise.                                           * }
{ *                                                                      * }
{ ************************************************************************** }

USES Applestuff;

TYPE Noise_Type = (N,T,D,V);   {Noise, Tone, Double Tone, Volume Tone}

VAR N_TYPE: Noise_Type;
    REPS, TIME, CTR_P_OR_F, P_OR_F, D_V_P2: INTEGER;
    X, Y: INTEGER;
    B0, B1, RPT, S_ON: BOOLEAN;
    KEY: CHAR;
    OUT_FILE: TEXT;
    ORIG_Y, JS_X, JS_Y, PF_MOD: INTEGER;

PROCEDURE Noise (N_TYPE: Noise_Type; REPS, TIME, P_OR_F, D_V_P2: INTEGER);
        EXTERNAL;

PROCEDURE Sound_On; EXTERNAL;

PROCEDURE Sound_Off; EXTERNAL;

PROCEDURE Read_Out;
BEGIN
  WRITE (OUT_FILE, 'Type: ');
  CASE N_TYPE OF
    N: WRITELN (OUT_FILE, 'NOISE');
    T: WRITELN (OUT_FILE, 'TONE');
    D: WRITELN (OUT_FILE, 'DOUBLE TONE');
    V: WRITELN (OUT_FILE, 'VOLUME TONE');
  END;
  WRITELN (OUT_FILE, 'Repetitions: ',REPS);
  WRITELN (OUT_FILE, 'Time: ', TIME);
  CASE N_TYPE OF
    N: WRITE (OUT_FILE, 'Filter: ');
    T,V,D: WRITE (OUT_FILE, 'Pitch: ')
  END;
  WRITELN (OUT_FILE, P_OR_F);
  CASE N_TYPE OF
    N,T: WRITE (OUT_FILE, 'Delta: ');
      V: WRITE (OUT_FILE, 'Volume: ');
      D: WRITE (OUT_FILE, 'Pitch 2: ');
  END;
  WRITELN (OUT_FILE, D_V_P2)
END; { Of PROCEDURE Read_Out }


PROCEDURE Display;
BEGIN
  REWRITE (OUT_FILE, '.CONSOLE');
  {Console command: move to 0,10 and clear to end of screen}
  WRITE (OUT_FILE, CHR (24), CHR (0), CHR (25), CHR (10), CHR (29));
  Read_Out;
  CLOSE (OUT_FILE)
END; { Of PROCEDURE Display }


PROCEDURE Print;
VAR NAME: String;

BEGIN
  REWRITE (OUT_FILE, '.PRINTER');
  WRITE ('Name for this sound: ');
  READLN (NAME);
  WRITELN (OUT_FILE, 'Sound: ',NAME);
  Read_Out;
  WRITELN (OUT_FILE);
  CLOSE (OUT_FILE)
END; { Of PROCEDURE Print }


PROCEDURE Read_Key;
BEGIN
  REPEAT
    READ (KEYBOARD, KEY);
    IF KEY = CHR (21) THEN TIME:= TIME + 1;
    IF KEY = CHR (8) THEN TIME:= TIME - 1;
    IF TIME ) 255 THEN TIME:= 1;
    IF TIME ( 1 THEN TIME:= 255;
    IF KEY = CHR (11) THEN CTR_P_OR_F:= CTR_P_OR_F + 1;
    IF KEY = CHR (10) THEN CTR_P_OR_F:= CTR_P_OR_F - 1;
    IF CTR_P_OR_F ) 255 THEN CTR_P_OR_F:= 0;
    IF CTR_P_OR_F ( 0 THEN CTR_P_OR_F:= 255;
    P_OR_F:= (CTR_P_OR_F + PF_MOD) MOD 256;
    IF KEY IN ['P','p'] THEN Print;
    IF KEY IN ['1'..'9'] THEN REPS:= ORD (KEY) - ORD ('0');
    IF KEY IN ['M','m'] THEN RPT:= FALSE;
    IF KEY IN ['R','r'] THEN RPT:= TRUE;
    IF KEY IN ['N','n'] THEN N_TYPE:= N;
    IF KEY IN ['T','t'] THEN N_TYPE:= T;
    IF KEY IN ['V','v'] THEN N_TYPE:= V;
    IF KEY IN ['D','d'] THEN N_TYPE:= D;
    IF KEY IN ['S','s'] THEN
      BEGIN
        IF S_ON THEN Sound_Off
          ELSE Sound_On;
        S_ON:= NOT S_ON
      END;
  UNTIL NOT KEYPRESS;
  Display
END; { Of PROCEDURE Read_Key }


BEGIN { Main Program }
  Randomize;
  Joystick (0, D_V_P2, ORIG_Y, B0, B1);
  JS_Y:= ORIG_Y;
  N_TYPE:= N;
  REPS:= 1;
  TIME:= 10;
  CTR_P_OR_F:= ORIG_Y;
```

```
  PF_MOD:= 0;
  RPT:= TRUE;
  S_ON:= TRUE;
  WRITE (CHR (28));
  WRITELN ('Apple /// noise finder by Al Evans (c) 1984 by ON THREE');
  WRITELN;
  WRITELN ('N: Noise; T: Tone; V: Volume tone; D: Dual tone; S: Sound on/off');
  WRITELN
  ('M: make noise, R: repeat noise, 1..9: Repetition factor, P: Print params');
  WRITELN ('Time: (- and =)');
  WRITELN ('Pitch/Filter: up and down arrows +/- Joystick vertical');
  WRITELN ('Delta/Volume/Pitch2: Joystick horizontal');
  WRITELN;
  WRITELN ('Press Q to leave the program.');
  Display;
  REPEAT
    P_OR_F:= (CTR_P_OR_F + PF_MOD) MOD 256;
    Noise (N_TYPE, REPS, TIME, P_OR_F, D_V_P2);
    IF RPT THEN
      BEGIN
        Joystick(0, D_V_P2, JS_Y, B0, B1);
        IF N_TYPE = V THEN
          D_V_P2:= D_V_P2 DIV 16
      END;
    PF_MOD:= JS_Y - ORIG_Y;
    IF ((KEYPRESS) OR (NOT RPT)) THEN
      Read_Key;
  UNTIL KEY IN ['Q','q']
END. { Of PROGRAM NOISEFINDER }
```

Business Basic Program Lister

by George N. Oetzel

Even with an effort to produce readable code, **BASIC** programs are often difficult to read. Practices that make programs compact and increase their speed also tend to reduce readability. For example, it is both convenient and efficient to join numerous short assignment statements with colons. Colons used to join conditional statements after an **IF...THEN** are often better than uninformative **GOTO** statements. Even though there is a good reason for each multiple statement, the program gradually grows more difficult to read. Use of the formatter described in this article will significantly improve the readability to your BASIC listings.

Listings #1 and **#2** illustrate the format. Remark statements are left-most, and they are self-formatting up to the 255-character limit set by BASIC, which is a bit more than three 80-character lines. Lines after the first in multi-line remarks begin with an asterisk and "(REM)" to indicate their purpose. The characters "*—" at the end of a remark line tell the program to fill the rest of the line with asterisks. This provides an easy way to get just the right number of asterisks into a listing. The number will change if you change the margins on your listing format.

Executable statements are indented, and they are broken and further indented at each colon. Business BASIC provides extra indenting of **FOR-NEXT** loops, and that extra is preserved. Checking the content of statements after colon separators is limited to additional colons, so there is no special treatment of remarks or extra indenting of statements inside FOR-NEXT loops when they occur after colons.

Each page has a header that gives the file pathname, date, and page number. The program listing given includes manual entry of the date, but it can be automated with DATE$ if you have the **ON THREE** O'Clock clock/calendar (or equivalent) installed. Experience with the **Apple][** made me concerned that a listing program in BASIC would be so slow that the performance would be unacceptable. The Apple][listing formatter produced by R. C. Clardy, et al. (**Call-A.P.P.L.E in Depth**, p. 102) produces good listings, but it doesn't keep the printer very busy. With only a little effort devoted to a maximum performance program, it was a pleasant surprise to discover that this program produces listings fast enough to keep my Epson MX-80 operating full-time.

The listing speed was so surprising that I converted the program to run in Applesoft, using the Emulation mode, simply to find out how the speeds compared. The results, described later, will make any **Apple ///** owner appreciate his investment.

The most straightforward approach to a listing formatter requires that you first capture Apple's version of the listing as a text file on a disk. The printed output is then formatted from the text file. The drawback of this approach is the extra step required to make the text file. The effort is not a complete waste, however, because you can then use your favorite text editor to edit the program. Utilities like those provided with **Apple Writer ///** can be used to transfer Apple][text files to the Apple ///, so the same formatting program can be used to list both Apple][and Apple /// programs.

The Business Basic reference manual describes the technique for capturing programs as text files (pages 33 and 34). It is useful to have two versions, with small and large line numbers, so you can capture your BASIC program even if you needed the extra-small line numbers to make room for some programming afterthought. You can enter the **EXEC** files with your text editor, or you can use the program shown in Listing 1 to build both. Instructions are given in REM statements that all are line number 0. They were added in the text editor before making the formatted listing.

The listing program begins with **GOTO 5000**, so the statements that are executed repeatedly during formatting have small line numbers. This organization speeds the execution of GOTO and GOSUB statements. The initialization, beginning at line 5000, presents a menu with choices for the printing format. The defaults presented are my favorites, with a left margin of 5 spaces and 70 printed characters per line. The defaults are set in statement 5060. The symbol definitions are in lines 5100-5160. Two-letter variable names were used to improve the execution speed and to facilitate the conversion to an Applesoft version of the program.

One of the nicest features of **SOS** is the handling of the destination for the formatted listing. It is entered as a string variable in subroutine 7800; the subsequent treatment in lines 5350-5360 is the same for all destinations — printer, console, or disk file.

A keyboard interrupt to abort the listing is initialized in Line 5320. This feature is quite handy if you discover early in a long listing that you didn't set up the format as intended, or you inadvertently asked for the wrong program file.

Most of the actual formatting is in lines 100-400. A few lines deserve special comments:

100 INPUT#4;A$

inputs a full line, even if it contains commas. This differs from Applesoft, where an "EXTRA IGNORED" message would result.

Line #110 determines the length of the input string. The first line output by the capture utility consists only of a RETURN, which is a zero-length string, so it is ignored. Initialize the quote flag, Q, which is used so that colons inside quotations are not treated as statement separators. (For example, see line 5000.)

120 A=INSTR(A$,S$,2)

finds the first space after the line number. The first character in each line is a space, so the search starts with the second character.

130-140 Right justify the line number in a six-space field.

150-170 Check to see if the statement is a remark.

190-400 Put together the rest of the line, character by character.

210-220 Check for expressions in quotations.

230 Check for colon statement separators.

240-360 Decide where to split lines that are longer than LL characters. They are separated at one of the characters in PU$=" ;,+-*/:" provided that this doesn't result in too short a line.

390 GOSUB 50 calls the subroutine to print the line that has accumulated in B$.

The lines beginning at 1000 handle remarks. The special treatment of "*—" is in lines 1000, 1090, and 1100. Most of the formatting of remarks is concerned with breaking up long lines. As with executable statements, PU$=" ;,+-*/:" defines the characters that are used to separate the lines. The prefix "* (REM)" is added to the beginning of extended remark lines in line 1080.

Conversion of the program to run in Apple][Emulation

highlighted the advantages of the Apple /// operating system. The **Apple /// version** of this program listed itself in **4 minutes, 45 seconds**, while the **Apple][version required 8:30**. Numerous computational bench marks have shown that the faster 6502 clock in the Apple /// just barely makes up for the memory management overhead. The Apple /// does numerical computations only slightly faster than the Apple][. The big difference in the operation of this program is due to the design of the operating system. SOS lives up to its official name, in spite of all the jokes the "sauce" pun may have inspired. It really is a **Sophisticated Operating System**.

When the Apple][version of the program runs, there is a quite noticeable pause between printed lines, and an even more significant pause when it is necessary to read more text from the disk. On the Apple ///, however, the printer doesn't pause between lines; it usually continues to print even while the disk is running. The reason is that SOS uses buffering and interrupts in I/O operations, so the processor is tied up with these activities for the minimum possible time.

It may be that you aren't familiar with buffered I/O and interrupts. If so, here's a simplified rundown on how the system works.

When you open the **.PRINTER** file, SOS allocates a memory buffer in which to store data that is ready for output. When the program completes computation of an output line, BASIC tells SOS there is output ready for the printer. SOS stores the data in the file buffer so your program can continue working on its problem without destroying the output, and then inquires if the printer is busy. If the printer is still working on the previous line, SOS doesn't keep the 6502 tied up waiting for the printer. It tells the printer driver to interrupt whatever else is going on as soon as the printer is ready, and the data will be transferred then. In the meantime, SOS returns use of the 6502 to BASIC and your program.

The interrupt occurs as a result of a hardware signal from the printer to the interface. For example, with a parallel interface, one of the wires between the interface and printer carries a voltage that is high when the printer is busy and falls to near zero when the printer can accept data. The change of state on the busy line triggers the interrupt, which forces an immediate transfer of control to an interrupt-handling subroutine that transfers the data to the printer. Since it doesn't take very long to transfer one line of print (assuming that your printer has a one-line print buffer), control of the 6502 returns quickly to your program. You can get several lines ahead of the printer, because SOS can buffer a number of lines of output.

The short program shown below illustrates the SOS buffering.

```
10    OPEN#1 AS OUTPUT,".printer"
20    PRINT CHR$(7);"start"
30    FOR i=1 TO 12
40      PRINT#1;"This is line number ";i;" printed
          on the printer for a buffered I/O test."
50    NEXT
60    PRINT CHR$(7);"Loop execution complete"
70    CLOSE#1
```

On my system, the **Loop execution complete** message appears on the screen while the printer is printing, "This is line number 8." Now we can understand why printing doesn't stop, even when the program requires new input from the disk. The input operation is a few lines ahead of the printer, and the printer has a small buffer of its own. Reading a new 512-byte block from the disk may take 200 ms or so, but it can take nearly a full second for the printer to complete a line. Disk operations are usually completed while the printer is in the middle of a line of print, and the program has more output ready before the SOS buffer is empty. As a result, there is always data ready whenever the printer can accept more. ///

Program Listing #1

```
0   REM ************************************************************
0   REM *                                        ------------------ *
0   REM *  Business Basic Capture Utility       | Copyright 1984 by | *
0   REM *  ------------------------------       |  O N   T H R E E  | *
0   REM *  by George Oetzel                     |     Volume 2, #2  | *
0   REM *                                       ------------------ *
0   REM *  This program will produce two EXEC files to capture Business *
0   REM *  Basic programs as text files on disk.                      *
0   REM *                                                           *
0   REM *  Type in this program and first RUN the low and then RUN the  *
0   REM *  high line numbers to create the EXEC files "CAPTUREHI" and   *
0   REM *  "CAPTURELO".                                               *
0   REM *                                                           *
0   REM *  Exec whichever version that doesn't destroy existing line    *
0   REM *  numbers.  Then type RUN (to start at line 1) or GOTO 63991   *
0   REM *  to create the text file.                                   *
0   REM ************************************************************
1   ON ERR GOTO 6
2   INPUT"LIST TO WHAT FILE PATH NAME? ";FI$
3   CREATE FI$, TEXT:OPEN#1 AS OUTPUT,FI$:OUTREC=255:OUTPUT#1
4   LIST 10 TO 63999:PRINT
5   CLOSE:OUTREC=80:END
6   PRINT"FILE ";FI$;" ALREADY EXISTS - DELETE BEFORE RUNNING":END
63991   ON ERR GOTO 63996
63992   INPUT"LIST TO WHAT FILE PATH NAME? ";FI$
63993   CREATE FI$, TEXT:OPEN#1 AS OUTPUT,FI$:OUTREC=255:OUTPUT#1
63994   LIST 0 TO 63990:PRINT
63995   CLOSE:OUTREC=80:END
63996   PRINT"FILE ";FI$;" ALREADY EXISTS - DELETE BEFORE RUNNING":END
```

Program Listing #2

```
0   REM  ************************************************************
1   REM  *                                      ------------------ *
2   REM  *  Business Basic Program Lister      | Copyright 1984 by | *
3   REM  *  ----------------------------      |  O N   T H R E E  | *
4   REM  *  by George Oetzel                   |     Volume 2, #2  | *
5   REM  *                                      ------------------ *
6   REM  *  This program produces formatted listings of BASIC       *
7   REM  *  programs that have been captured as Apple /// textfiles, *
8   REM  *  using the CAPTURE utility or other means.               *
9   REM  ************************************************************
10   REM *'
11   REM
12   REM          Listing formatter for BASIC programs.
13   REM                    George Oetzel
14   REM
15   REM     This program produces formatted listings of BASIC programs that h
     ave been captured as Apple /// textfiles, using the CAPTURE utility or ot
     her means.
16   REM
17   REM *'
18   REM
20   GOTO 5000
45   REM *'
46   REM
47   REM PRINT-A-LINE SUBROUTINE
48   REM
49   REM *'
50   PRINT TAB(LM);B$:LC=LC+1:M=0:IF LC)PL THEN GOSUB 6000
60   RETURN
90   REM *'
91   REM
92   REM MAIN PROGRAM BODY -- READ A LINE; DETERMINE IF IT IS A REMARK; IF NOT
     , THEN CHECK THE PROGRAM LINE FOR COLON SEPARATORS AND LENGTH; FINALLY, P
     RINT THE LINE.
93   REM
94   REM *'
100   INPUT#4;A$
110   L=LEN(A$):Q=0:IF L=0 THEN 100
120   A=INSTR(A$,S$,2):B$=LEFT$(A$,A):N=A+1
130   B$=S4$+B$:B$=RIGHT$(B$,6)
140   A$=B$+RIGHT$(A$,L-A):M=LEN(B$):N=M+1:L=LEN(A$)
150   C$=MID$(A$,N,1):IF C$=S$ THEN B$=B$+C$:N=N+1:GOTO 150
160   C$=C$+MID$(A$,N+1,2):N=N+3
170   IF C$="REM" THEN 1000
180   B$=B$+S4$+C$:M=LEN(B$)
190   FOR I=N TO L
```

```
200   C$=MID$(A$,I,1):B$=B$+C$:M=M+1
210   IF C$=Q$ THEN Q= NOT Q
220   IF Q THEN 240
230   IF C$=":" THEN GOSUB 50:B$=S8$:M=LEN(B$)
240   IF M<LL THEN 370
260   FOR R=LL TO PM STEP-1
270     A=INSTR(PU$,MID$(B$,R,1),1):IF A>0 THEN A=R:R=PM
290     NEXT
300   IF A<PM OR(A=M) THEN GOSUB 50:B$=S8$:M=LEN(B$):GOTO 370
350   C$=RIGHT$(B$,M-A):B$=LEFT$(B$,A):GOSUB 50
360   B$=S8$+C$:M=LEN(B$)
370   NEXT I
380 X= FRE
390 GOSUB 50
400 GOTO 100
410 IF PD$=".PRINTER" THEN PRINT CHR$(12);
420 REM IF PD$=".PRINTER" THEN FOR I=1 TO LPP-TM-HL-LC:PRINT:NEXT       (ALT
    ERNATE VERSION OF FORM FEED)
430 CLOSE#4:OFF EOF#4:OFF KBD:CLOSE#1
440 PRINT:PRINT"LISTING COMPLETE"
450 IF P$<>"" THEN PREFIX$=P$
460 END
990 REM *
991 REM
992 REM THIS SECTION FIXES UP REMARK STATEMENTS, WHICH HAVE DIFFERENT LENGTH
     AND SEPARATION RULES FROM EXECUTABLE STATEMENTS.
993 REM
994 REM *
1000  IF RIGHT$(A$,2)="*" THEN 1090
1010  IF L<LL THEN SWAP A$,B$:GOSUB 50:GOTO 100
1020  B$=LEFT$(A$,LL)
1030  FOR R=LL TO PM STEP-1
1040    A=INSTR(PU$,MID$(B$,R,1),1):IF A>0 THEN A=R:R=PM
1050    NEXT
1060  IF A<PM THEN GOSUB 50:A=LL:GOTO 1080
1070  B$=LEFT$(B$,A):GOSUB 50
1080  A$="    *  (REM) "+MID$(A$,A+1,L-A):L=LEN(A$):GOTO 1000
1090  B$=LEFT$(A$,L-1):N=LL-L+1:IF L>LL THEN 1020
1100  FOR I=1 TO N:B$=B$+"*":NEXT:GOSUB 50:GOTO 100
4900  REM *
4910  REM
4920  REM INITIALIZE EVERYTHING.  ESTABLISH ALL VARIABLES AND PRESENT A MENU
     TO SET UP LISTING FORMAT.  OPEN FILES, AND SET UP KEYBOARD INTERRUPT TO
     ABORT LISTING GRACEFULLY.
4930  REM
4940  REM *
5000  S$=" ":Q$=CHR$(34):PU$=" ;,+-*/:"
5010  SPC$="
5020  S4$=LEFT$(SPC$,4):S8$=LEFT$(SPC$,16)
5030  PD$=".PRINTER"
5040  M=0:I=0:Q=0:N=0:A=0:LC=0
5050  LL=70:LM=5:PL=55:BM=5:TM=2:LPP=66:HL=3:PG=1
5060  TEXT:HOME:T=30
5070  PRINT"LIST PARAMETERS"
5080  PRINT
5090  PRINT"1.  LEFT MARGIN ="; TAB(T);LM
5100  PRINT"2.  LINE LENGTH ="; TAB(T);LL
5110  PRINT"3.  TOP MARGIN ="; TAB(T);TM
5120  PRINT"4.  HEADER LINES ="; TAB(T);HL
5130  PRINT"5.  PROGRAM LINES ="; TAB(T);PL+1
5140  PRINT"6.  BOTTOM MARGIN ="; TAB(T);BM
5150  PRINT"7.  LINES PER PAGE ="; TAB(T);LPP
5160  PRINT"8.  OUTPUT DEVICE ="; TAB(T);PD$
5170  PRINT:TEXT
5180  PRINT"ENTER NUMBER OF ITEM YOU WANT TO CHANGE, RETURN FOR NONE ";:GET A
    $:PRINT A$
5190  IF A$=CHR$(13) THEN 5250
5200  A=ASC(A$)-48
5210  IF(A<1) OR(A>8) THEN 5060
5220  WINDOW 1,18 TO 80,24
5230  ON A GOSUB 7100,7200,7300,7400,7500,7600,7700,7800
5240  HOME:GOTO 5060
5250  PM=INT(0.65*LL)
5260  TEXT:HOME:VPOS=5:HPOS=10
5270  INPUT"TODAY'S DATE:  ";DT$
5275  REM ** with a clock, use DT$=DATE$
5280  DL=LEN(DT$)
5290  PRINT:HPOS=10
5300  PRINT"If you enter a prefix (/VOLUMENAME/) or drive (.D2) below, the he
    ader"
5310  PRINT TAB(10);"will give a complete description of the file location"
5320  PRINT:HPOS=10
5330  INPUT"ENTER THE PREFIX OR DRIVE: ";A$
5340  PRINT:HPOS=10
5350  INPUT"ENTER THE FILENAME: ";B$
5360  FILE$="":P$=""
5370  FOR I=1 TO LEN(B$)
5380    X=ASC(MID$(B$,I,1))
5390    IF X>96 THEN X=X-32
5400    FILE$=FILE$+CHR$(X)
5410    NEXT
5420  IF A$="" THEN 5440
5430  P$= PREFIX$:PREFIX$=A$:FILE$= PREFIX$+FILE$
5440  ON KBD GOTO 8000
5450  OPEN#4 AS INPUT,FILE$
5460  ON EOF#4 GOTO 410
5470  OPEN#1 AS OUTPUT,PD$
5480  OUTPUT#1
5490  GOSUB 6020
5500  GOTO 100
5900  REM *
5905  REM
5910  REM BOTTOM MARGIN, TOP MARGIN, AND HEADER TREATMENT
5920  REM
5930  REM *
6000  IF BM<1 THEN 6020
6010  FOR F=1 TO BM:PRINT:NEXT
6020  IF TM<1 THEN 6040
6030  FOR F=1 TO TM:PRINT:NEXT
6040  IF HL=0 THEN LC=0:RETURN
6050  PRINT TAB(10);"FILE = ";FILE$; TAB(LM+LL-DL-13);DT$;S4$;"PAGE ";PG
6060  PG=PG+1:LC=0:IF HL-1<1 THEN RETURN
6070  FOR F=1 TO HL-1:PRINT:NEXT
6080  RETURN
6900  REM *
6910  REM
6920  REM HERE WE HAVE THE SUBROUTINES THAT HANDLE THE FORMAT MODIFICATION ME
    NU.  THE HUNDREDS DIGIT IS ALSO THE MENU SELECTION NUMBER.  7900 IS THE
     MENU ERROR HANDLER.
6930  REM
6940  REM *
7100  INPUT"NEW LEFT MARGIN:  ";LM
7110  RETURN
7200  INPUT"NEW LINE LENGTH:  ";LL
7210  RETURN
7300  INPUT"NEW TOP MARGIN:  ";XTM
7310  XPL=LPP-XTM-HL-BM-1
7320  IF XPL<5 THEN 7900
7330  TM=XTM:PL=XPL
7340  RETURN
7400  INPUT"NEW HEADER LENGTH, PRINTED LINE PLUS SPACES:  ";XHL
7410  XPL=LPP-TM-XHL-BM-1
7420  IF XPL<5 THEN 7900
7430  PL=XPL:HL=XHL
7440  RETURN
7500  PRINT"To expand the number of program lines, first reduce top and botto
    m margins,"
7510  PRINT"or change the number of lines per page."
7520  PRINT:INPUT"NUMBER OF PRINTED PROGRAM LINES:  ";XPL
7530  XLPP=XPL+TM+HL+BM+1
7540  IF XLPP>LPP THEN 7900
7550  PL=XPL-1
7560  RETURN
7600  INPUT"NEW BOTTOM MARGIN:  ";XBM
7610  XPL=LPP-TM-HL-XBM-1
7620  IF XPL<5 THEN 7900
7630  PL=XPL:BM=XBM
7640  RETURN
7700  INPUT"NEW PAGE LENGTH, total of margins, header, and program lines: ";X
    LPP
7710  XPL=XLPP-TM-HL-BM-1
7720  IF XPL<5 THEN 7900
7730  LPP=XLPP:PL=XPL
7740  RETURN
7800  INPUT"NEW OUTPUT DEVICE OR FILE PATHNAME: ";XPD$
7810  ON ERR GOTO 7860
7820  OPEN#8 AS OUTPUT,XPD$
7830  CLOSE:OFF ERR:PD$=XPD$
7840  RETURN
7860  HOME:VPOS=5:PRINT TAB(20);"BAD PATH NAME"
7870  VPOS=1:GOTO 7800
7900  HOME
7910  PRINT"You must have made a mistake.  That value gives ";XPL+1;" printed
     program"
7920  PRINT"lines per page."
7930  PRINT:PRINT TAB(T);"HIT ANY KEY TO CONTINUE"
7940  GET A$:PRINT
7950  RETURN
8000  CLOSE:OFF EOF#4:OFF KBD
8010  PRINT:PRINT"LISTING INTERRUPTED FROM KEYBOARD"
8020  IF P$<>"" THEN PREFIX$=P$
8030  END
```

Sample Output

```
  FILE = /PRO/A3LIST.L                1 Sep 84 11:43 AM    PAGE 1

 10   REM ************************************************************
 11   REM
 12   REM           Listing formatter for BASIC programs.
 13   REM                      George Oetzel
 14   REM
 15   REM     This program produces formatted listings of BASIC
  *  (REM) programs that have been captured as Apple /// textfiles,
  *  (REM) using the CAPTURE utility or other means.
```

Continued on page 5.

Review ON: Apple Speller ///

by Dr. Michael Sexson

When I bought my Apple ///, I thought I would never need a spelling program. After all, as a fifth-grade spelling bee champion and as a current university professor of English, what would I need one of those things for? My spelling, if not perfect, is better than most, and my typing skills are excellent. After a year with the ///, however, I noticed that I seemed to be making more typographical errors than I ever had with my Selectric. Even after proofing the document carefully on the monitor, I found the impish things turning up by the squadron on the printed page. Apparently it is common to miss typing errors while reading a monitor. Also, I have a habit, given the light touch on the /// keyboard, to run words together. I was always having to reprint the same document after finding typos in what I hoped was the **final** draft.

Since I knew that a spell-checker not only caught misspellings--a crime I could never be guilty of--but typographical errors, I decided to invest. After a considerable wait between the rumor of a speller program for the /// and its actual manifestation at my friendly Apple computer store, I wrote out my check in fear and trembling (you'll know why later), took the intimidating box home, opened it, and took the first step toward determining whether it was worth the price.

After two months of fairly constant use, I am ready to render a verdict on the program and on my former belief that I did not need such a thing. Last things first. I do need it. If you are in a profession which demands the creation of squeaky-clean prose, free of misspellings and typographical errors, you'll need a spell-checker. I ran a half-dozen **finished** documents through the speller, and found, to my horror, that every one of them contained typos. Then I found that some of them contained misspellings. Omelet does not have two l's; benefited does not have two t's; publicly does not end **cally**; and so on an on, to my utter embarrassment (two r's, not one). A good spell-checker not only find typos; it bestows humility.

But what about the Apple Speller ///? Created for Apple Computer by Jonathan Spence, it is an extremely fast, very sophisticated program containing 80,000 words based on the concise edition of the Random House Dictionary. It is easy to learn and the documentation is superb--clean, simple, direct prose with lots of illustrations. Although it works on any Apple /// with any word processing software that uses SOS, it is intended to take advantage of the newest version (2.0) of the Apple Writer /// with 256K. With this setup, you can enter the speller directly from the Apple Writer file you are dealing with. Those of us with the earlier version of Apple Writer and/or the 128K system must exit Apple Writer, load the speller program, check the spelling (it is almost essential to have two drives), exit the program, re-load the word processor, and fix the document. It's really not as cumbersome as it sounds, but anyone who has purchased the now standard 256K system and intends to upgrade to the latest version of Apple Writer will discover what a treat they have if they invest in the speller.

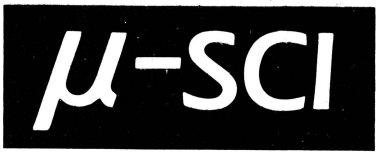
The total operation is very simple. One keeps the file to be checked in Drive #2, loads the speller, and types C for **Check the spelling of a document**. The program brings up your catalog and asks for a filename. After you supply it, the speller goes to work counting the number of unique words in the document. Within seconds, the program will assert (for example) **1897 words were read from the document. Excluding duplicates, you have 1367 unique words**. If you like, you can see the unique words listed on the screen--for some of us, a graphic demonstration of the poverty of our vocabulary.

The program then checks the words against its main dictionary, which contains approximately 40,000 common words. After spending about a minute comparing your words with the dictionary's, it then allows you to use other dictionaries containing less common, and even quite rare words. Three discs are included with the package, the program disc, and two dictionary discs. On the second dictionary disc are the less common and rare words, numeric forms, place names, proper names, and abbreviations. Spence argues (and after working with the Speller ///, I agree), that 90 per cent of the time you'll find you need only the main dictionary of common words to get most jobs done.

After the comparing is done, the fun begins. The program announces that, (say) **10 document words have not yet been found in a dictionary**, meaning that ten words in your text have no corresponding words in the program's dictionary. You are now permitted to view those words. A portion of your text is displayed in a rectangle called the **Document Box** at the top of the monitor while two large squares occupy the rest of the screen. In the first square is a command menu with a series of options and the second is blank. A highlighted cursor sits on the first word of your text that doesn't match. You can: (1) Have the program guess at a spelling of the word; (2) List words from the dictionary that contain one or more letters similar to the one in question; (3) Add the word to a dictionary; (4) Ignore the word and go on; (5) Replace it with a new word; or (6) Mark it as misspelled. Most of the time, you'll probably **Ignore it** and go on since the word that doesn't match is a proper name, abbreviation, or place name that's not in the program's main dictionary. You do have the option, of course, of creating your own dictionary filled with these sorts of things so that you need never linger over them again. In fact, you may create as many other dictionaries as you want, containing specialized words appropriate to your needs.

When you do come to a typo or misspelled word, it's like finding a treasure. Aha! The highlighted word is **mysogyny**. You press G to have the program guess the proper spelling. Magically, the second square puts on its display and there is the word you want, spelled properly: misogyny. Or, if you want, you can press L and type =gyny to find all words in the main dictionary that end with those four letters, and in seconds there they will be: androgyny and misogyny. One of the delightful distractions of the program is that a logophile (a word you'll have to add to your own dictionary) will get so engrossed in the display of dictionary words that he'll forget he's supposed to be proofreading. What philologist (yes, it's there, on dict. 2) could resist scanning through the arcane (also on dict. 2) treasures of **Random.House.3** : **baht**,

ON THREE Presents . . .



MICRO-SCI

Micro-Sci Disk Drives

Every once in a while a product appears that is so good **ON THREE** decides to offer it for sale to our readers. ***The Micro-Sci*** line of disk drives (and the ***Gameport ///***) are the first of these superior type products. Byte for byte, these drives offer greater speed and more value than any comparable drive on the market today. If you are looking into purchasing an external disk drive for your ///, **ON THREE** encourages you to look into this fantastic product line.

Expanding disk storage on the Apple /// can be an expensive proposition.

But ***Micro-Sci*** has a better proposition for you, because our disk drives for the Apple /// give you greater capacity and performance for every dollar spent.

And there are no compatibility problems. The A3 is a direct replacement for Disk /// drives, and the 70-track A73 and 140-track A143 are supplied with a driver that is easily added to the SOS driver module, affording extra storage and fast seek rates for all of the programs that run under SOS.

Talk about compatible! All three are the same size as your built-in drive and they use the same diskettes!

Are all of your slots full? Don't worry, these drives plug right into the back of your /// and they don't need a power cord! Up to three extra disk drives can be daisy-chained and they can be mixed in any combination of Disk ///, A3, A73 or A143.

The A3 offers identical capacity to the Disk /// and is an excellent choice for a second disk compatibility in the Apple][emulation mode.

At 286 KBytes, the A73 has double the capacity of the Disk /// while the ***A143 packs 572 KBytes*** of data onto a diskette. With over ***half a megabyte*** of storage space, the A143 makes a truly viable backup device for the Profile Hard Disk.

With that large a capacity, many people find that they ***don't need a hard disk!*** Since up to three A143's can be used with your ///, you can have over one and three quarter megabytes of data on-line at all times!

ON THREE is pleased to announce the following low, low prices on these great disk drives.

	A3	A73	A143
Suggested List Price:	$379	$529	$659
ON THREE Price:	$299	$409	$498
Savings	$80	$120	$161

To order, use the attached envelope and add $6.50 for postage and handling for each drive ordered. Please allow four weeks for delivery.

Gameport ///

You don't have to be chained to your job, and neither does your Apple ///. After the working day is done, release your computer into the exhilarating world of adventure and challenge with a ***Gameport /// from Micro-Sci.*** The new Gameport /// game controller adapter lets you use game paddles, joysticks and all your favorite Apple][amusement packages with your Apple /// computer. The Gameport /// is easy to use and simple to install - your only challenge is to conquer the invaders!

The Gameport ///

- Allows all games written for the Apple][to be used on the Apple ///.
- Works with all Apple][game paddles and joysticks.
- Allows programs which require a game I/O protection key to run in Apple][emulation mode.
- Can be installed in any slot.
- Does not interfere with the normal operation of the Apple ///.
- Package includes: Gameport /// board, Apple][Emulation Modification Diskette and complete, easy-to-follow instructions (Apple][game controllers not included).

ON THREE proudly sells the ***Gameport /// by Micro-Sci.*** For only $59.95 you can now get the best that the Apple /// and the Apple][has to offer. That's $15 off the suggested list price so don't be left out, place your order today! Please use the attached envelope for ordering and remember to add $2.50 for postage and handling.

cabochon, **xebecs**, **quadroons**, **hagiographer**.

Meanwhile, back in the practical world, once you come up with a misspelling or typographical error, you can either replace it (unhappily, on a 128K system, you can only replace words that are the exact number of letters as the highlighted ones), or mark it. After you've finished proofing and adding words to one or more dictionaries, you go through your text to find the marked words. If you have Apple Writer III, you simply use the Find function to locate all instances of the symbol |. You change the words, and voila!, you've got a clean document, but not necessarily squeaky clean. There are things this spell-checker cannot do. It won't catch words which, though wrong, are still to be found in the dictionary. For example, if you write, **I here you singing over their**, the two misspellings, **here** and **their** will pass because they are legitimate dictionary words. Also, the program is not a style analyzer. If you use jargon, are too mechanical, pompous, redundant, boring or inflated, this program cannot help you. There are, I understand, such programs in the works. And you'll have to stand behind me in line to get one. Actually, the program I'm really waiting for is the Oxford English Dictionary on a single floppy. Imagine rummaging through such riches! Additionally, (a word which appears to have been generated by the computer culture), the Apple Speller /// is not a thesaurus. You can't look up synonyms for the word you're using. It simply checks your words against the words in its memory. And that it does very well.

Apple Speller /// also contains a utilities program that allows you to check the dictionaries to see that they are in good working order; combine, subtract, and delete dictionaries; and create empty dictionaries. Most impressive is a feature which allows you to customize the speller, changing it to suit your own needs. You can, for example, have the program recognize special characters such as soft-hyphens, tabbing characters, foreign character sets, etc. This feature, the manual says, is probably only for **technical whizzes**. But since the discs are copyable, even English teachers might risk a venture into these intricate innards just to see what might happen.

For Profile owners, it's possible to combine all the dictionaries into one giant dictionary so you don't have to do so much disc swapping.

All in all, Apple Speller /// is a superb program for people who need to produce error-free documents. It's fast, easy, efficient and fun. Its documentation is excellent. Two drawbacks, however, need to be mentioned. First, the speller can only check up to 2048 unique words at a time. When that figure is reached, the statement **Memory Full** appears and you must then proof your text in chunks. While this is not a major problem, a program of this sophistication should be able to handle longer documents at a single proofing. It is simply not true, as Spence says, that most documents don't contain that many unique words. Half or more of mine do. The second, and most serious drawback, is the price. $175 is a bit steep for a program that is a spelling checker—a most sophisticated one, admittedly, but still exclusively a spell-checker. For that price one might expect a thesaurus function or even a rudimentary style analyzer. I'm not being unrealistic, either. I know there are other programs for other machines selling for less than half of what this costs that not only check spelling but provide synonyms and antonyms. I would also guess that these programs lack this one's sophistication. It would be most regrettable though if the high price tag on the Speller /// kept those who need it most from purchasing it. I can hear these people saying, as I used to, **I'm a pretty good speller and I type all right. I don't need that expensive program**. Well, they would be surprised and humbled to discover just how many typos and misspellings they do commit and how they really do need a good spell-checker like this one. Let's hope Apple tunes in and trims that price so that it can reach the wide audience it deserves.

Review ON: The Art Of Negotiating

by Bob Consorti

Every once in a while something new comes out in the field of micro-computers that is very exciting! **The Art Of Negotiating - Computer Preparation Program** is the latest. It's not a spreadsheet, it's not a word processor, it's not a data base, it's something new. **The Art Of Negotiating** is one of the new type of **Idea Processors**. But that's an understatement, it goes beyond the handling of ideas - it helps you handle real life situations.

Each day of your life you negotiate about something. It may be something as insignificant (in the grand scheme of things) as your pay raise, or as important as a treaty between two nations. Most people lack **ANY** training whatsoever in the all important skill of negotiating. The Art Of Negotiating program will help you make the right decisions.

The program asks questions and offers suggestions about **BOTH** sides of a negotiation. In creating your strategies and agenda for negotiation, the program will not let you get away with poor preparation. The Art Of Negotiation will remind you when you need to ask important questions. It shows the alternatives to the various strategies that you can use and their probable results. After seeing all of the possibilities, you can channel your thinking into the key areas of the negotiation.

This program helps you structure your negotiating energies into methods that have been proven successful. Based upon the The Art Of Negotiating Seminars by **Gerard I. Nierenberg**, this program is based upon the idea that the most successful negotiation is one whereby both sides benefit. The concept of each side in a negotiation winning will foster a long-lasting solution to the object of the negotiation.

The entire package is quite impressive. The book **Fundamentals of Negotiating**, and the **Art of Negotiating Work Text** by Gerard I. Nierenberg are included. These two books are the basis for the program. Gerard I. Nierenberg is one of the foremost authorities on negotiating in the world. If you aren't one of the 115,000 executives who have already attended the Art of Negotiating Seminars, you can now use his methods with the help of your personal computer. The manual for the Computer Preparation Program is excellent. It comes complete with a tutorial that steps you through an imaginary negotiation for a raise.

Continued on page 14.

Call THREE: Hot Line

The **Call THREE: Hot Line** is a service whereby **Apple ///** users with problems can call a relatively local number to get help. The people answering the phones are fellow **Apple ///** users who have volunteered to help others over the rough spots.

If you know enough about your machine to answer questions, please write in so we can include you in a future listing. For those who write in, please state your areas of expertise and the hours during the week you are willing to take calls.

For those of you who need questions answered: **PLEASE** call only within the hours specified. Also note any time zone differences. If these volunteers start getting calls at all hours, the service will have to be discontinued, so **PLEASE** follow this guideline!

Legend

Accounting: AC
Agriculture: AG
Assembly Language: AL
Business Basic: BB
Catalyst: CT
Cobol: CO
CP/M: CP
Data Base: DB
Education: ED
Financial: FI
Fortran: FO
General Use: GE
Graphics: GR
Micro-Sci: MI
Modems: MD
Pascal: PA
ProFile: PR
Quark: QU
SOS: SO
Spread Sheet: SS
Telecom: TC
Word Processing: WP
][Emulation: AE
/// E-Z Pieces: 3P

Under the topics column, there are two character mnemonics that specify the individuals area(s) of expertise. They are outlined above.

User Groups

If you would like to get together with other Apple /// owners and exchange ideas, a user group is for you! Below is a listing of all the Apple /// user groups that we know of. If you have recently started a new one, or if you know of one that is not listed here, please write ***ON THREE*** so that we can let everyone know of them.

CALIFORNIA

Apple /// User's of Northern California
220 Redwood Hwy #184
Mill Valley, CA 94941
(415) 383-0203

Fresno Area Apple /// User Group
4175 N. Blackstone
Fresno, CA 93726
(209) 224-2983

Silicon Valley /// User Group
707 Continental Cir.
Mountain View, CA 94040
(415) 969-6093

Business Apple Group
1850 Union St. #494
San Francisco, CA 94123
(415) 921-3374

Los Angeles-South Bay Apple /// Users Group
c/o Sun Computers
1848 Pacific Coast Hwy.
Lomita, CA 90717
(213) 541-2311

Orange County Apple /// User Group
22501 Eloise Ave.
El Toro, CA 92630
(714) 951-1231

Continued on page 29.

Hot Line Listing

Name	State	Phone Number	Days	Hours	Topics
Coville Woodburn	NH	(603) 863-5590	M,Tu,Th,F	7-8PM	CT, QU
Ken Johnson	MA	(413) 253-3700	M-Su	6-9PM	BB, PA, MD, WP, MI
Richard F. Malley	CT	(203) 727-0353	M,Tu,W,F	6-9PM	GE, SO, WP, SS, QU, CT, PR
Harry T. Hanson, Ph.D.	NJ	(201) 467-0712	M-F	6-9PM	CO, SS, PR, MD, CT
Edward N. Gooding Sr.	VA	(804) 747-8751	M-Su	6-9PM	CO, SS, PR, MD, CT
Al Johnston	FL	(904) 739-1042	M-F	9AM-6PM	GE
Paul Sanchez	FL	(305) 266-5965	M-SU	10AM-4PM	SS, PR, CT
John & Lisa Beckett	MO	(417) 678-2500	M-F	6-9PM	GE
Jim Ferencak	IL	(312) 599-7505	M-F	10AM-5PM	GE, 3P, DB
Neil Quellhorst	IL	(217) 434-8727	M-Su	7-9PM	AL, BB, GE, PA, SO, TC
David B. Hays	KS	(316) 722-1242	M-F	7-11PM	GE
Art Schumer	ND	(701) 282-7907	M-F	6-10PM	AL, BB, GR, SO, SS, AE
Terri Wiles	CO	(303) 850-7472	M-Su	10AM-6PM	PA
Jeff Fritz	CA	(415) 864-2600	M-F	10AM-6PM	GE, BB, MD, WP, SS, DB, PR
Vincent F. Latona	CA	(818) 703-0330	M-F	9AM-5PM	GE, WP, BB, SS, AE
Carl & Anita Reynolds	CA	(714) 734-9324	M,Tu,F	4PM-9PM	GE
Wayne Hale	CA	(619) 450-3856	M-F	7-11AM	BB, GR, CT
Dennis R. Cohen	CA	(213) 956-8559	Su-Th	10AM-10P-M	GE
Kelly C. McGrew	WA	(206) 943-8533	M-Su	6-10PM	DB, GR, PA, SS, PR, MD, CT

Psychologists on ///
413 W. Heatherglen Ln.
San Dimas, CA 91773
(818) 963-2890

West Covina/Glendora Apple /// User Group
413 W. Heatherglen Ln.
Sam Dimas, CA 91773
(818) 963-2890

Sacramento Apple /// User Group
1433 Elsdon Cir.
Carmichael, CA 95608
(916) 482-6660

FLORIDA

Sarasota Apple /// User Group
c/o Computer Centre
909 So. Tamiami Trail
Nokomis, FL 33555
(813) 484-0421

GEORGIA

Atlanta /// Society
385 Saddle Lake Dr.
Roswell, GA 30076
(404) 992-3130

ILLINOIS

MThird Apple Users Group
P.O. Box 176
Chicago, IL 60690

KANSAS

Kansas City Apple /// User Group
3800 Cambridge
Kansas City, KS 66103
(913) 588-6025

MAINE

Southern Maine Apple Users Group
Casco St.
Freeport, ME 04033
(207) 865-4761

MARYLAND

Apple /// SIG Chairman
Washington Apple Pi
8227 Woodmont Ave. #201
Bethesda, MD 20814
(314) 654-8060

MASSACHUSETTS

Applesauce
24 Dickinson St.
Amherst, MA 01002

NORTH DAKOTA

North Dakota Apple /// User Group
Harwood ND 58042
(701) 282-7907

OHIO

Cincinnati Apple /// User Group
7960 Shelldale Way
Cincinnati, OH 45252

TEXAS

Houston Area Apple Users Group
(Apple /// Division)
P.O. Box 610150
Houston, TX 77063
(713) 688-3102 or 974-5153

VIRGINIA

Charlottesville Apple /// User Group
216 Turkey Ridge Road
Charlottesville, VA 22901

Greater Tidewater Apple /// User Group
Rt. 2, Box 216
Hayes, VA 23072
(804) 642-5655 or 898-3500, ext. 2671

WASHINGTON

Seattle Apple /// User Group
9630 240th Pl. SW
Edmonds, WA 98020
(206) 546-3019

Three Shorts

by Dennis Cohen

Once again, three short graphic programs are presented below. Two are written in **Pascal** and one in **BASIC**. The BASIC program requires that **/BASIC/BGRAF.INV** be online. If you are placing this program under *Selector ///* or *Catalyst*, change line 10 to point to where you have located BGRAF.INV.

All three of the demos use the full 560 x 192 B/W graphics screen for the clearest images possible. If you will be typing in the Pascal programs, make note of the **UNITS** required in each and be sure that they are in your **SYSTEM.LIBRARY** or your program library file.

```
PROGRAM Spirograph;

{ ************************************************************ }
{ *  Spirograph -- by Dennis Cohen (minor mods. by B.C.)     * }
{ *  ----------                                              * }
{ *  Most people have seen spirograph patterns.  One of the  * }
{ *  easiest ways in which to construct such designs is the  * }
{ *  rotation of polygons.  The following program rotates    * }
{ *  four such figures: an equilateral triangle, a square,   * }
{ *  a regular pentagon, and a regular hexagon.  Each figure * }
{ *  will stay on the screen until the user hits any key.    * }
{ ************************************************************ }

USES REALMODES, TRANSCEND, PGRAF;

CONST  radian = 57.29578  { Degrees in a radian} ;
       xcen = 280; ycen = 96;
       size_x = 180; size_y = 90;
       percent = 0.1;

TYPE   arr10 = ARRAY [1..10] OF INTEGER;

VAR    i, m, n: INTEGER;
       x, y: arr10;
       angle, angle1: REAL;
       deltax, deltay: INTEGER;
       ch: CHAR;
       next: BOOLEAN;


FUNCTION keypress: BOOLEAN;

VAR charcount: INTEGER;

BEGIN
  charcount := 0;
  UNITSTATUS (1, charcount, 21);
  keypress := (charcount <> 0)
END; { Of FUNCTION keypress }


BEGIN { Main Program }
  initgrafix;
  grafixmode (BW560, 1);
  viewport (0, 559, 0, 191);
  grafixon;
  FOR i := 4 TO 7 DO
    BEGIN
      next := FALSE;
      fillport;
      angle := 0.0;
      angle1 := 360.0 / (i - 1);
      FOR n := 1 TO i DO
        BEGIN
          x [n] := xcen + ROUND (size_x * COS (angle / radian) );
          y [n] := ycen + ROUND (size_y * SIN (angle / radian) );
          angle := angle + angle1
        END;
      REPEAT
        moveto (x [1], y [1]);
        FOR n := 2 TO i DO
          BEGIN
            m := n - 1;
            lineto (x [n], y [n]);
            deltax :=  x [n] - x [m];
            deltay := y [n] - y [m];
            x [m] := x [m] + ROUND (percent * deltax);
            y [m] := y [m] + ROUND (percent * deltay);
            IF keypress THEN
              BEGIN
                READ (keyboard, ch);
                next := TRUE;
                UNITCLEAR (1)
              END
          END;
        x [i] := x [1];
        y [i] := y [1]
      UNTIL (next = TRUE)
    END;
  texton
END. { Of PROGRAM Spirograph }
```

```
0   REM ************************************************
1   REM *  Surface -- by Dennis Cohen                    *
2   REM *  -------                                       *
3   REM *  This program draws a 3-D surface on your      *
4   REM *  Apple ///'s graphic screen.  The surface      *
5   REM *  looks much like a cowboy hat.  This one       *
6   REM *  can take some time, so RUN the program and    *
7   REM *  take a coffee break!                          *
8   REM ************************************************
10   ON ERR INVOKE"/BASIC/BGRAF.INV"
20   PERFORM initgrafix:OFF ERR
30   PERFORM grafixmode(%2,%1)
40   x%=0:xm%=559:y%=0:ym%=191
50   PERFORM viewport(%x%,%xm%,%y%,%ym%)
60   ON KBD GOTO 340
70   DIM ub%(424),lb%(424)
80   xc%=280:yc%=96:xr%=153:zr%=105:h%=40:w=0.043:za%=93
90   FOR i%=1 TO 424
100     ub%(i%)=0:lb%(i%)=1000
110     NEXT i%
120   PERFORM grafixon
130   PERFORM fillport
140   FOR x%=-xr%+1 TO xr%-1 STEP 5
150     z1%=INT(zr%*SQR(1-(x%*x%)/(xr%*xr%))+0.5)
160     z%=-z1%
170     y%=h%*SIN(w*SQR(x%*x%+z%*z%))
180     x1%=x%+xc%+z%:y1%=INT(191.5-(yc%+y%+z%/2))
190     FOR z%=-z1%+1 TO z1%-1
200       y%=h%*SIN(w*SQR(x%*x%+z%*z%))
210       x2%=xc%+x%+z%:y2%=INT(191.5-(yc%+y%+z%/2))
220       IF y2%)=lb%(x2%-za%) THEN GOTO 260
230       lb%(x2%-za%)=y2%
240       IF ub%(x2%-za%)=0 THEN ub%(x2%-za%)=y2%
250       GOTO 280
260       IF y2%(=ub%(x2%-za%) THEN GOTO 300
270       ub%(x2%-za%)=y2%
280       PERFORM moveto(%x1%,%y1%)
290       PERFORM lineto(%x2%,%y2%)
300       x1%=x2%:y1%=y2%
310       NEXT z%
320     NEXT z%
330   TEXT:HOME:END
340   IF KBD=27 THEN POP:TEXT:HOME:END
350   ON KBD GOTO 340
360   RETURN
```

```
PROGRAM Golfball;

{ ************************************************** }
{ *  Golfball -- by Dennis Cohen                   * }
{ *  --------     (minor mods. by B.C.)            * }
{ *  This program will draw a golfball on your ///! * }
{ *  It is actually just drawing a series of rays  * }
{ *  from one point on the circle to other points  * }
{ *  on the circle that makes the golfball.        * }
{ ************************************************** }

USES REALMODES, TRANSCEND, PGRAF;

CONST scalefactor = 0.299199;
      xcenter = 280;
      ycenter = 96;
      size_x = 180;
      size_y = 90;
      escape =  27;

TYPE  arr21 = ARRAY [1..21] OF INTEGER;

VAR   x, y: arr21;
      theta: REAL;
      i, j, k: INTEGER;
      ch: CHAR;


FUNCTION keypress: BOOLEAN;

VAR charcount: INTEGER;

BEGIN
  charcount := 0;
  UNITSTATUS (1, charcount, 21);
  keypress := (charcount () 0)
END; { Of FUNCTION keypress }


BEGIN { Main Program }
  initgrafix;
  grafixmode (BW560, 1);
  viewport (0, 559, 0, 191);
  fillport;
  grafixon;
  FOR i := 1 TO 21 DO
    BEGIN
      theta := i * scalefactor;
      x [i] := xcenter + TRUNC (size_x * COS (theta));
      y [i] := ycenter + TRUNC (size_y * SIN (theta))
    END;
  FOR i := 1 TO 20 DO
    BEGIN
      j := i + 1;
      FOR k := j TO 21 DO
        BEGIN
          moveto (x [i], y [i]);
          lineto (x [k], y [k]);
          IF keypress THEN
            BEGIN
              READ (keyboard, ch);
              IF ORD (ch) = escape THEN
                BEGIN
                  texton;
                  EXIT (Golfball)
                END;
              UNITCLEAR (1)
            END
        END
    END;
  REPEAT { Nothing! }
  UNTIL keypress;
  UNITCLEAR (1);
  texton
END. { Of PROGRAM Golfball }
```

Classified Ads

256K Apple /// System for sale complete with; Second Disk ///, Silentype ///, UPIC, PFS File & Report, Visicalc ///, SOS Manuals, Apple Writer ///, Business Basic, Pascal, SOS Device Drivers Writers Guide & Service Reference Manual. Total Price: $3100.00 Delivered. Call Larrie Easterly 503-656-7159 Or Compuserv 73125,553

DISK LOG THREE — Tired of waiting for your printer to print out the catalog of all your disks one at a time? Use DISK LOG THREE: Menu driven 5 1/4" floppy disk inventory program for use with two disk drives or Profile. Simple. Insert program in one drive, disks to be inventoried in other. Catalog information is stored by disk. Inventory listing can then be printed out for all disks at one time without the wait. DISK LOG THREE is $15.00 plus $5.00 for postage and handling. Send $20.00 check or money order payable to Leon Group of Bay County 3913 Pisa Drive, Suite I-6, Panama City, Florida 32408. For information send SASE.

Used Apple /// Equipment for sale: Amdek Color][RGB monitor, complete with Apple /// cable. This is a good color monitor but you can only get 8 out of 16 Apple /// colors. Price: $325. Epson MX-100 in perfect condition. This wide carriage printer will print over 230 characters across, enough for a very wide report. Price: $425. Apple DMP in perfect condition. Price: $450. DTC 300/S-KSR Letter Quality Printer. Can be attached to your Apple /// via the RS-232 serial port or any serial card. This can also be used as a stand-alone typewriter and terminal. Price: $600. Apple ///plus, Monitor ///, ProFile 5 MegaByte Hard Disk with Catalyst. Like new. Price: $2850. AppleMouse //e. This mouse and interface card will work directly with Draw ON ///. Perfect condition. Price $150. All prices include shipping. XCOMP 16 MegaByte hard disk. How would you like 30720 blocks of ultra-fast hard disk storage! You can configure this drive as one 16 MegaByte drive or two 8 MegaByte drives. Works with Catalyst, Selector and all other programs that work under SOS. Includes interface card and all documentation. Only $2995! Contact ON THREE (805) 644-3514.

ON THREE can assume no responsibility for any item presented for sale in the *Classified Ad* section. If you have something to sell for the Apple /// and would like to put an ad in the Classified section, send your copy to:

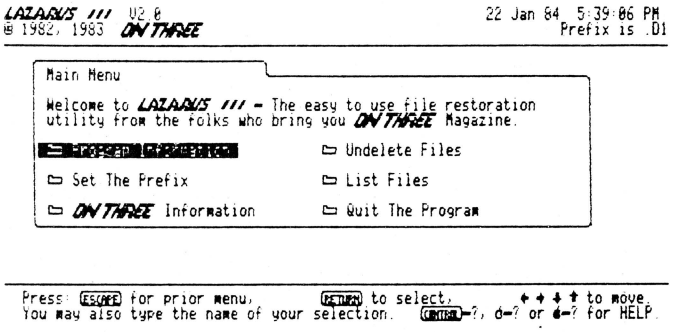
ON THREE
Attn: Classified Ads
P.O. Box 3825
Ventura, CA 93003

Include payment of $1.00 per word of copy. There is a $25 minimum.

Why Don't You Have A Copy Of Lazarus?

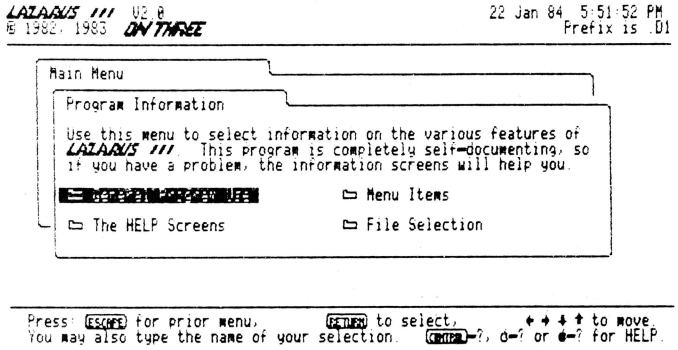
It's been a long day and you're tired. Just a few more sentences and you'll be finished with that long report your boss wants on his desk tomorrow morning. Just as you are about to save it, you press the wrong key and your last 10 hours of work is erased from disk. What do you do? If you have ***Lazarus ///*** from ***ON THREE*** it's simple! Just press a few more keys and voila!, your file is restored to original condition. If you don't have a copy of ***Lazarus***, well - there's always the unemployment office.

Lazarus /// V2.0



If you have ever accidentally erased a file from one of your disks and have wished for a little magic to bring it back, ***Why Don't You Have A Copy Of Lazarus?*** This extremely easy to use program will restore any file that you have accidentally deleted from any of your disks. It will work with *ProFile*, the *Micro-Sci A3, A73 & A143*, and every other disk drive that you can connect to your Apple ///. If you can save a file on it - or delete a file from it, ***Lazarus*** can restore your little accidents. As long as the disk that you are working with is an Apple /// SOS diskette, ***Lazarus*** will work ***Lazarus*** can't restore files from Apple][or CP/M disks.

Lazarus is the very first window based program for the Apple ///. When you select something to do, a window **Pops-Up** and shows you what to do. When you are finished working in that window, your original screen reappears. When a new window (or menu) pops-up, it only takes up a part of your screen-the part below the menu behind it. For example, if you want some information on how to use ***Lazarus***, select the **Program Information** window and it will appear on your screen as shown below.



As you can see, the TAB portion of the Main Menu is still visible, so even after a new window has popped-up you will never get lost and wonder where you are in ***Lazarus***. Just as in most other menu driven programs, pressing **ESCAPE** will bring you back to the next highest window. If you press **ESCAPE** here you will go back to the main ***Lazarus*** menu. All of the windows in ***Lazarus*** work exactly the same way-easy.

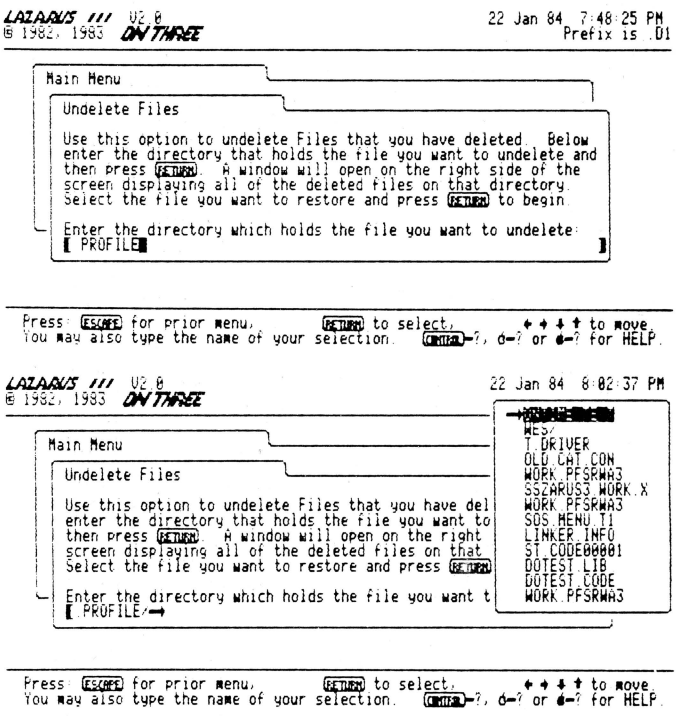
Help - ALL The Time!

The new **Pop-Up** Menus are extremely easy to use but what happens if you are stuck on something and can't remember what key to press? The answer is to simply look at the menu screen. If you will notice the bottom two lines on the screen you will see a quick description of the various keys that you can press. In addition to the keys that control movement, there are three keystrokes that will bring up **HELP** screens. The keystrokes and their effects are listed below:

CONTROL- Lists the keys you can press and tells what each one will do.
OPEN-APPLE- Describes the highlighted item.
CLOSED-APPLE- Describes the screen and its options.

Undeleting Files

To undelete a file, simply tell ***Lazarus*** where the file is located and press **RETURN**.***Lazarus*** will look for deleted files on that disk and show you a window filled with files that you can undelete. Just select which one you want, press **RETURN** and ***Lazarus*** does the rest!



Lazarus is so easy to use, that no matter what experience you have with the Apple /// you will be able to use it. If you are working with an Apple ///, you *need* ***Lazarus***. Almost half of the people who order ***Lazarus*** pay for next day shipping costs because they have to have the program **NOW**. Why don't you order your copy of ***Lazarus*** now - *before* you need it.

To order your copy of ***Lazarus ///***, use the enclosed postage paid envelope or send your check or money order for $49.95 + $2.00 shipping & handling to:

ON THREE
Attn: Order Dept.
P.O. Box 3825
Ventura, CA 93006

* ***Lazarus ///*** requires an Apple /// with 256K or an Apple /// plus.

Disk Of the Month

Do you have the time to type in the programs in each issue of ***ON THREE***? Wouldn't it be great if there was a way to get all the programs without having to type them in? — There is, all you have to do is order the disk!

DOM #1 - Extra Disk Space Plus!

This disk contains all the programs in the ***January*** and ***February-March*** issues of ***ON THREE***. Included are ***Disk Pak1***, which gives you four extra blocks of disk space on all your data disks (a very handy feature for those of you who don't have a hard disk!); ***Disk Pak2***, which lists the files on a directory using Pascal; all of the ***Graphics and Sound Demos*** and much, much more!

DOM #2 - Changing the Characters of Your Printer

This disk contains a program that will do a most amazing thing, it will enable you to change the characters that your ***Apple Dot Matrix*** (or ***ProWriter C.Itoh 8510***) printer prints with. Now your ***DMP*** can print with the same characters that are shown on your text screen. ***Fancy Gothic letters*** and many other fonts are available to use on your printer. Complete documentation makes this program very easy to use. Also included on this disk is a program to list the files on your ***Apple][*** diskettes, all the other programs in the ***April-May*** issue, and many more Graphic demonstrations.

DOM #3 - Changing Your Keyboard

This disk contains all the programs in the ***June-July*** issue of ***ON THREE***. Included is the program that lets you redefine the positions of the keys on your keyboard, all of the ***WPL*** programs, the ***Disk Formatting Utility***, the ***Graphics Sketching*** tool and everything else!

If you would like to SpreadSheet from Basic or even use the program Headfirst to print headings for our program listings, this Disk Of the Month can do it for you!

DOM #4 - Emulation Patch

This disk contains all of the Pascal programs from the ***Volume 2 Number 1*** issue of ***ON THREE***. Included are the ***Apple][Emulation Patch*** program, which lets you use any ***Apple /// Font*** in ***Emulation Mode*** and the new Pascal startup program for ***Access ///*** that lets you ***AutoDial***!

Also on this ***DOM*** is the program and ***UNIT*** that lets you do calculations from within your Pascal programs! Not forgetting the demos, we have also put the ***Radiate Graphics Demo*** and the ***Beatles Music Demo*** on this ***DOM*** . Even a number of pictures created by ***DRAW ON*** are included. These pictures can be viewed with the program on ***DOM #5***. It's also packed with programs and information that you can use!

DOM #5 - Access Draw ON!

This disk contains many of the utility programs in the ***Volume 2***, ***Number 1*** issue of ***ON THREE***. Included are the new Basic startup program for ***Access ///*** that lets you ***AutoDial*** and ***Ben's SUPER Slot Machine***.

All of the **VisiCalc** and **WPL** programs are included as well as the ***Circling Graphics Demo***. It will whow some of the beautiful pictures that can be created with **Draw ON**. Included on **DOM #4** and **DOM #5** are a dozen creations of **Draw ON** for your viewing pleasure.

DOM #6 - Basic Lister Plus!

This disk contains all of the programs in the **Volume 2**, **Number 2** issue of ***ON THREE***. Included are the program that will give you formatted listings of your **Business Basic** programs and the **Pascal** program that will guide you in **Selecting Noises** for animations and game programs.

All of the graphics demos of are included on this **DOM** as well as some nice documentation programs for the **Business Basic Program Lister** and the **Pascal Noisemaker** program. Also included on this **DOM** are some more pictures that **Draw ON** created. We've even thrown in a couple of new fonts for you to use with **Draw ON**. To view the pictures you will need the **Draw ON /// Picture Demo** program on **DOM #5**. If you want to use the fonts you will have to have **Draw ON ///**.

For only $14.95 (plus $2.00 postage and handling) you can get any of these great packages. All Six may be ordered for the extra low price of just $75.00 (plus $4.00 for postage and handling). Mix and match any of the **DOM**'s for the low prices shown below. Order today!

Bulk and group purchase rates are as follows (any mix of DOM's):

2-9 disks: **$12.50** apiece	+ $4 total shipping
10-24 disks: **$12.00** apiece	+ $6 total shipping
over 24 disks: **$11.50** apiece	+ $8 total shipping

ON THREE O'Clock

Calling all you time conscious **Apple ///** owners out there! How would you like a working **clock/calendar** for your **Apple ///**? Just as it was originally intended (and now included in the **Apple /// plus**), this kit comes complete with a plug in clock chip with a battery backup and easy to follow instructions.

With an **ON THREE O'Clock** installed, any time you save or modify a file, the current time and date will be stored on disk. Thus you will now be able to tell which file you last worked on. Your programs can now use the **Apple ///** built-in time and date routines to give you an up to the second read-out of what time it is.

Extremely easy to install and adjust, it is completely compatible with **SOS** and doesn't use up a slot! This is the one you have been waiting for! The package contains comprehensive instructions and a ***Six Month Warranty***! Try and get that deal anywhere else!

Coupled with the **ONTIME** clock driver (see the ad inside this issue), the **ON THREE O'Clock** is an invaluable tool for everyone who has an **Apple ///**. What's the best part? - The price! While others are selling their clock for $60 and up, we have broken the $50 barrier.

For only $49.95 (plus $3.00 for postage and handling) you can get the best little clock in town! See the **Special Offer** for the **ONTIME - ON THREE O'Clock** combination.

2-9 clock sets: **$46.50** apiece	+ $ 6 total shipping
10-24 clock sets: **$43.25** apiece	+ $ 8 total shipping
over 24 clock sets: **$41.00** apiece	+ $10 total shipping

Bulk and group purchases must have one mailing address. Please use the attached envelope for orders. If the envelope is missing, send to:

ON THREE
Attn: ORDER DEPT.
P.O. Box 3825
Ventura, California 93006

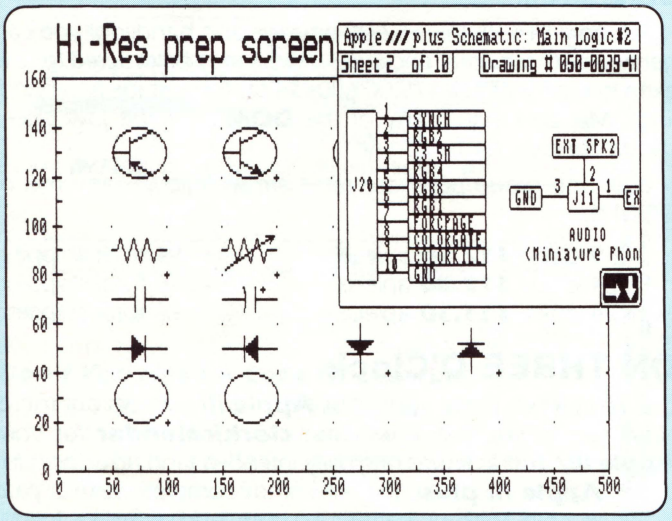
Or call (805) 644-3514 direct for a **Visa**, **MasterCard**, or **American Express** order.

Draw ON /// is a powerful and versatile graphics tool designed exclusively for the Apple /// and Apple /// plus computers. **Draw ON ///** transforms your Apple /// into a combination drafting table, easel and sketch pad. **Draw ON** works in all of the Apple ///'s color and Black/White graphics modes and brings the power of *LisaDraw* and *MacPaint* to your ///.

Draw ON provides powerful cut and paste facilities with the ability to create unlimited libraries or your own special figures and objects (such as circuit components, business logos, animation characters...). You may select these objects and move them onto, between, and around any of the many drawing screens.

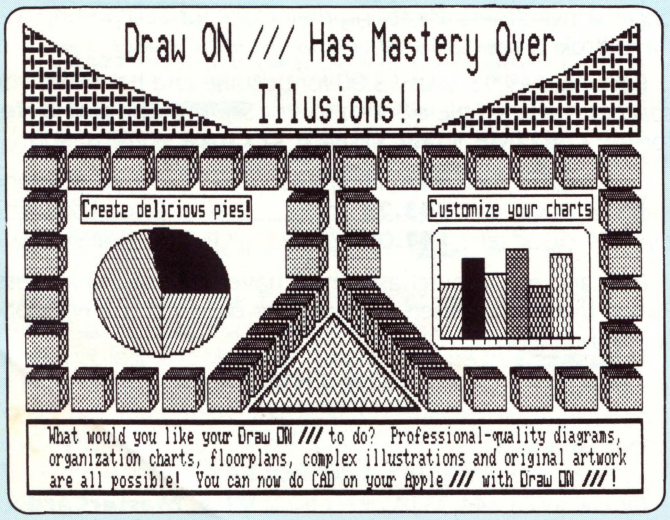
You can also mix text directly with your drawings. A variety of fonts are supplied in sizes up to 14x24. If you don't like any of the typefaces that come with **Draw ON** you can design your own! You can label your drawings with any of these fonts and even use them in your other programs.

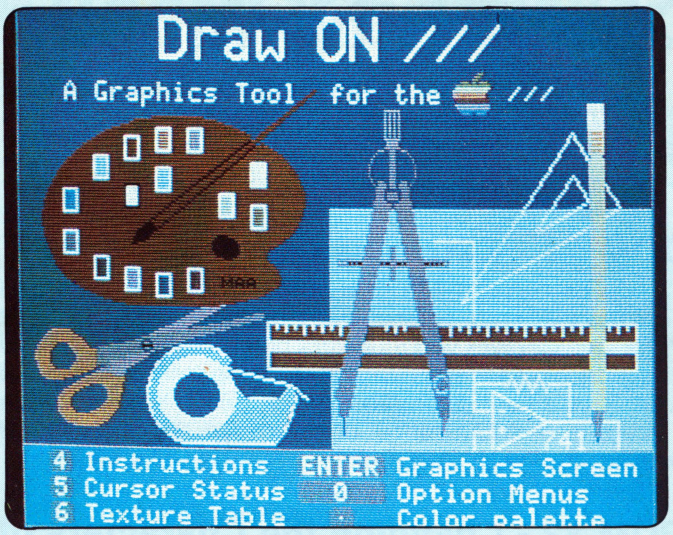
Features such as *rubber banding* of lines, user adjustable grids, built-in help screens and easy to follow menus make **Draw ON ///** the ONLY graphics



The screen that **Draw ON** uses is a window into a much larger document. **Draw ON** allows you to scroll to different parts of the document at will. If you have an *Apple DMP* or *ImageWriter*, **Draw ON** can print your drawings directly. For those of you who don't have either of these printers, **Draw ON** works with all of the printers that work with the **PKASO** or **PKASO U** interface card. This includes *Epson*, *Centronics*, *NEC*, *Okidata*, the *IDS Prism* or *Color Prism* and many others. The Black and White pictures on this page were each designed in less than 30 minutes and printed on an standard Apple Dot-Matrix Printer (DMP). To print out the color pictures you will need a PKASO or PKASO U interface card with an IDS Color Prism.

To control **Draw ON** either a joystick or mouse is used. Since there are no mice available for the Apple ///, ***ON THREE*** has enabled **Draw ON** to use the Apple //e mouse and interface card. If you would like the ease of use that the mouse provides, purchase an Apple //e mouse and follow our instructions for installing it in your Apple ///. **Draw ON** is so versatile, it will work directly with the Apple //e mouse, no modifications are needed for using it in the Apple ///.



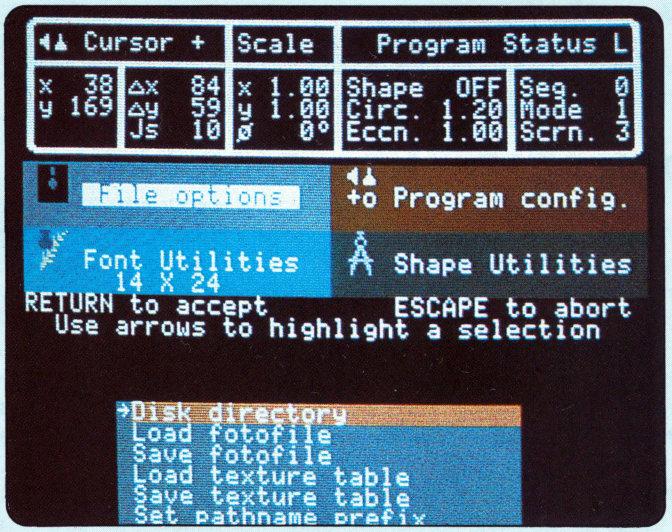


package for the Apple /// that is both powerful and simple to use. Combined with an excellent (Apple style) instruction manual, you can be doing useful work in less than an hour. The only limit to what you can do with **Draw ON** is your imagination.

Draw ON gives an individual the power of a graphic arts studio. Use it in creating charts, preparation of slides and tables for presentations, and letterhead design. With **Draw ON** you can make changes to the dull graphs that your other programs create by adding borders, textures and different typefaces. Even Computer Aided Design applications such as circuit layouts, drafting, and flowcharting are now possible on your Apple /// with **Draw ON ///**.

Objects may be *shrunk* or expanded, rotated, textured and inverted. They may be moved on top of objects already on the screen, behind previously drawn objects, or overlayed along with other figures on the screen. You can also zoom in on a particular portion of the screen to do detailed work.

When working on the color drawing screens you may paint with the Apple ///'s 16 colors and an unlimited number of color-texture combinations. And on all of the drawing screens there is no limit to the types of **Brush** that you can use. Simply pick up an object and draw with it. You can also fill objects with a pattern that you can choose from a menu. If you don't like the patterns that **Draw ON** normally provides, you can change them with the touch of a mouse!



Program Requirements:

Apple /// with 256K or an Apple /// plus
B/W Monitor

Optional Equipment:

Cursor /// Joystick or AppleMouse //e
RGB Color Video Monitor
Dot-Matrix Printer

A PKASO or PKASO U interface card is needed if you don't have an Apple DMP or ImageWriter.

Suggested retail price: $179.

* **Draw ON ///** is not copy-protected and may be installed under Selector /// and Catalyst.

ON THREE
P.O. BOX 3825
VENTURA, CA 93006

BULK RATE
U.S. POSTAGE PAID
VENTURA, CA
PERMIT NO. 90

ADDRESS CORRECTION REQUESTED