DOUBLE ISSUE

Volume 1
Number 4

ON THREE

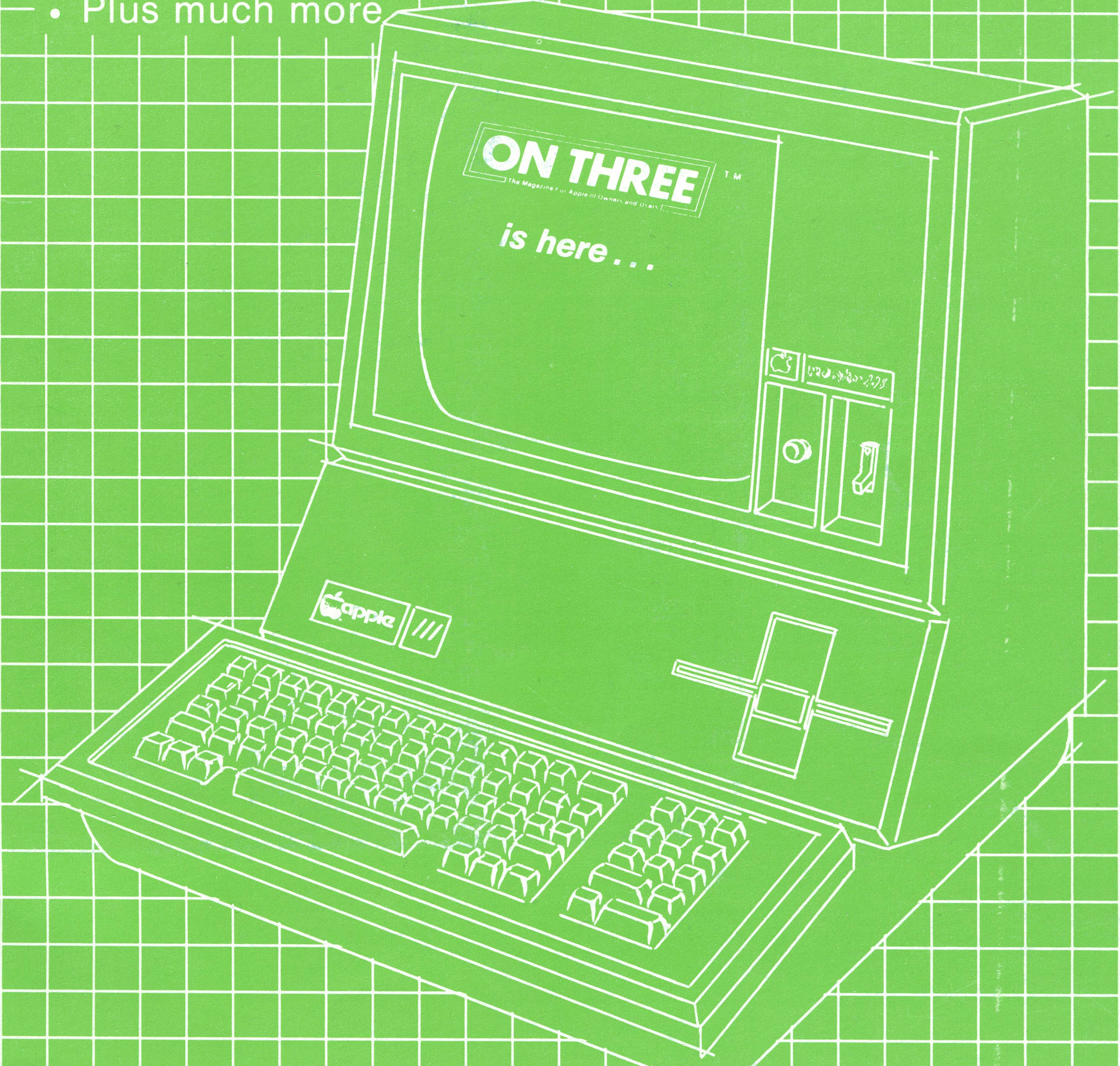
T.M.

The Magazine For Apple III Owners and Users

$3.00

June-July
1983

- Changing Your Keyboard
- Formatting Disks
- PFS: Hints and Shortcuts
- Graphics Sketcher ///
- WPL Revisited
- A Ton of Letters
- Plus much more



ON THREE Presents . . .

μ-SCI

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	$509
Savings	$80	$120	$150

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.

ON THREE

/// TABLE OF CONTENTS ///

/// FEATURES ///

/// DEPARTMENTS ///

Next Month in ON THREE

MLM Utility... Decision Support with Visicalc... PasCalculator... Automating Access ///...tutorials... something for everyone...and more!

June-July, 1983
Volume 1, Number 4

Editor/Publisher:
Bob Consorti

Managing Consultant:
Joseph Consorti

Cover Design and Artwork:
William V. Padula
Cranford, New Jersey

Interior Artwork:
Virginia Carol

Printing Services:
Ojai Printing &
Publishing Company
Ojai, California

ON THREE **- THE** Reference Source for the Apple /// is published somewhat monthly by ***ON THREE,*** P.O. Box 3825, Ventura, California 93006.

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

Subscription information: U.S. - $30 for 12 issues. For first class mail, please remit an extra $10.

All foreign subscriptions should include additional postage in the following amounts:

$8 for Canada and Mexico
$12 for Central America - Caribbean
$16 for South America - Europe - Africa
$19 for Asia - Pacific Islands - Australia

Funds should be remitted in either U.S. dollars drawn on a U.S. bank, International Money Order, or by a direct bank draft.

Group Rates are as follows:

2 - 9 members, $28 per subscription
10-49 members, $25 per subscription
50 + members, $22 per subscription

Group purchases must have one mailing address.

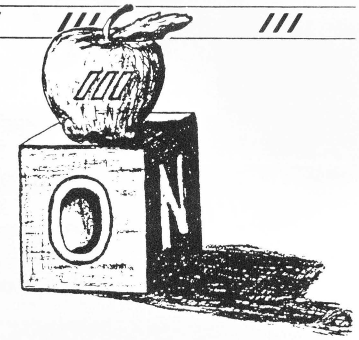
Return postage must accompany all manuscripts, drawings, and diskettes submitted if they are to be returned, and no responsibility can be assumed for unsolicited materials.

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.

Dealer inquiries are welcomed. Please write to the above address for volume pricing and terms.

The Editor's Block

Bob Consorti



Back and better than ever! Keep those article submissions coming in and we'll be able to stay on our current schedule of going to a monthly format starting in October. The past few weeks we have been deluged with articles and new subscription orders.

As many of you know, Apple Computer recently did a mailing to the installed (registered owners only!) Apple /// user base which contained a plug for ON THREE. That turns our to be 18701 users and we have been getting swamped with all the orders, but I love it! Ah, when it rains it pours. For those of you wondering about orders you have placed, we are currently looking at a three-week turn around time because of the volume of mail we have been getting. We are revamping our order system and should get that figure down to around a week or so in the coming months.

The turn-around on getting letters to the editor is somewhat greater, averaging about four weeks. Because of this, if you are having problems with an order, write C/O the ORDER DEPT, because if you address it to me you're in for a little wait.

As I mentioned above we are getting quite a few new article submissions. I really have to thank every one of you who have sent something in, you've made my job an order of magnitude easier.

For those of you who have been considering the purchase of one of the Micro-Sci disk drives, let me try to sway your opinion. By now you've probably read the enthusiastic plug for the drives on the inside front cover. As I said there, they are great! If you are tired of using 35 normal diskettes to back up your ProFile, the A143 is for you! Even if you just feel cramped with those 140K drives and want a higher capacity storage device, how can you beat it!

The price per byte ratio is one of the best in the industry for floppy disk drives. If you buy the A143 you will get a pleasant surprise: It will read a standard diskette! This means that if you don't already have an external disk drive, with the purchase of an A143 you can do copying of normal 140K diskettes. Just put the disk to be copied in the A143 and copy to a disk in the internal drive. Talk about compatibility.

And if you're not in the market for a high density drive, the A3 direct replacement for the Disk /// is a heck of a buy — even at the suggested list price! Just as a note, these drives come complete with very thorough instructions and the A73 & A143 have a diskette with the necessary device drivers for both SOS and CP/M. All three come with a six-month warranty! The A3 works as a standard 140K floppy in emulation mode but the A73 and A143 are not supported under emulation due to problems with the Apple][DOS 3.3 head seek routine — but who's using emulation mode anyway?

Just one more Micro-Sci note and we will go on: the Gameport /// provides so much compatibility with the Apple][that it only uses Apple][paddles or joysticks! It cannot operate with the Apple /// joysticks for two reasons. First of all, the connectors are completely different. Secondly, and most importantly, the potentiometers that the Apple /// normally connects to are way out of range of the Apple][paddle and joystick potentiometers. Thus compatibility with Apple /// products is just not possible.

Switching now to a more complicated subject: So many people have written in saying that they think Apple should be doing a better job selling and supporting the /// that I think it deserves a good response. The Apple /// group is part of the Personal Computer Systems Division of Apple. PCS also holds the Apple][(and now //e) group. Since the //e is selling approximately the same number of units in a month that have been sold in total for the ///, there may be some resource allocation problems.

I really can't blame Apple if it gives priority spending, advertising and the like to the //e, after all they do have to go with their bread and butter products. However, that doesn't help our situation any. What will? You tell me! Possibly a revamping of thought concerning the Apple /// product is needed. It is such a fine product that I sincerely hope that it starts to be marketed as just that.

Quite a number of people have also written saying that if the /// didn't get off to such a horrible start, the IBM PC — well, there wouldn't be an IBM PC. Or if there was, it would be selling in very low numbers.

Does the IBM allow you to upgrade to a hard disk or high density floppy and use all your old software without modification? No — but the /// does. Just as with the Apple][, to change disk drives on a software package usually means sending a large check to the software publisher for a copy of the package that will work with brand 'XYZ' disk drive. That's a horrid situation on every system except the ///!

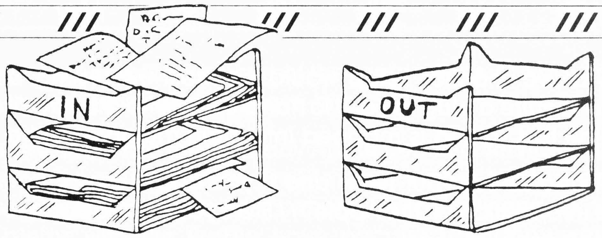
When the /// starts to be marketed for what it is — the BEST micro on the market today — sales will surpass even the Apple //e's torrid pace. When can we expect this? Soon I hope! Who knows, John Sculley, the new Apple CEO has no doubt brought some fresh ideas into the system. Maybe we will see the type of change that we need.

Enough of that, let's now look inside the latest issue of ON THREE. If you thought we answered a lot of questions in the last issue, take a look at the letters to the editor this time! Twice as big as last time, with twice the amount of information, this is something that every reader should go through. Almost every conceivable problem is presented and answered (well, almost all of them).

This month Al Evans brings us a very interesting utility program. This one allows you to redefine the position of the keys on your keyboard! Visicalc users can now put the '+' key in a more accessible place. Pascal programmers can remove the 'CONTROL-Back Slash' so that users can't break out of programs. Move keys around, change what you like — it's all now user definable!

The Editor's Block Continued on Page 15

Ask THREE: (Letters to the Editor)



Dear Mr. Consorti,

We at Great Divide Software would like to thank you for your review of our CRITICAL PATH SCHEDULING SYSTEM in the April-May, 1983 issue of ON THREE. It was certainly a fair and in-depth review of this product.

Your comments, both positive and negative, were greatly appreciated. We are taking steps to fix the few negative comments that you made including the following:

1. We are changing the Option 0 QUIT to allow the user a chance not to exit the program if he has inadvertantly selected this Option or has changed his mind.

2. An addition has been made to the manual which cross references formatted information and adds a listing to the Table of Contents.

3. The two formatted reports that are mentioned on pages 34 and 35 of the manual were inadvertantly left off the dealer demo package that you received, but are now there. They have and still are on the normal end user package.

Again thank you for your comments and keep up the good work in ON THREE.

Sincerely,

Craig A. Mills
President
Great Divide Software, Inc.

Dear Bob,

Enclosed is my check and order form for disk #2. Keep up the good work.

There are a number of us Apple /// users active in the UCSD Pascal User's group (USUS) and we communicate via the MUSUS SIG on CompuServe, new members welcome.

My ON THREE O'Clock works great. It took me an hour to install — including a lot of time spent in being extra careful because I was afraid I would break one of those little metal pins on the motherboard, or disconnect a wire somewhere.

I still hope to write an article on managing viewports on the ///, but till then, here is one tip for using the Pascal editor as a word-processor. There is probably a better way to do this, but this one works for now.

One drawback I noted was that there did not seem to be a convenient way to put a page break from the Pascal editor. One day (about ten months later) it occurred to me that I could copy in a file with a page break character in it. This works fine. The program below creates such a file, titled "page.text".

```
PROGRAM MkPage;
VAR Outfile: TEXT;
BEGIN
  REWRITE (Outfile, 'page.text');
  WRITE (Outfile, 'pg', CHR (12);
  CLOSE (Outfile, LOCK);
END. (Of Program MkPage)
```

The "CHR (12)" is for the Epson MX-100 printer. Others may differ. Also on the MX-100, turning dip switch 2-4 to 'off' will cause an automatic 1 inch skip at the end of each page. This is great for word-processing and most other uses if you stick to one size paper.

I have some product suggestions.

The Pkaso card sounds great. Unfortunately it still will not do what I would like. I want to be able to print any screen to my Epson printer.

I have set my Pascal programs up to print any text screen if I type ⟨Open Apple P⟩ at any input prompt. With the Pkaso board I will be able to print any screen, including graphics. The problem is that in preparing user documentation I often want to print screens created by other systems, eg. the System Utilities program, or a run-time error message from the Pascal operating system.

Given that Pkaso has the capacity to dump any screen — I want a ⟨PRINT SCREEN⟩ function button — independent of anything else — that will do just that. I realize this may be impossible — but then again...

The second idea may require the use of a multi-task environment such as MODULA)(seems to have. I would like to have my computer control two independent monitors. I know that it's possible to split the screen into two view ports and control each independently — or to toggle between two or more screens, but that lacks the real versatility of a two screen system. Given that a monitor only costs about $100, such a hookup could add a lot of power for relatively low cost.

Imagine scrolling through a program on one screen while holding the part being edited on the other, or a help screen system on one side while the user was entering data on the other.

I am not burdened by knowing how easy or hard these things would be to do from a hardware point of view. As a user I see that they would be very valuable.
Thanks again for starting a much needed support system for Apple /// users.

Sincerely,

Harry Baya

Dear Mr. Baya,

Thank you very much for your last letter. I'm always glad to hear of Apple /// SIG's wherever they may be.

On your note about printing screens, ON THREE will employ the convention of having its applications programs print-out the text screen when ⟨Open Apple P⟩ is pressed. Hopefully this will become a standard.

Such a two monitor setup would be most interesting but it would require a plug in card to handle the extra display due to how the text screen is mapped into main memory on the ///.

Thanks again for your letter and if you could give us a little more information on the Apple /// SIG's you know about we will promptly publish them.

Dear Bob,

I would appreciate your comments on the following:

1: The Toaster by XCOMP, which provides 5 megabyte minicartridges, two at a time.

2: Why the Apple /// Softcard system requires so many patches to run the diskettes which do not require patching on the Apple)(CP/M system It seems like this is extremely poor planning. Are there plans for an update to the CP/M system for the /// which will be fully compatible with the Apple)(CP/M system?

3: Does anyone know of any software which will permit the printing of custom fonts on the Epson MX-80 with the UPIC card? I have been able to use the fonts (after reversing them with an assembly routine I wrote) on the graphics screen and then dumping the graphics screen with Alpine Computing's PSCREEN, but in 80 column mode, the printing looks "compressed, and subscripted".

4: Why is my dealer having so much trouble laying his hands on the new "SOS Reference Manual" and "Device Driver Writer's Guide"? (All $75 worth) Do you suppose that ON THREE could offer these at a discount?

5: Has anyone been able to adapt the Apple /// to communicate with TDD Baudot equipment?

Keep up the great work!

Yours Truly,

Stephen M. Dorman
Washington

Dear Mr. Dorman,

I haven't had a chance to view the Toaster yet, however it does sound very nice. I am going to try to get a review on it in the magazine very shortly.

NOTE: Since I wrote the reply to this letter I found that the Toaster was having some reliability problems that have just been cleared up. It is under final testing now and should be available within a couple of months.

The Apple /// Softcard system is somewhat of a problem. It is mainly due to the way that Apple /// SOS drivers are handled under CP/M. I haven't heard of any updates in the system to fix things up but we will keep all our readers informed on this.

I haven't heard of the custom font software for the Epson & UPIC card, hopefully one of our readers will respond on this subject.

I can't say what the problems are involved in the delay of those manuals but I do think things will clear up in the near future.

NOTE: Again, I just found out that Apple only initially printed 750 copies of the manuals. They have just completed a second printing and the manuals should be available within a very short time (Probably by the time you read this).

I am not familiar with the equipment you mentioned and I hope that one of our other readers can help.

Dear Robert;

I just bought the Gameport from my local dealer and installed it in slot one. Then I found out that the connectors did not agree with the paddle connectors.

My paddle is the one made by TG Products. The connector is very different from the Gameport socket and it plugs into the back of the /// but not on the Gameport. With all the descriptions as being so compatible with the Apple /// Computer, I expected a compatible connector.

Please advise how the connector is meant to be used with the socket on the Gameport. Is there an adapter available? I would prefer to purchase one.

I hope that I will be able to use the 'board and stick'. Nothing is able to 'move' yet. By the way, will this move the cursor when I am using Visicalc?

The review of Pkaso /// (and all the others) in the last issue are very helpful to the newcomer, Thank you! But one point (to me anyway) is missing...

What is the advantage of a serial connection over parallel? Or reversed, is there any difference? It seems to me that I have read somewhere that a modem has to be used as a serial connected device to have the data available in a line one bit behind the next one to go out on the telephone lines. Right? I do not have a modem yet.

The ability of the Pkaso board and disk is very enticing, but I do not know what would have to be done to my Diable 630 and my Epson MX-100 to connect them both to my Apple ///.

Are they capable of other features that I am not aware of? Must the other features, the special print and pictures be presented in a parallel version only?

Thank you for the WPL series... It does work, after all! My dealer was not able to help me with WPL and said so! How about that!

Very best wishes,

Robert Scattergood
Florida

Dear Mr. Scattergood,

On the subject of the Gameport connectors, it is apparent to me that you bought the TG paddle that plugs into the back of the ///. The paddle connectors for the Apple)(and Apple /// are different and therefore can't be interchanged.

Our ad for the Gameport clearly states that it works with Apple)(game paddles. The "compatibility" is with Apple)(paddles, joystick and all software. Since the pin signals are fairly different it will be hard to build an adapter. I'd suggest bringing it back to the dealer who sold you the Gameport and paddles and demand an exchange as Apple /// paddles cannot be connected to a Gameport (he should have known at least that much).

The paddle will not move the cursor in Visicalc. Quite a few patches would have to be made in the CONSOLE driver for that to happen. Since Apple will not release the source code for the

CONSOLE driver (proprietary information) this is not likely to happen in the near future.

There is no real advantage of serial over parallel except that a parallel interface can transmit data at a much greater rate. This, of course is true only if the device receiving the data can take it that fast. The Apple /// has a built in serial interface that can be connected to a variety of printers and communication devices.

Depending on your configuration you may need a parallel and another serial interface. The Pkaso board is a parallel interface that will directly connect to the MX-100 and provide full text and graphics output. However, the special features can be sent to the printer in either form of interface.

If you don't want to purchase the Pkaso board you can buy the Alpine printer driver that can be used with the MX-100 to also provide full text and graphics capability.

I'm glad that the WPL series seems to be helping, it makes all the work I'm doing worthwhile! I hope that this has helped a bit, remember we're here to help you so write us if you have a problem.

Dear Mr. Consorti:

In the April-May issue you published a letter from a reader regarding interface problems between the Apple /// and the Apple Dot Matrix Printer using the Grappler interface card.

The problem that the Apple /// can not be booted unless the printer is "ON" and "Selected" is correct, however the problem is not with the Apple /// nor with the printer. The problem lies with a bug in the printer driver supplied with the Grappler interface card.

I have contacted Orange Micro, the makers of the Grappler and they have resolved this problem and corrected their driver so that any printer attached to the Grappler need not be turned on for the system to boot.

Your readers with the Grappler interface problems can probably obtain a copy of the corrected printer driver by calling Orange Micro at (714) 779-2772. Orange Micro, in particular Bob Mickey, has been very helpful and cooperative in solving this problem.

Regarding your review on the ProFile hard disk, you stated that not one single ProFile has been returned due to hardware failure. Only a naive person can believe that. While it is true that the ProFile has a very low failure rate, like any other mechanical device, some have failed and to lead anyone to believe otherwise doesn't do justice to the credibility of your publication.

Sincerely,

Richard Ferrandiz
California

Dear Mr. Ferrandiz,

Thank you for your comments, I'm sure our readers will appreciate your input on the Grappler — Apple DMP problem.

If you look back over the article on the ProFile you will see that I said "I understand that to date not one has been returned to Apple due to a hardware failure.". The key word here is "Apple". A few short months ago one of the senior techs. at Apple gave me that number and I believe him.

You see, there are many, many Apple certified Service Centers around the country. When ProFile (or any other Apple product) has a serious problem this is where they usually end up. Some problems with the ProFile have occurred but they were all handled by the Service Centers through means of simple repair or major head disk assembly replacement.

I didn't intend to mislead anyone with that statement and I'm sorry you think we lost some credibility with it. Our only purpose was to show that the ProFile is extremely reliable and that any problems that do occur are ALWAYS dealt with locally.

Dear Bob,

In the April-May issue you said the only way to upgrade from 128K to 256K "is the Apple way — high price". (page 6) I have received quotes ranging from $650 to $950 for this service. Would you, or a reader, care to comment on a "best price"? And where it can be obtained?

Now a technical question. My machine crashes when the reading on a thermometer taped to the front panel next to the disk drive door gets up to 86 degrees F. I now use a fan to keep this reading below 80. An old radio/tv repair trick to find intermittent component failure of this sort is to cool individual components selectively with a spray coolant like freon. Can this be done with chips or are they too sensitive to gentle cooling? Any other suggestions?

How about some info. on surge supressors. Beats me why a machine which cost $5000 (a pre-update model) needs a $50 device to plug it in safely.

I, too, am looking for a way to operate my NEC Spinwriter in the Emulation Mode faster than 300 baud.

Last but not least, a kudo for Apple support. The RS232.DRIVER file which operates my NEC Spinwriter 3510 for Visicalc /// and Business Basic would not function for Quickfile ///. A telephone call from "John" to reconfigure the device type to $41 solved the problem.

Sincerely,

James D. Tovey, M.D.
Idaho

Dear Dr. Tovey,

The prices on the memory upgrade are just crazy!! There is only one way around it and you may not fall into the category. It all depends on what memory board you have.

If you take the bottom off of your Apple /// you will see the memory board riding piggy-back on top of the motherboard. If you see two rows of chips you have the new 5 volt memory board and everything is ok. For a 128K machine only one of these rows are filled with memory chips, while a 256K machine has both rows filled. However, if you see three rows of memory chips (all filled) you have the old 12 volt 128K board and you will have to go the high priced way.

If you are lucky and have the new 5 volt board all you have to do is get 16 of the 64K ram chips (part number 4164, 4864 — most any 64K chip will do just fine!) and plug them into the

empty chip sockets on the memory board. Remember to align them in the same direction as the other chips when you put them in.

Voila! An almost instant transformation to a 256K system. To make sure that the chips are all right, use the Confidence Diskette (ask your dealer for a copy if you don't have one) to test the new memory and you will be set. All your programs can now take advantage of the expanded memory space.

On the subject of the heat problems, it sure does sound like you have a heat related failure. The Apple /// Service Manual which I have says that the proper way to test for heat failure is by a heat gun. Since this process of checking different areas of the motherboard is both complicated and time consuming, I'd suggest letting your dealer look into this problem.

For a long time I didn't use a surge suppressor, then one of my Apple ///'s (the one at home) ProFile had a very bad head crash because someone turned on the vacuum during a write cycle. Those few little devices have now safely protected tens of thousands of dollars of equipment from damage.

I should hopefully have an answer on the NEC problem in a very short time. When we get it I will pass it along to you.

Hooray for Apple! I'm glad they could help you.

Dear Mr. Consorti,

Thanks very much for your reply. Getting back to the small controversy regarding returned ProFiles, I regret that someone at Apple told you that no ProFiles had been returned to Apple.

I work at Apple in Irvine and even though I am not in the Service Department, I know for a fact (my own eyes) that many a ProFile has been returned to "Apple" as being defective, my own included (twice). However the return rate is still extremely low and the ProFile is indeed a very reliable machine.

For a while the ProFile had severe head failures, but whatever the problem was it has now been corrected. Apple Regional Service Centers (Level 2) no longer repair ProFiles, if they are returned defective, the entire unit is replaced, and except for module swapping at the dealer level (Level 1) no "repairs" are made to the ProFile. It is no longer included in the Level 1 service training given to dealers.

I am sure that your intent was not to mislead anyone, however your source was incorrect in making such a statement. If you ever have time please come visit the Regional Support Center in Irvine and accept this as an "un-official" invitation, just call me and let me know when you wish to go.

It is not a huge facility, but you may want to talk to some of our technical support people — both hardware and software, and yes, you will get to see some returned ProFiles also.

Thanks very much for your fine magazine and the support of the Apple ///.

Sincerely,

"Apple Source"

Dear Source,

At the risk of repeating myself, I didn't intend to mislead anyone with that article and I'm sorry if it has confused anyone. Thanks for the letter and I look forward to getting enough free time in the near future to visit you.

Dear Mr. Consorti,

Your April-May issue of ON THREE carried my letter in which I mentioned some problems that I had encountered in hopes of getting some help from your readers. One of the problems had to do with getting good transmission at 1200 baud between the Apple /// and a NEC Spinwriter 7720 printer, in particular in the Apple)(Emulation Mode. One of your readers (me!) did solve the printing problems and I thought you might like to hear how it was done.

Apparently this model Spinwriter needs the configuration that I will describe and I do not know whether other Spinwriters, or for that matter other printers, will respond in the same way.

The problem had to be attacked on several fronts for a successful conclusion.

First — Cabling: When operating in Apple /// (native) mode, there are drivers that establish the communication link between the computer and printer. However, when the computer is in Emulation Mode, there is no driver involvement in the sense that "handshaking" between the computer and printer (at least with this printer) does not occur. The net result is that transmission speeds greater than the printing speed will cause the printer buffer to overflow. The NEC does provide for a signal from the printer to control this, but a new cable, i.e. "modem eliminator", is required. This cable should be set up as follows:

Apple	NEC
1	1
2	3
3	2
4-5	8
6	19
7	7
8	4-5
20	6

It is pin #19 on the NEC side that carries the all-important signal. It is my understanding that in Emulation Mode, pin #5 on the Apple side is NOT monitored, so printer signals should not be sent to this pin.

Second — Transmission Protocol: If transmission rates are to be greater than about 300 baud, the ETX/ACK protocol is recommended to minimize, if not eliminate, dropped characters during transmission.

Third — Driver Configuration: The printer driver was configured as follows:

Byte:	00	01	02	03	04	05	06	07	08	0A	0B
Value:	08	22	05	00	00	40	03	06	00	80	00

The various entries are explained in the Standard Device Drivers Manual. The third parameter (byte #02) was given a value of 05 in order to insert a 5 character delay after a carriage return to ensure complete transmission of all data. The eleventh parameter value of 08 provides a 128 character block size per transmission.

By incorporating these three modifications, my Apple /// successfully transmits information of all kinds to the NEC printer at 1200 baud, even in Emulation Mode. All characters are printed and the printer head always returns to its starting position.

Thank you for publishing my letter. My other problem with Apple /// Business Graphics has not been resolved yet, and if it ever does get corrected, I will let you know. My reason for sending this letter is to pass on my findings to someone else that may be facing the same dilemma.

The Conn./Westchester Computer Society has just started an Apple /// Special Interest Group and at the moment we have 23 members. Anyone in the So. Conn-Westchester, N.Y. area can contact me for information.

Sincerely,

John Lomartire
24 Burr School Road

Westport, Conn. 06880

Dear Mr. Lomartire,

I'm glad that you solved your problem and I'm sure our readers enjoyed hearing how it was done.

Thanks for the information on the Apple /// SIG in your area. Hopefully quite a few of our readers will contact you.

Dear Mr. Consorti:

At last, a ray of light at the end of the tunnel. After ten months with an Apple ///, I was beginning to think I had purchased the world's only one and that there would be no more sold. There has been plenty to complain about with this machine, but it all revolves around a lack of information. Your exellent magazine (I now am reading my second issue) has filled that void. So I can get on with using and experimenting with this wonderful piece of hardware.

I have several comments about items in the April-May issue. Having purchased your ON THREE O'Clock, I can say that it is the first piece of hardware or software with complete documentation. Everything else purchased necessitated at least one call to the dealer or manufacturer. It sure works great with the ProFile hard disk and Backup ///. The article on Backup /// was excellent, but I have one comment. You must have been lucky not to have lost data from your ProFile. In the six months I have owned mine, I have a damaged directory (whatever that means) and a head crash. Fortunately, that occurred during the warrantee period. Unfortunately, we lost a 450 name data base we were just finishing prior to backing up.

Mr. Lomartire's letter on connecting a NEC Spinwriter to an Apple /// was of great interest because I have just fixed my system so my 3510 can print Apple)(programs. I had tried six dealers with no luck and my long distance call to Apple gave me the suggestion to "change the SOS.DRIVER file on the emulation disk" (still looking for it). Even the long distance number didn't work, Apple didn't know. A friend, Brian White, and I now have working Spinwriters with Apple ///'s. The secret is to use hardware-handshake. We used the following cable connections:

	Apple ///	NEC 3510
pins	1	1
	2	3
	7	7
	8	19
	20	20, 5, 6, 8

The printer configuration block was set to:

Byte:	00	01	02	03	04	05	06	07	08	09	0A	0B
Value:	08	22	00	00	00	00	03	06	00	00	FF	80

(Note: the values above are for the .RS232 but the .PRINTER driver uses the same values as far as you can enter.)

On the printer, we used a 1200 baud, parity check off, full duplex. The other setting will be dependent upon what your software needs.

By the way, you print Apple)(programs by PR#5, not PR#1 as was suggested to one of your readers.

On the subject of Apple /// Business Graphics, I have it and it works just fine except for a one character offset on the vertical axis when the numbers are only one digit. This seems to clear up when printing at 10 characters per inch (CPI). The 15 CPI definitely does not work. The program does work with some letter quality printers but requires a software modification first. My dealer had the package so configured by software for a nominal fee. I believe the package is called PIK.

I was convinced that there was no software for the Apple /// and voiced my displeasure loudly to several dealers. One finally gave me a copy of the 3rd Wave which is a listing (fairly thick) of Apple /// software. It has been very useful, with only minor innacuracies. Any Apple /// owner who doesn't have one should ask his dealer for a copy.

Something you might wish to consider is a section on "quickies" or little hints that people submit. They would not be long enough for an article but offer some little help to others. I offer the first "quickie" for your magazine: To use the printer control characters in Apple Writer ///, use the glossary instead of Control V, printer character, Control V. I simply build a glossary of printer commands which I load after booting Apple Writer ///. For example, my Spinwriter uses Escape semi-colon for a half-line up. I just enter glossary definition (G)?, then type an up arrow (naturally) escape semi-colon. Then to use it, I type (G) up arrow and the escape semi-colon is printed on the screen. When printed on the Spinwriter, the paper goes up a half a line at that point. I have a 3 x 5 card with the most used printer commands in front of me when using Apple Writer. Sure works fine.

In closing, I wish to thank you for your honest evaluations of software and hardware. Before ON THREE, we had to "read between the lines of glowing verbage" to tell what were the program's faults.

Sincerely,

Donald Glenn
Nebraska

Dear Mr. Glenn,

Thank you very much for your letter. I hope that we can continue supplying Apple /// users with the kind of information that they want and need.

I'm glad that your clock works well and you didn't have any problems. I only wish that all of our readers who purchased the clock kit have as few problems. By the numbers we are getting,

about 1/3 of 1% of the Apple ///'s out there do not work with the clock kit.

This is due to the fact that Apple doesn't test out the clock I/O section of the motherboard during the final test of the manufacturing process. Thus some bad motherboards slip through the last test and reach the end user.

In reponse to the ProFile article, I haven't lost any data but I have had some problems. Yet these were quickly fixed at an Apple certified Service Center. The problem you had with the damaged directory is (more than likely) because of a bug in one of the old versions of SOS. If you use the new version (1.3) of SOS and the ProFile driver things will work out right.

Dear Friends:

Enclosed is $30, please begin my subscription with the first issue. I bought two of your clocks and they are great. We use the computers for business purposes and word processing and run a few educational games on them in emulation mode.

My favorite programs are from Quark. They know the /// system and how to make useful programs for it. Word Juggler is so superior to Apple Writer ///, that there is no comparison. I am disappointed with the support from Apple . I keep hearing things like Applenet or the ///e computer, but they seem to be mostly indifferent to little things also, like Quickfile /// has a bug in its event fence that prevents reports from being spooled on the Discourse. When I brought this to the attention of their technical department, they told me they had no plans to correct it, but did promise to send an earlier unreleased version with limited memory. They never did.

I appreciate your efforts, and keep up the good work.

Respectfully,

Cecil N. Clark, Ph.D.

Dear Dr. Clark,

Hold on just a little while longer because Apple will clear up these problems, it just takes time.

Bob,

Just received my first issue of ON THREE (February-March). My hat's off to you for producing a fine magazine. If future editions are anything like this one, you have a real winner. I plan to show several of the local computer stores with which I do business that there is indeed a fan club of the much maligned Apple ///.

I am considering starting a consulting business specializing only in the Apple /// and //e, and have been doing a very extensive survey of several dealers as to how many ///'s they have sold and to whom. Needless to say the)(and //e have the lion's share of the market. In asking why they sell so many more two's than three's, two facts loom prominent. Lack of support from Apple Computer and a basic lack of understanding on the part of many sales personnel as to the true power of the ///. Perhaps through efforts of yourselves, Apple Computer will wake up and show some support for one of their own. This in turn might encourage the dealers to get behind the product.

I am a RF engineer with the Military Electronics Division of TRW Systems Group here in San Diego, and am responsible for the procurement, upkeep and training that goes along with the effective useage of any computer system. We purchased our first Apple /// a little over one year ago and since that time we have added three more to our group. Our manufacturing side of the house just purchased one full up /// system complete with a ProFile and are looking to add another one already. Several other sections are taking note of our success and are considering the /// over the IBM PC. Our only problems have been lack of software tailored to circuit design but with Access /// and a Hayes Smartmodem II, we are able to access our Tyme Share Computer system at our primary facility in Redondo Beach and use its circuit analysis and optimation routines.

Our primary useage of the systems presently is as a word processor using Apple Writer ///. For the most part, Apple Writer seems to fulfill most of our needs. The major problem I hear about is not being able to see on the screen what the finished document will look like. Horizontal scrolling would be nice since our printer is capable of printing much wider than 80 columns, as would an indication of page breaks. If you take the time to really learn how to use the programs features it can do some pretty amazing things. The documentation on WPL is a joke. My //e has a fantastic 207 page manual for Apple Writer and a separate 147 page manual for WPL. The manual for the /// looks more like a first cut and they just never bothered to finish it. As far as I can determine, the two versions of Apple Writer (//e and ///) are virtually identical in structure and operation. I have heard rumors that a new version of Apple Writer /// is due to be released, but then I heard that about the Spelling Checker too. The ///'s Apple Writer Utilities disk makes quick work of converting Apple)(and /// text files in either direction. This same utility allows you to convert Visicalc files also, bearing in mind the memory limitation when going from the /// to the)(. We also make extensive use of Catalyst, Quick File, Business Basic, Pascal, Advanced Visicalc, Versa Form, and Business Graphics. In regards to a letter in your February issue asking about drivers for Business Graphics, any dealer that handles the /// should have access to a package called PIK that is specific to creating drivers to be used with Business Graphics. PIK supports several popular printers and also the Houston Instrument DMP series as well as Hewlett Packard X—Y plotters. We have standardized on the ANADEX DP9501A dot matrix printer driven by the UPIC card. This printer comes standard with both serial and parallel interface ports and we have found them to be very fast, quiet, and highly reliable. They are a bit bulky however and benefit from a sturdy printer stand that allows the paper to feed up through a slot in the bottom of the printer. The interconnect cable is non-standard compared to other Centronics style interfaces. I will be happy to supply the instructions for modifying an Apple DMP to UPIC printer cable. We elected to use the UPIC card to free the RS-232 port for use with a modem and a DMP-7 plotter. A side benefit is being able to print "on line" while using the modem and specifying the printer as the recording file. In addition, the Anadex provides four font sizes, 10, 12, 15, and 16.7 CPI with a 12 inch wide carriage. We also have the choice of 6 or 8 lines per inch, automatic skipover of page perforations, software selected or manually, via dip switches, and a full individually selected dot high res graphics. While not letter quality, the type is very readable and has a crisp appearance. Using 11 x 8.5 inch paper and 16.7 CPI, we are able to get about 183 characters across a page which is a real plus when creating large Visicalc or Quick File reports. For business users, I can never understand the rationale for purchasing one of the inexpensive printers that are a pain to use, emit ear piercing noises, and require special interface cards and all sorts of special tricks to utilize their features. Your printed output is usually the most important reason for using your computer in the first place. I have found that the relative printing speed is much more related to how fast the printer can traverse the print head and advance the paper. The character per second rating loses its meaning if it takes 1/2 second or more to execute a line feed and advance the paper.

Would like to hear from anyone who has a useable screen dump routine for native mode graphics generated in Basic or Pascal and also a CAD system such as CAD-APPLE on the)(from TW Systems that runs in /// native mode and has the provision for creating block diagrams and schematics. Mixing test and graphics is mandatory as is output to a printer or plotter. Looking forward to the next issue and hope to be able to contribute information that may be of interest to other /// users.

Yours,

Wayne Hale
California

Dear Mr. Hale,

Thank you for your letter and thank you for your help in spreading the word of ON THREE's existence.

There are a number of systems to print out text and graphic screens from the native mode Apple ///. The Pkaso system and the Alpine printer driver does this but I don't know of any that work with the ANADEX. Maybe one of our readers knows of one.

Apple has a development system and it may be what you're looking for. It is currently only available to qualified developers but with your situation you could probably get a copy. Contact the people at Cupertino for more information.

Dear Mr. Consorti;

I have just seen the most recent edition of ON THREE, and the enclosed $30 check for a subscription represents my reaction. Keep up the good work!

Dr. Dorman asked about screen dumps for the Epson, and doesn't want to invest in a new card (and I can't blame him). I have been using the Alpine Computing's "Epson Screen Printer" which is an invokable module useable from Basic.It works nicely, and costs considerably less than a new card.

There is another problem with the MX-80 and Apple Writer /// which several people in our institution have reported — using underlining and superscripts which require sending the null character out of Apple Writer ///. Apparently, Alpine Computing has produced a new driver which gets around this problem, but not when using the (not-so) Universal Parallel Interface Card. The solution for UPIC owners has been a Basic program which goes through an Apple Writer /// created file and converts some other symbol (I use Control-Z) into ACSII 0, and sends the file to the printer. This is a slow and cumbersome process. Can you offer any other solutions?

Sincerely,

Jay A. Zimmerman
New York

Dear Mr. Zimmerman,

Thanks for your letter, and the input. While I don't know of any other way to get the null character out of Apple Writer, maybe one of our readers does.

Gentlemen:

Enclosed is a check of $30 for a subscription.

It is hoped that your obligation can be a source and a dialogue means for 'somewhat green' Apple /// users/programmers. Literature on the /// is scarce in our remote area, and the Apple /// manuals (particularly Business Basic) are somewhat less than 100% clear.

To put your readers to the test, I offer two 'stumpers':

1) Overstriking in Business Basic: How does one create a print statement that will result in an overstrike (backspace or supress spacing) such that two or more characters occupy the 'same space' on the monitor or printer?

2) Changing FONTS 'on the fly': In both Apple Writer /// and Business Basic it would be handy to change character type at will. For example, I would like to 'sign' this letter in Script. How to?

Very Truly Yours,

Harold Graham
New York

Dear Mr. Graham,

We're here to help you so if you have any other questions or problems, write us! As answers to your questions, there is no way to doublestrike characters on the monitor screen. To doublestrike on a printer, after you send the first character to be printed send a BACKSPACE character to your printer and then the second character.

This BACKSPACE character is usually different from printer to printer so without knowing what printer you're working on I can't say exactly.

Our DOM #2 allows you to change the character type of an Apple DMP or Prowriter but you must boot a disk to use it. I don't know if anyone has devised a system to exactly what you want though. Hopefully one of our readers will have an answer.

Dear Mr. Consorti,

Thank you for your great magazine! I have thoroughly enjoyed both issues.

I've purchased the ON THREE O'Clock and it has been working perfectly. It was a very easy installation, and your directions were simple, direct, and easy to follow. I'm also looking forward to the next Disk Of the Month.

I have a few questions about software for the Apple ///. Do you know if and when the Apple)(Emulator will be revised to support 64K and the Apple //e software? What is the current version of Business Basic, Apple Pascal, Apple Writer ///, Mail List Manager, and of course SOS?

I've been looking for database management software, other than Quick File. Any suggestions?

A big disappointment has been the inability to play Apple)(games that require joysticks or paddles. Before subscribing to ON THREE I had purchased a product from TG to allow me to use joysticks and paddles in the emulation mode. Unfortunately, it didn't quite get the job done. I returned it to my local dealer. I have since purchased the Gameport /// from Micro-Sci that was mentioned in the last issue. It works!

Are you set up to receive mail on the SOURCE or CompuServe? Thanks again for a great magazine!

Sincerely Yours,

William J. Stevens
Vermont

Dear Mr. Stevens,

I know that some people are working on updates to the Emulation Disk but I don't know of anyone who is working specifically on a revision that would support //e software. Maybe one or more of our readers can help us on this.

The most current version of Business Basic is version 1.2, Apple Writer /// is version 1.0, Apple Pascal is version 1.1 (Just out!), MLM I don't know and the latest version of SOS is 1.3.

A handy DBMS is PFS: File & Report. We will soon publish an article that will show you how to easily use Quick File's files in your own programs. Thus if you like to program you can create custom programs that enhance Quick File.

Currently we are not set up on either of those systems, but there are other Apple /// SIG's on those services.

Gentlemen:

It is sad that your readers have lost hope on the long awaited clock for the Apple ///. I may as well join the group. Enclosed is my check to cover the cost of one clock.

In all other respects (except the clock), I have found Apple (and particularly my dealer) to be most supportive. My first Apple /// crashed with a catastrophic system failure and Apple replaced it without cost. However I still had constant problems with the replacement equipment and could never keep it on line longer than 2 hours without something happening. Apple put aside a third machine for me (again at no cost), when it occurred to me that I should have the line voltage checked.

When I called the electric utility's business office to request installation of a voltage recorder, they quite brazenly told me that the computer and disk drive failures were due to surges and low line voltage. "In fact," they said, "we have problems with home computer users all over the country." They could not immedately install a recorder because they were all out testing the voltage in the homes of computer owners and they suggested that I investigate the Sola Mini-UPS system. I blew another $1400 and purchased the model with 10 minute battery backup. I have never had any problems since.

In fact, I have just finished computerizing my methodology for electricity price forecasting which involves 20 huge spreadsheets, many of them interacting. I did this using Advanced Visicalc which allows columns of different widths and has some great new features. The Attribute options and recalculation feature are a big help. The programming allows the use of my own judgment each step of the way (every electricity utility analysis is different), but worksheets are filled in as if by magic and the Dif. format moves them into other tables. With the Advanced Visicalc I can do almost anything. Fantastic!

Very truly yours,

Gwen Crooks
California

Dear Gwen,

In a time when we hear so many bad things about Apple it's refreshing to hear some good news. I also know about the electrical problems of computers in the home and I have installed a surge suppressor to correct these difficulties.

Dear Bob,

I received the February-March issue of ON THREE, it is just what I need. Have tried all the other computer magazines and found little or nothing at all for the ///.

Enclosed is my check for a subscription. The clock sounds good, hope to order one next month. If you have a copy of the first issue I would like to have one.

I just got the Apple /// Business Graphics Software. I have been told that I can print out graphs, etc. on my printer which is a Xerox 620 but I cannot get it to work. If any of your readers have solved the problem I would appreciate hearing from them.

Once again, your magazine is terrific and I hope you can get enough subscriptions to keep it going.

Very Truly Yours,

Roland A. Lapierre
North Carolina

Dear Mr. Lapierre,

I'm glad you like the magazine and I hope we can keep you as satisfied in the future.

I'll put the question about the interface problem out to our readers in the next issue: Does anyone know how to print out graphics on a Xerox 620 connected to the ///?

Dear Bob,

I very much appreciate your taking time to answer my questions in your May 26th letter. I am pleased to hear that Apple will continue to support the /// as I have been hearing mixed reports. I hope you will disclose some of the plans as I would hate to invest in a lot of single-programs if Apple plans to devise an integrated Lisa-type program to work in the ///. Meanwhile, I shall look forward to examining the new Apple Writer and the Speller program. I have decided to get the Datafax program and will let you know how it works for me.

I have been using a conversion program from Apple Orchard (Apple-Con Converter) to convert some statistical Applesoft programs to Business Basic ///. The program works very well on all but the Graphics portions of the programs. Are you aware of any instructions or a program for converting Applesoft graphics to Business Basic /// graphics? My Apple Basic /// manuals seem a bit too limited to give me much help. At this point in time, I am not terribly interested in becoming a programmer although I may be forced into it in order to make some program modification to fit my needs.

I look forward to your future issues of ON THREE and hope you will include more information of use to your "user" readers who do not have prior computer experience. Literature specifically pertaining to the /// for the "user" seems rather meager. Also, I hope that you might inform us of some of the public domain programs for the ///.

The ON THREE O-Clock is working great. I was delighted with your very clear instructions and was able to install it myself. Is there any danger that the heat generated in the computer will cause the batteries to leak or do something worse? I wondered about putting on a longer set of wires to place the battery case outside of the computer but decided to leave it next to the speaker as was prescribed in the instructions. I will be interested in other equipment you discover for the ///.

Again, thanks for responding to my letter. Keep up the good work.

Sincerely,

Richard N. McKinney
Illinois

Dear Mr. McKinney,

As I said in my last letter, Apple is not going to discontinue support of the ///. Things may not be going too well but they aren't going to drop it.

I haven't heard of any instruction on converting Apple)(to Apple /// graphics but hopefully one of our readers will respond with the necessary information — an article or two would be nice.

I'm glad you like the clock. I wouldn't worry about heat ruining the batteries, it doesn't get hot enough to melt the battery holder.

Dear Bob,

I might as well be consistent like the other letters and congratulate you on your efforts with ON THREE. I think it is 'produced' in excellent taste and I am looking forward for future issues.

I would like to explain a little bit about myself and my company. I am 21 years of age and have been in the micro-computer field for six years. I have been the key person in opening three computer stores, Berliner Computer Center being my most recent. I recently resigned as one of the Vice Presidents to take on greater challenges.

We are an OEM developer for Apple Computer. We also do consultation and equipment sales to professional end users. Personally, I think that the Apple /// is one of the finest pieces of gear on the market. SOS is eons ahead of its time in both concept and implementation. We deal with many Apple /// users.

I have some information that you might want to put in your next issue. I hope it is of some help to your readers.

The SOS Reference Manual (A3L0027) and SOS Device Driver Writer's Guide (A3L0023) are now available at your dealer.

The Apple /// Record Processing Service (RPS — A3D0018) is also available. RPS is a set of intrinsic Pascal units that free the application software developer of the tedious and difficult tasks of key file management. RPS, when passed key values, will seek and retrieve your records. It will support files up to 16 meg. and it makes use of the SOS's sparse file ability (allocating 16 meg in the directory structure but only taking up what it really needs). RPS is based on a B+ Tree Index Structure which provides uniform performance as the database grows.

For further information on RPS, contact:

Vendor Support/PCSD Tech. Support Dept.
Apple Computer, Inc.
20525 Mariani Avenue
Cupertino, CA 95014

The reason Unofile/Duofile are not available is due to the marketing effort for Lisa. The Lisa project has these drives on allocation to enable Lisa to get to the end user.

Micro-Sci has available the A143 disk drive. This is an Apple /// compatible drive which gives the user 572K of disk space. It plugs into the daisy chain port in the back of the Apple /// and can be mixed with the normal Disk /// drives.

A note about SOS 1.3: It contains new drivers that fix the CP/M problem of Control-P not working at system level, it improves the performance and reliability of the ProFile driver, and it cleans up both the Console and Grafix drivers.

I hope the above information is of some help. I am looking forward to writing some articles for your magazine in the near future.

Sincerely,

Al DiBlasi
New York

Dear Al,

Thanks for your letter and all the information. I look forward to your upcoming article contributions.

Dear Bob,

I was intrigued by issue #1 enough to subscribe. Issue #2 met my expectations, leading me to believe that ON THREE would prove to be a valuable resource for /// owners. Unfortunately, issue #3 is a bit of a let down. There are a number of inaccuracies that I feel need a response.

Your reply to Thomas Spreitler's question regarding the most desirable way to interface an Epson MX-100 is incomplete, misleading, and unnecessarily harsh. No MX-100's sold anywhere in the world come from the factory set for serial operation. An additional board is necessary in the printer to convert a serial signal to parallel, the true nature of the printer. The two serial board options offered by Epson are both flawed, one not allowing the use of GrafTrax, the other not responding correctly to x—on/x—off protocol. While Practical Peripherals makes a very fine buffered serial board for installation in the Epson printer, the cost of this board rivals the cost of what I feel is the preferred interface for the Epson, the Apple UPIC Card.

Without knowing Mr. Spreitler's complete system, your comment that a dealer selling a serial interface card (presumably the Apple Super Serial Card) for interfacing a serialized Epson to a /// is 'cheating' and that Mr. Speitler should 'demand a refund' is inflammatory and ill advised. If his system includes modem, a common addition to a full feature system like the Apple ///, is it logical to use the built in serial port for the modem (its baud rate is software controllable) and a serial interface card for a fixed baud rate device like a serialized printer.

Mr. Lomartire's statement that Business & Professional Software does not have an Apple /// with which to experiment is incorrect. The company develops their software under the P-System and uses (and admires!) the /// for many of its development efforts. As to his interfacing problem, I'd like to point out that it is common, unfortunately, for printer manufacturers to change ROM's with no announcement to anyone. Epson has done this 4 times, each time causing a number of interfacing headaches for graphic applications in particular. This may be the cause of his problem.

Additionally, I think it is a poor idea to use a letter quality printer as a graphics output device. They weren't built for it. Most of the LQP's just bash the periods all over the page providing poor resolution and much wear. Someone moderately interested in graphic output would purchase a dot matrix printer, while the more serious user would purchase a plotter. To suggest that handshaking problems in emulation mode might be solved by 'pressuring the Apple reps' is not the answer. An Apple Super Serial Card is.

On the issue of emulation mode (a major mistake by Apple, the cause of many of their /// problems), too many people expect too much! Who else even bothered to try? Radio Shack? Cassette port in emulation is impossible simply because the hardware isn't there. The 64K emulation is extremely difficult because of the memory management hardware used in the machine. For those bit twidlers, lower case should be fairly easy to patch to the emulation disk.

Concerning the Micro-Sci add on floppies, it should be pointed out that these drives do not work under CP/M. Micro-Sci blames Microsoft for their implementation of drivers under CP/M (and I'm sure it's true), BUT Apple's floppies and ProFile and Davong's hard drives have no problem. It's easy to blame others, but the more support oriented and proficient companies seem to work around the problem. I don't like CP/M, but, like many /// owners, have found that the limited choice of SOS software has forced me to look at a number of CP/M offerings.

Regarding Apple's commitment and support of the ///, look at Apple's support of the)(. IT WASN'T APPLE, IT WAS THIRD PARTIES!! While Apple has provided the core software needed for the machine (spreadsheet, graphics, word processing, and communications), the most interesting software comes from the third parties (Quark, State of the Art, PFS, and others). IBM sells more PC's and XT's in 3 months than Apple has sold in total for the ///. It will be concerned users and publications like yours that will make the difference.

Sincerely,

Kevin Clougherty
New York

Dear Mr. Clougherty

I'm sorry that you feel we let you down with the last issue but I do believe your statements are incorrect.

The serial board that I am using in my Epson works fine with GraphTrax. Since this board was manufactured by Epson I can't accept your statements on this subject. Because the Apple /// has only four slots I have no intention of filling one with a single purpose UPIC card when the built-in serial interface works fine. The built-in serial interface option for the Epson cost me about forty dollars a couple of years ago and this is considerably less expensive than the UPIC card and it doesn't waste a slot.

You're right when you said my statement about demanding a refund is inflammatory, it was meant to be. I have seen far too many users cheated by dealers to sit back and watch again. His letter did give me an indication of his system, namely an Apple /// and an Epson MX-100. If you look over my reply to Mr. Spreitler I did say that if he is using a modem with his /// he will need a plug in card.

On the subject of Mr. Lomartire's letter, I can only base my opinions on a company by my own personal experience and by the notes that readers send me. Since I have received other complaints about that company I had no reason to doubt Mr. Lomartire.

Please remember that I receive many, many letters each day and that there is no way I can verify every statement of each letter. If any inaccuracies creep into the magazine they are not purposeful and we will try to correct them.

I don't know where you got your information, but the Micro-Sci add on floppies (all of them!) work just fine in the CP/M environment. You are right when you say that "the more support oriented and proficient companies seem to work around the problem...", Micro-Sci is a very supportive company. They have a fully qualified sales AND technical staff that will help you over any problem that you may be having.

The A143, A73 and A3 are the best (and only) buys in the floppy disk field for the /// — not simply because of their low prices, but because Micro-Sci provides some of the best customer support in the industry.

On the subject of Apple's support for the)(you are partially correct but you must remember exactly how the)(grew so popular. The reason for the vast amount of Apple)(software is because of the large number of 'hackers', those people who wrote programs as a hobby and found the)(a nice (and "open") machine to work with.

Tim Gill of Quark, John Page of Software Publishing Corporation and most all of the others started out at the bottom and worked upwards. One of the most important pieces of information that I received with my first Apple)(was what is now affectionately called the Big Red Book. It contained all the necessary information for programming and hardware interfacing of the Apple)(. That is what caused the explosion of Apple)(software.

The Apple /// only recently received the same type of 'Support' with the release of the SOS Reference Manuals and the Device Drivers Writer's Guide. In your eyes Apple didn't provide support for either the)(or the ///, yet relatively speaking the)((and //e) has gotten the lion's share.

NOTE: About the software which you say Apple has provided, three out of the four were not created by Apple: 1) Spreadsheet — Visicorp, 2) Graphics — Business & Professional Software and 3) Word Processing — Paul Lutus.

Dear Bob,

Enclosed is a check for my subscription to ON THREE. As an Apple /// owner who has been looking without success for published literature specifically for the Apple ///, I was extremely happy to receive your first issue. I have been to numerous stores and as you know Apple /// support material is very hard to find.

I enjoyed your articles, and find the letters to the editor especially helpful. One question on transporting the Apple ///. If, for example, I need to move my Apple from my office at home to my work location — Are there any special precautions I need to take if the machine is in the original carton and will be travelling in a car? Trips would be infrequent, but I was unsure how the internal components, especially the disk drive, react to the vibrations incurred in transportation.

Once again, thanks for your magazine. ON THREE is definitely an oasis in a desert of support literature. I am looking forward to the upcoming issue.

Sincerely,

Mark Snell
Ohio

Dear Mr. Snell

On the subject of moving your ///, after packing in the original carton the machine will go just about anywhere without problems.

About the only item you must be especially careful with is a hard disk. When transporting you must pack it in the original carton and handle it with care! Just a little bumping could cause serious problems with a hard disk drive.

Dear Bob,

Enclosed is my subscription to ON THREE. Also I would like to have you pass on the following comments to Apple Computer Inc.:

1) There is a definite lack of software for the Apple ///. This could be alleviated by development of a new emulation disk that is "fully" compatible with the new Apple //e.

2) Another approach would be the development of a softcard that would allow the /// to run the rapidly expanding amount of IBM software.

3) The Apple /// disk drives operate too slowly and do not create much storage capacity. Something new is definitely needed.

Thank you for your consideration.

Joseph M. Gevock
Illinois

Dear Mr. Gevock,

I have passed on your note to the various people at Apple. Hopefully some of the things you mentioned will soon be developed by either Apple or third party developers.

Dear Bob,

I appreciate the suggestions to use a word processor for editing Business Basic programs. That indeed is possible, but clumsy. Those who need the "GPLE" in the)(environment will appreciate the ease of doing the editing from within Basic rather than booting up another (locked) program.

A comment regarding Business Graphics interface with a printer. I found Business & Professional Software very helpful in providing a driver for the MX-100. Their original version came with a manual which clearly stated it could drive only the Qume or Silentype, so the buyer should have been forewarned, unless later editions changed that. BPS then developed a Printer Interface Kit (PIK) for the dealers which included drivers for a large number of other popularly used printers. BPS even offered to add the Epson driver to my program disk if I sent it in. I could not have wanted more support than that. My guess is that unless one has a version of Business Graphics /// that includes a driver specifically for his printer that it will be hopeless to try to get hard copies of the graphics.

Regarding the question from a reader for a Hires dump to a printer, I am surprised that your answer omitted reference to the PKASO card/software package, which was ably reviewed in the same issue!

You might want to remind your readers that Apple dealers should have available a new Software Revision disk which will update SOS 1.1 to version 1.3 on all of your boot disks. In addition to removing a few obscure bugs, SOS 1.3 is intended to support Backup ///'s unique ability to backup only files that have been modified since the last backup operation. The revision disk also has a series of new drivers for those who want to go to the trouble of updating their drivers to the latest (version 1.3) driver. I understand the revised drivers take care of a few bugs in the Pascal CP/M environments that are relatively obscure and unlikely to appear during normal operation.

Sincerely,

John M. Miller
Alaska

Dear Mr. Miller,

Hopefully someone will soon come up with a GPLE type editor for Business Basic. Until then, to ease the pain of constantly booting disks, use one of the interpreter switching programs like Quark's Catalyst that allow you to put Apple Writer, Basic, etc. on your hard disk and quickly go between them.

On the item about the HIRES dump and the Pkaso system, I simply wrote the reply to that letter before I got a chance to do the review.

All users should see their dealer about the new version of SOS (1.3) and update as soon as possible.

Mr. Consorti:

I was delighted to discover ON THREE, which will contribute greatly to enhanced use of the Apple ///. I found your review of Apple Writer /// quite to the point and generally balanced. It should be observed that there is an index for the manual — it is in the separate Operating Manual Addendum.

There are a few more serious problems with Apple Writer ///. On page 65 the manual states how to stop the printer during printing. This command has no effect on an MX-100 with Graftrax Plus. Another problem is that of forcing a null character into a MX-100 to turn off underlining or to turn on superscripting from Apple Writer /// through a UPIC. To be specific, how does one embed a bit pattern for NULL in the sequence 'CTRL V ESC — NULL CTRL V' from inside Apple Writer ///? It has been experimentally confirmed that an Fx-80 sequence 'CTRL V ESC - 0 CTRL V' shuts off the underline, but then Epson made a slight, but highly effective, modification to the control commands in the FX-80. It would be useful if Dr. Dorman would elaborate on his hardware solution for the UPIC card. I would also like to know the proper configuration for the DIP switches inside the MX-100 and the switch on UPIC (two Apple dealers give incompatible recommendations).

Most beneficial of all would be if Apple would cough up the Apple /// Technical Reference Manual (promised on page 68 of Apple /// Standard Device Manual). Apple's reluctance to deliver important information in a timely fashion to the Apple /// owners does Apple's cause little good. Such a lack of support is downright aggravating, especially in light of problems turning up.

I have a few questions on the ON THREE O'Clock. Does it require a driver or not? And how accurate is it? Is this chip the National Semiconductor MM58167 chip?

Sincerely,

Paul Schaich
California

Dear Mr. Schaich

The clock chip in our kit is very accurate and it doesn't need a driver. It is an update to the old National Semiconductor 58176 clock chip that didn't work with the ///.

Thanks for the note about the Apple Writer /// manual index. We recently purchased another /// and when we looked through the new Apple Writer /// package sure enough we found it.

The Alpine Printer Driver gets around the problem of nulls and Apple Writer ///. Using that driver full use of the features of the MX-100 are now possible with Apple Writer ///.

Hopefully we will soon see an Apple /// Technical Reference Manual. Only Apple knows if and when this will be released, but I also think it would help.

Dear Bob,

I am counting on finding a "friendly face" out there somewhere! You and ON THREE are giving me some hope.

My problem is that I have had my Apple /// for a little over a year and I am still pitifully ignorant about its operation. Don't misunderstand, I am enjoying it very, very much. Word processing (Apple Writer ///), three or four PFS applications and a little start with Business Basic have kept me occupied. My big frustration has been that I have to do a lot of travelling and so I get precious little time to learn.

Would you please help me with four questions:

(1) My son recently sent me a gift of Business Graphics. I was delighted but I soon found that I could not print out graphs! When I finally carved out some time to see my "friendly Apple Dealer" about enabling the program to work with my Epson MX-80 I was told that it would cost me $75 to get the proper driver put on my disk or disks! From airplane conversations I hear that this is highway robbery and nonsense. There has got to be another way. Is there?

(2) Some time ago in another conversation while travelling I discovered that my SOS should be updated. When I inquired at my "friendly Apple Dealer" they reluctantly gave me the update (SOS 1.1). Now I see that you have a more recent update (SOS 1.3). Do I simply substitute SOS.KERNEL for those on my bootable disks? Or is there more to it than that?

(3) Since I have no understanding of Pascal, I cannot make head nor tail of a lot of things that you write about or that are on the January DOM. In fact I am so backward that it took me quite a while before I could even follow your directions for using the disk. My expanded PFS disks work fine except for the fact that the Utilities disk warns me that the disk is damaged! Your correction on the Fondemo helped me with one problem, of course. But tell me, will I need to get into Pascal if I am to get the most out of ON THREE or even out of my Apple ///?

(4) Finally, do you know of any Apple /// user groups in Manhattan? My "friendly Apple Dealer" has repeatedly ignored my inquiries. The Computer Factory had persuaded me that I would be wise to pay them a higher price for my Apple /// because they would be giving me the support that a beginner needs. HA! Since I am not a Corporation they are clearly not interested in helping me. I cannot afford either the time or the patience to continue to deal with such an outfit. But that clearly means that I must find some source of reliable information close at hand — and also a reliable repair source.

I am eagerly looking forward to the next issue of ON THREE. I hope that Apple /// owners will rally to your support so that you can keep up what you have started so well.

Sincerely,

Francis K. Wagschal
New York

Dear Mr. Wagschal,

The price on the necessary drivers for Business Graphics is a bit high! From the rest of your letter I'd suggest looking for a different dealer. Other dealers should be able to solve your problem with little or no cost.

It's a sad fact but most dealers (regardless of what you paid for your system) aren't supporting the /// in any way, shape or form. Your dealer has the new SOS 1.3 and they should be able to update your disks for no charge. If not, just copy the SOS.KERNEL from the DOM to your disks.

In addition to copying the SOS.KERNEL file, you should also update to the 1.3 version of the CONSOLE driver and other drivers included on the DOM. Likewise your dealer will have all the new drivers.

If we do our jobs right you won't have to learn Pascal to get the most out of ON THREE. Please tell us if you need some help on a particular subject. There is no problem with that error message using the Utilities disk. It's normal.

I don't know of any Apple /// user groups in your area but when we get notices of user groups we publish them.

Dear Bob,

Well, here's my $30; sign me up as a new member! I was very impressed with the February-March issue of ON THREE; just what I've been looking for. How can I get my hands on the January issue?

Your article "SOS Directory Structure Revealed" prompted several questions regarding the current state of affairs existing between Mother Apple, Apple /// dealers, and the bottom of the totem pole, we lowly Apple /// owners and enthusiasts. Firstly, have you any news regarding the whereabouts of the often rumored "SOS Reference Manual?". No one in my area seems to know that it exists!

Your comment regarding the "old Version 4.0" of the System Utilities Disk really surprised me. For over a year now, I've been merrily listing, deleting and swapping files to and fro with Version 1.1, thinking all the while how I'd been blessed with state-of-the-art everything. Numerous inquires to Mother Apple, the local Apple service rep, and several Apple dealers yielded the same result: "Version 1.1 is the only existing version of the System Utilities Disk." However, much to my surprise, I discovered that SOS has been updated to Version 1.3 (what ever happened to Version 1.2??). A trip to my local dealer did indeed turn up a copy of SOS 1.3, but still insistent claims that Verson 1.1 of the Utilities Disk is the one and only. What gives?

In response to Dr. Dorman's query regarding Business Graphics /// screen dumps via a UPIC card, it is indeed true that one can do graphic screen dumps to an Epson MX-80 (and a MX-80 Graphtrax, a MX-100, and numerous other printers). Your dealer should have an entire disk chock full of printer drivers which can be added onto your copy of Business Graphics///. To dump a graphic screen to my MX-100, all I need to do is enter:

```
SET UNIT 6
WRITE SCREEN MX-100
```

The only minor drawback is that this method supports only full-page (10 x 7) inch graphs.

Onward with more Pascal routines! As a novice Pascal-er, every little bit helps. Martin Nichol's program on reading that second keyboard byte is really neat; now if I can only find the time to incorporate it into my System Library.

Sincerely,

John W. Morgan
Texas

Dear Mr. Morgan,

I have to apologize to you and everyone else who has asked for a copy of the January issue. By the time this letter reaches print we should have done a reprint of the January issue and sent a copy to you. The reason we haven't done a reprint before now is simply because not enough people (for us to break even money wise) have ordered it.

In reference to the long awaited SOS manuals, they have been out for a number of months now — but for the average user to get them, you must order them! The part number is #A3L0027. With this number your dealer can get them. (Note: Since replying to this letter I found out that only 750 copies of the SOS Reference Manuals were originally printed. By the time this reaches print Apple will have done another printing and you should soon be able to get a copy.)

I suppose I should clarify the item about the old System Utilities Disk. A long time ago (2+ years ago) there was a version of SOS with the numbers 1.0. This was the first released version of the Apple /// operating system. Part of the initial release was a crude version of the Utilities disk with the Version number 4.0. When Apple came out with the SOS 1.1 they also had an updated version of the Utilities disk.

As far as I can tell, just for the sake of having similar numbers, Apple put "Version 1.1" in the upper right hand corner instead of "Version 4.1" or "Version 5.0". To date the most current System Utilities release is the 1.1 version. Everyone should also update their SOS to the new 1.3. I don't believe version 1.2 was ever released to the general population of Apple /// users.

We look forward to helping you with all of your Apple /// questions and problems. ///

The Editor's Block:

Continued...

Brooks Lyman joins the ranks of contributing authors with his fine article 'PFS: Hints and Shortcuts'. This one shows you how to get the most out of that very good Data Base Management System. John Lomartire also brings us some useful information with his short piece 'WPL Shorts'. Read it over and you'll learn how someone else has automated their word processing using Apple Writer /// and WPL.

Elwynn Taylor shows us how to change the Cursor /// to operate a little more effectively in emulation mode, and George Oetzel brings us another hardware construction project. His article allows you to get color in the emulation mode!

Michael Gardner brings us a little program to print out headers before your program listings, and Dean E. Russell shows us how to do Spread Sheeting from Basic! Stephan M. Dorman and L.A. Ditson have teamed up in this issue to write Graphics Sketcher ///, a very nice graphics utility program that lets you draw some interesting pictures and shapes on your graphic screens.

Just as he promised, Martin Nichols brings us a very handy utility program this month. Using the routines he gives, all programs can now format their own disks. Very well thought out and documented, the routines will even format Micro-Sci drives! the WPL Tutorial is back with some more useful things to do with that very underrated language.

While we were able to persuade Earl Curlson to write some more of his very popular columns on learning Basic — The Easy Way, Louis Hanson has other commitments and won't be able to write any more of the Pascal tutorials for us in the forseeable future. That means that we're going to have to find someone else well versed in that language to put together a tutorial. Anyone out there feel that they are qualified?

What's on-line for next time? Well, if you think this issue is good, wait until you see the next one! We're now getting bigger and better thanks to YOU. Upcoming will be a complete dissection of MLM and a Pascal Unit to aid you in using your MLM files the way you want to. Mail list merging with PFS and WPL, and even Making Decisions with the help Visicalc. A mini-graphics tutorial will be upcoming, a PasCalculator? and a special on automating Access ///. We've got a lot of things coming up, so stick around — there's something for everyone.

Before you turn the page and become engrossed with all the info. in this issue, let me put out another call to our readers who know their machine. We are trying to set up a HOT-LINE service to help Apple /// users overcome their problems. If you know enough to answer questions, please write or give us a call. Response has been less than pleasing so far and I want to get this rolling. We all have to band together and help one another over the rough spots, so I'll ask again — please drop us a note or give us a call. ///

Hints and Shortcuts with PFS

by Brooks Lyman

If you are a doctor, sell real estate or manage some other small business the chances are great that you have a data management problem. Fortunately, the Software Publishing Corporation offers a package for Apple /// owners called PFS:FILE, which is both simple to use and enormously helpful. In addition, they offer programs, called PFS:REPORT and PFS:GRAPH, which are used to summarize and manipulate those data or display it in graphical form. While the manuals which accompany the program diskettes are very adequate, the little niceties are learned only after many hours of using them. Perhaps my experiences in keeping track of the many variables in growing avocados will be of help to illustrate some pitfalls to avoid and shortcuts which can be implemented.

Form Design

Users of PFS:FILE surely understand that the system is based upon custom designing one's own forms. The blank spaces, which the user provides, are used later for the data to be entered. There will be a great temptation to design a "pretty" form. In fact, instructions given with the Radio Shack TRS-80 equivalent, named PROFILE, tell the user how to "pretty it up". Don't do it! Every keystroke used in designing the form should be with two thoughts in mind: will it make the data entry easier; will it make data reduction easier?

Terminate Titles with a Colon

In designing your form, keep it simple, which includes avoiding titles if possible. Should your design require a title at the top of the page remember that unless it is terminated with a colon, as for any other heading, the title will become the beginning of the first item name. This in itself is not a problem except you must remember that if the title is changed in any way at some later date you will lose all of the data which has been stored in the first item. Now if you still feel compelled to have a title and decide to terminate it with a colon, place the colon unobtrusively in column 80 of the title line. (This safe way to use a title is illustrated in Fig. 1)

Figure 1.

RECORD OF AVOCADO PURCHASE		:
NAME:	PICKING DATE	:
ADDRESS:	PRICE PER LB	:
CITY:	PICKING CHGE	:
PHONE:	CONTRACTOR	:
GROVE ADDRESS:	TARE 165# OR	:
	WEIGHMASTER	:
RECEIVED BY:	FIELD REP	:
	INVOICE #	:

Align Colons Vertically

A fine point which most users seem to ignore is to align the item-name colons vertically. You will appreciate this detail later particularly if the form you have designed is complex and has a large number of headings. To explain further, there are three ways to move the cursor to an item: you may use the tab key (one stroke to advance one item); you may use the return key (not "Enter") to advance the cursor one line at a time down the extreme left edge of the form; and you may use the direction arrows (which need no further explanation). The idealized form in Fig. 1 illustrates the concept of colon alignment. Note that it is quite simple to advance the cursor to a desired item by simply keeping the tab key or down arrow key depressed. While it is true enough that depressing and holding the tab key will eventually move the cursor to any item on a form, no matter how randomly it was designed, you will see very quickly that its progress across and down the page is much more difficult to track by eye if the headings have not been aligned as recommended. Whenever design permits, keep them aligned.

Leave Plenty of Space After the Colons

The manual advises users to allow adequate space between the previous colon and the next item name inasmuch as you may want to "Set New Headings". I would do one step further and URGE you to do that! PFS:REPORT has a useful feature which allows one to change any heading at will. The original heading will remain associated with the PFS:FILE form but the heading which is displayed in a report can be quite different. Consider the case where an item is headed by, say, "GALLONS OF CHLOROFORM:" In a report, you wish to generate, however, you may wish to shorten this to, say, "GALLONS". If you had provided only two or three spaces on the form because you knew in advance that the quantity would never exceed 2 digits, imagine your problem when you attempt to fit the word "GALLONS" into two spaces. By having left some extra spaces, just in case, you will not be forced to improvise later on if changing the heading is necessary.

Date First

Most of the forms you design will probably contain a date. Furthermore, it is likely you will often want to retrieve information according to some time span. Since the fastest retrieval (3-5 seconds) is obtained with a "full item match" in the first item of the form, you can see the advantage in making the date the first item. (Obviously, if some other item will be used most often for retrieval, by all means place that item first.)

Lower Case for Data

I cannot recall ever having seen a specimen form whose item names were not fully capitalized. This seems to be the natural thing to do. On the other hand, neither can I recall having seen others enter data on these forms in lower case letters. While it may take a bit more effort to capitalize the first word and then use lower case letters, I believe you will find that it is easier to scan the form when the data are not capitalized. Similarly, reports have a more finished look when the column of data and the headings are not typed alike.

Digit "1" Instead of "X"

You can often save valuable data entry time by selecting item names which need to be identified by only a single keystroke: "Y" for "Yes", "N" for "No", and "X", etc. Whenever this occurs, I find it prudent to use the digit "1" (or "0" for "No") instead. You may find a need sometime to use these data in summaries and it is always nice to be able to manipulate them arithmetically as opposed to simply counting the "Y"s, "N"s or "X"s.

Changing Form Design

No matter how careful you are, the laws of probability insure that eventually you will have to make changes to a form you have already designed. You have been told time and again that, if not already done, you MUST make a backup copy first. This precaution is especially important if the form already contains data. So a final warning: no matter how trivial the form may seem at the moment, make a copy of it before attempting to change the design!

The procedure for making a design change is quite simple but here are a few hints which may save you some anxiety:

Experiment with the Design Early

Unless you have worked out all of the formatting details with pencil and paper beforehand, it is quite likely that you will decide the initial form design will benefit from a slight improvement the instant you press the "Enter" key to save it. (Murphy's Law at work, you know.) Fortunately, this is the ideal time to make changes, if change you must. Corrections to a blank form can be effected in virtually moments. On the other hand, if your form is relatively complicated and you have already stored a large quantity of data, be prepared for a long long wait while your machine labors to correct your indiscretions. Twenty minutes is not unusual at all. In fact on one occasion I left the console to have lunch while the data was being juggled around. The message here is that if you are going to make changes, try to make them before a lot of data has been entered.

Provide Sufficient Space for Each Item in the Redesigned Form

If you are an inveterate form changer, eventually you will run into the condition where insufficient space has been left after a colon (in the redesign) to accept all of the data which was stored in the original. The program, anticipating such an error on your part, will stop the redesign process and warn you that an item is too long. To make matters easy, however, the cursor will have been positioned at the item in question. The solution to your problem is equally easy. Simply shorten the item by judicious abbreviation and press "Enter". If your abbreviation was adequate the revision process will continue. If the item is still too long, the machine will repeat the message until you finally force a fit of some sort.

The Redesigned Item Names Must be Letter-Perfect

Suppose you want to change the position of an item on a form. As stated above, the procedure is quite simple. Remember, however, that the item name cannot be altered in ANY way whatsoever at the risk of losing all of the data which is associated with it. The program has absolutely no way of knowing where to put data on the redesigned form unless it can find an item name identical to the original. In connection with moving items from one location to another, rest assured that the same straightforward procedure even permits you to move an item from one page to another — so long as you do not make a change to the item name itself.

Wait for PFS:FILE Main Menu to Reappear

Assuming that you have done everything properly, some time after the form change process has begun, you will notice that the program appears to hang up while displaying "Form #1". Keep your cool (and your mitts off the keyboard)! In a little while the process will start up again and, hopefully, continue to the end. The signal that the changes have been completed is, of course, the display of the Main Menu. At this point you will probably select "4", and with baited breath, test to see if things came out as you'd planned. They will!

— PROTECT — YOUR DATA INVESTMENT ADVICE FROM SOFTWARE PUBLISHING CORPORATION

The Software Publishing Corporation once sent me some advice which I have not seen published elsewhere. A portion is included here with their permission:

"You should have at least two backup copies of each of your files. On Monday, backup your files to one of the backup disks. On Tuesday, backup to the second one. On Wednesday, backup to the disk that you used on Monday. Each day, alternate backup disks. This way, if you encounter a problem in the middle of backing up (which not only tells you that your data file is bad, but that your backup now has the bad data also) you still have the previous day's backup disk.

The frequency of backing up depends on how much information you can afford to potentially lose. If you rarely update your files, then backing up once a week is enough. If you are updating your files daily, you should back up your files daily (or twice a day)."

Broken Data Base

Some day you may experience the horror of a broken data base. That is, you will be unable to access information which you know is contained on the disk.

"Symptoms of a broken data base are:

PFS tells you the disk is full, when it isn't.
Your disk is full, but you can't remove any forms.
Negative form numbers appear on the screen.
Part of the design of your form is missing.
Some of the data of your form are in inverse.
PFS says it has an I/O error.
Searching stops suddenly and dies."

Software Publishing goes on to tell the many causes of a broken data base: magnetic fields, fingerprints, bending disk, etc. One which you may do inadvertently is to remove a data disk or "Reset" before you are at the Main Menu. Be careful!

EDITOR'S NOTE: What happens if your data base goes bad? Hopefully you will never have to face this question, but in the event it happens don't worry! Software Publishing Corporation has a program called PFS:RESCUE. Most dealers who sell the PFS line have a copy of it. It's great! It recovers whatever data that is left on the disk. Simply bring a copy of the bad disk to your local dealer and they will have the program restore whatever it can.

Enhanced Version of PFS

Boot your PFS:FILE then, before doing anything else, enter the letter V in the space for "SELECTION NUMBER:" When you then press "Enter" the version number of your program will appear at the bottom of the screen. If the version number is B.01, you have an enhanced version which insures that your files are updated even if you fail to return to the main menu. (Depending on the version you have, the company may exchange your disk(s) at no charge. Call Software Publishing Customer Service for the update policy). PFS:REPORT is not affected by this enhancement.

Apple /// Clock

ON THREE offers a "clock-calendar" for your Apple ///. At $39.95 it's a steal (mine cost $100.00) and you will find it a very useful accessory, indeed. Not only will the clock record the date and time automatically as a file is created or changed, but it will also insert the date or time on command after entering "Control D" or "Control T" from the keyboard. You will find it very gratifying to enter (D) or (T) and see the current date or time appear. Unfortunately the latter feature is available only when you use PFS:FILE. It is not available when you use PFS:REPORT. (Software Publishing contends this is because there was insufficient room in the PFS:REPORT program. I hope they find room, somehow, the next time someone gets the urge to enhance their latest version.)

Search Criteria

Page 4-5 of your PFS:FILE Users Manual lists a summary of all the retrieve specifications one may use. I would like to suggest another which I discovered by accident. If you enter ".." as a retrieve specification the program will display every form in the file that contains ANY data whatsoever in that item. For example, if you have completely forgotten the nature of the information contained in a given item, simply enter the two periods as a retrieve specification. I know of no other way to find these data except to display each form, one by one, to see what is contained at that location. Try it!

Make a Hard Copy

There may come a time when you create a file which is of such great importance that you just cannot afford to lose it — no matter what! When this moment arrives you had better think in terms of making a hard copy of the file contents. You have then a number of options:

Print out a complete copy, page-by-page. (Rather time consuming)
Print out a copy without item names. (Faster but hard to read back)
Print out selected portions. (see bottom of p 5.4 PFS:FILE Manual)
Use PFS:REPORT. (The best of options above)

PFS:REPORT

If you already own PFS:REPORT you certainly know its principal features: alpha-numeric sorting, column calculations, derived columns using all conventional arithmetic operations, etc. In fact, PFS:REPORT is such a genuinely useful companion program to PFS:FILE that it is a mistake not to have both — as anyone who has used them in combination will attest.

Sorting

The manual explains in considerable detail how to use column 1 and 2 to sort alphabetically or numerically (p. 1-9). It may not be quite so clear regarding unsorted reports. Specifically, if you call for data to be printed in column 1 or 2, it will be sorted whether you want it or not. On the other hand, the same data directed into column 3, for example, will not be sorted.

Machine Hangs Up

I have found, on my system at least, that the computer will hang up if you try to produce an UNSORTED report immediately after booting the program. The only way to get restarted is to reboot the REPORT diskette. Unfortunately this condition persists until a report is specified to be sorted; after that, the non-sorted report will run. (Why is that? I have not the faintest idea. Software Publishing believes it may be a "cold start" problem with my Apple ///. Whatever the cause, if you experience the same problem, at least be comforted that you are not alone.)

Alpha vs. Numeric Sorting

The manual explains how to sort numbers by specifying column 1 (or 2) and including the letter "N" to indicate that the numbers are to be treated numerically. This works very nicely but perhaps the sorting sequence is backwards from what you want. That is, the highest number appears at the head of the column, etc. Suppose you want the numbers to be sorted with the lowest number first. (I always do.) You will find that omitting the letter "N" will cause the numbers to sort the way you want — well almost. Since the program treats the numbers as words, in a sequence from, say, 1 to 15 the numbers will be arranged 1, 10, 11, 12, 13, 14, 15, 2, 3, 4, 5, 6, 7, 8, 9. The arrangement is equally distressing as the sequence continues in length. Fortunately, there is a way out of this predicament. Simply add enough leading zeros so that each number contains the same number of digits. (ie. For the example above: 01, 02, etc.) Obviously you will have to take care of this little detail at the time the data is entered in PFS:FILE. After that it is too late.

In connection with numerics, the manual does not state that it is unnecessary to specify the letter "N" if you elect to do an arithmetic operation on numbers. For example, suppose you specify the above sequence (1-15) in column 1 of the report. If you elect to average this sequence by specifying that operation as a column calculation the program immediately understands that these are to be treated numerically and will not only perform the averaging routine as specified but will also sort the numbers in high-to-low order. On the other hand, had you specified that the sequence of numbers be counted, the program will count as instructed but sort in the alpha mode — from low-to-high.

Calculations

PFS:REPORT does all of its calculations in a mathematically correct manner. Unfortunately many of us are not mathematicians and can get some unexpected results from seemingly straightforward arithmetic.

This section is not intended to be a lesson in mathematics but there are a few tips which may make your life easier.

Precedence

Arithmetic operations are carried out according to a precedence recognized the world over. The operations are done in the following order:

First	Apply a minus sign to the appropriate number.
Second	Perform all exponentions.
Third	Perform all multiplications and divisions. (Always from left to right)
Last	Perform all additions and subtractions. (Always from left to right)

For those who don't work with numbers too often, there is a way around these rules. Fortunately PFS:REPORT understands the way to avoid Precedence also.

If you want to enclose parts of your arithmetic expression in parentheses the program will perform that portion of the problem first. Similarly, if your expression is long, you may enclose other portions in parentheses as well.

Now Back to Plain English

Suppose you want PFS:REPORT to add the numbers 5 and 6, multiply the result by 3, and then add 50 to the whole business. Obviously there are provisions for doing this using the rules of precedence. But if you have forgotten how, simply write your expression just as you would say it using parentheses to enclose each step.

$$((5 + 6) * 3) + 50$$

Note that the program executes the portions in the innermost parentheses first. Consequently 5 and 6 are added to get 11. Then, since 11 * 3 remains in the next outer set of parentheses, that is done to produce 33. Finally 33 and 50 are added to obtain 83. Note also, though, that if parentheses had not been used the expression would have been evaluated by the machine with an answer of 73.

Precision

One final warning on arithmetic. PFS:REPORT usually treats numbers as integers unless otherwise instructed. That is, if you have the digits 1, 2, 3, and 4 on a form in PFS:FILE and you instruct PFS:REPORT to average them, your answer will be 3 and not 2.5. as you might have expected. To obtain greater precision in your answer it will be necessary to include a decimal point and an appropriate number of following zeros. As a result, if you have added 3 zeros you will be given an answer correct to 3 decimal places. (In the above example: 2.500)

Now let us assume that your PFS:FILE is all loaded with data and you suddenly decide it would be nice to average some of it. Too late? The items are all integers. Not at all! Simply add a decimal point and an appropriate number of zeros to any single one of the numbers in question. Thereafter, PFS will process them all as if each had the same number of zeros.

Predefined Reports

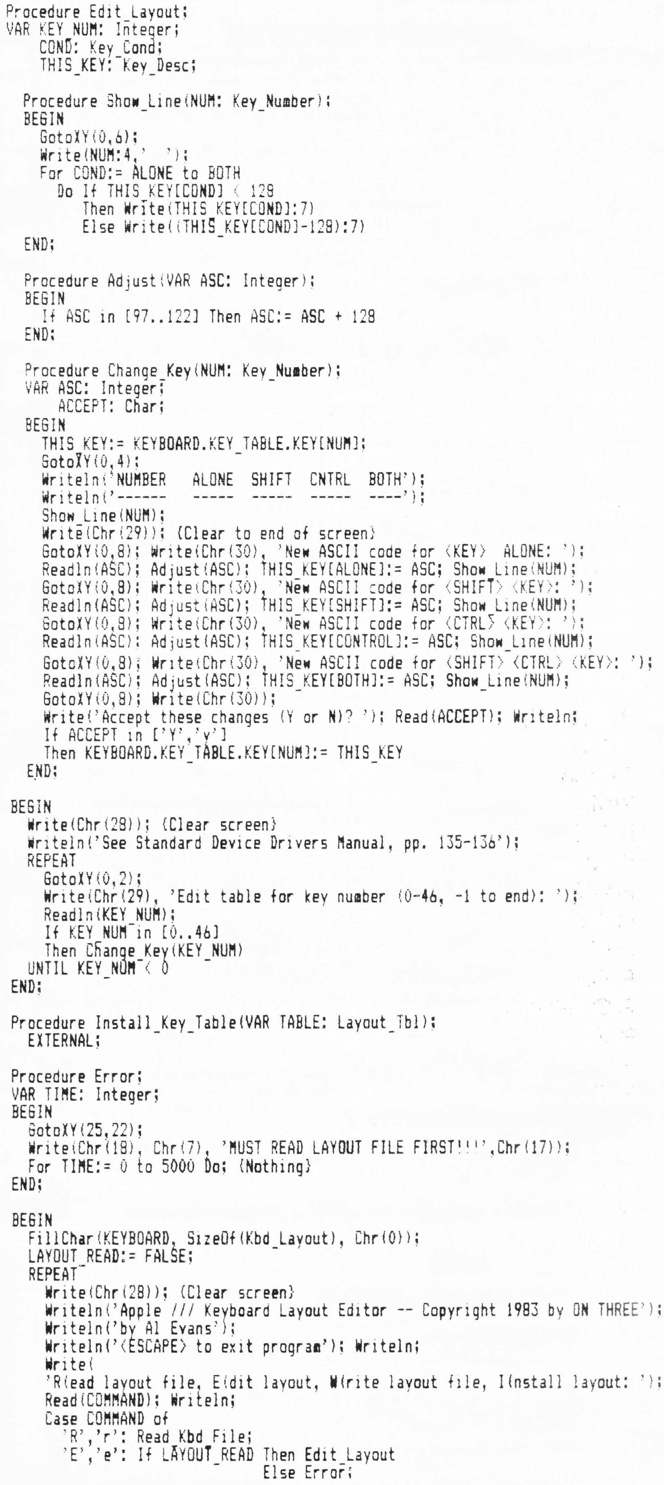
You will want to take full advantage of the feature "Predefined Reports". The instructions are sufficiently clear for anyone to follow. Some day, however, you may find that the 8 reports you are permitted to predefine are not sufficient. Would you like 8 more? Simple!

Using the PFS:FILE commands to copy the entire file, duplicate the file on the same diskette (preferably) but use a slightly different name. For example, add the digit 1 as a suffix. When you then use PFS:REPORT you will find that the new file has the same "Predefined Reports" (and modified headings) as its twin on the disk — EXCEPT — they can both be changed in any manner you want without affecting the original!

I presume this procedure can be repeated for adding as many predefined reports as desired.

This brings us to the end of Hints And Shortcuts with PFS. If YOU have any more to add, write ON THREE and show the rest of us the things you've found about this great data base management system. ///

Continued from page 33...



```
Procedure Edit_Layout;
VAR KEY_NUM: Integer;
    COND: Key_Cond;
    THIS_KEY: Key_Desc;

  Procedure Show_Line(NUM: Key_Number);
  BEGIN
    GotoXY(0,6);
    Write(NUM:4,'  ');
    For COND:= ALONE to BOTH
      Do If THIS_KEY[COND] < 128
         Then Write(THIS_KEY[COND]:7)
         Else Write((THIS_KEY[COND]-128):7)
  END;

  Procedure Adjust(VAR ASC: Integer);
  BEGIN
    If ASC in [97..122] Then ASC:= ASC + 128
  END;

  Procedure Change_Key(NUM: Key_Number);
  VAR ASC: Integer;
      ACCEPT: Char;
  BEGIN
    THIS_KEY:= KEYBOARD.KEY_TABLE.KEY[NUM];
    GotoXY(0,4);
    Writeln('NUMBER   ALONE  SHIFT  CNTRL  BOTH');
    Writeln('------   -----  -----  -----  ----');
    Show_Line(NUM);
    Write(Chr(29)); {Clear to end of screen}
    GotoXY(0,8); Write(Chr(30), 'New ASCII code for <KEY>  ALONE: ');
    Readln(ASC); Adjust(ASC); THIS_KEY[ALONE]:= ASC; Show_Line(NUM);
    GotoXY(0,8); Write(Chr(30), 'New ASCII code for <SHIFT> <KEY>: ');
    Readln(ASC); Adjust(ASC); THIS_KEY[SHIFT]:= ASC; Show_Line(NUM);
    GotoXY(0,8); Write(Chr(30), 'New ASCII code for <CTRL> <KEY>: ');
    Readln(ASC); Adjust(ASC); THIS_KEY[CONTROL]:= ASC; Show_Line(NUM);
    GotoXY(0,8); Write(Chr(30), 'New ASCII code for <SHIFT> <CTRL> <KEY>: ');
    Readln(ASC); Adjust(ASC); THIS_KEY[BOTH]:= ASC; Show_Line(NUM);
    GotoXY(0,8); Write(Chr(30));
    Write('Accept these changes (Y or N)? '); Read(ACCEPT); Writeln;
    If ACCEPT in ['Y','y']
    Then KEYBOARD.KEY_TABLE.KEY[NUM]:= THIS_KEY
  END;

BEGIN
  Write(Chr(28)); {Clear screen}
  Writeln('See Standard Device Drivers Manual, pp. 135-136');
  REPEAT
    GotoXY(0,2);
    Write(Chr(29), 'Edit table for key number (0-46, -1 to end): ');
    Readln(KEY_NUM);
    If KEY_NUM in [0..46]
    Then Change_Key(KEY_NUM)
  UNTIL KEY_NUM < 0
END;

Procedure Install_Key_Table(VAR TABLE: Layout_Tbl);
  EXTERNAL;

Procedure Error;
VAR TIME: Integer;
BEGIN
  GotoXY(25,22);
  Write(Chr(18), Chr(7), 'MUST READ LAYOUT FILE FIRST!!!',Chr(17));
  For TIME:= 0 to 5000 Do; {Nothing}
END;

BEGIN
  FillChar(KEYBOARD, SizeOf(Kbd_Layout), Chr(0));
  LAYOUT_READ:= FALSE;
  REPEAT
    Write(Chr(28)); {Clear screen}
    Writeln('Apple /// Keyboard Layout Editor -- Copyright 1983 by ON THREE');
    Writeln('by Al Evans');
    Writeln('<ESCAPE> to exit program'); Writeln;
    Write(
    'R(ead layout file, E(dit layout, W(rite layout file, I(nstall layout: ');
    Read(COMMAND); Writeln;
    Case COMMAND of
      'R','r': Read_Kbd_File;
      'E','e': If LAYOUT_READ Then Edit_Layout
                              Else Error;
```

Program Listing Continued on Next Page

WPL Shorts

by John Lomartire

The April-May issue of ON THREE carried an article (page 20) on WPL with Apple Writer /// which asked for comments and examples of use. I have been using WPL for some time now and I am convinced that it is one of the strongest assets of Apple Writer ///. As a matter of fact, I use it every day in my correspondence in the following fashion.

There are three standard formats on which I do 90% of my word processing.:

8 1/2 by 11 inch sheet
5 1/2 by 8 1/2 sheet
11 by 15 inch sheet

I have therefore set up these 5 WPL programs:

```
PRINT.1           VIDEO.1      File names
  PLM10             PLMO
  PPMO              PPMO
  PRM88             PRM78
  PTMO              PTMO
  PBMO              PBMO
  PPN1              PPN1
  PPL54             PPL54
  PPI66             PPI66
  PLIO              PLIO
  PSPO              PSP1
  PPD.PRINTER       PPD.CONSOLE
  PCR1              PCRO
  PLJ               PLJ
  PNP               PNP

PRINT.2           VIDEO.2
  PLM6              PLM6
  PPMO              PPMO
  PRM58             PRM58
  PTMO              PTMO
  PBMO              PBMO
  PPN1              PPN1
  PPL42             PPL42
  PPI51             PPI51
  PLIO              PLIO
  PSPO              PSP1
  PPD.PRINTER       PPD.CONSOLE
  PCR.1             PCRO
  PLJ               PLJ
  PNP               PNP

PRINT.3
  PLM00
  PPM0
  PRM160
  PTMO
  PBMO
  PNP1
  PPL57
  PPI66
  PLIO
  PSP1
  PPD.PRINTER
  PCR.1
  PLJ
  PNP
```

So it is an easy matter to enter my text, then typing either:

(P) DO .D2/VIDEO.1 or (P) DO .D2/VIDEO.2

I can see the manuscript formatted on the screen in the correct size, decide on page breaks, examine left and right margin appearance, etc. Once satisfied, I can type either:

(P) DO .D2/PRINT.1 or (P) DO .D2/PRINT.2

as desired to get hard copy. If I am printing on wide sheets, then my command would be: (P) DO .D2/PRINT.3

That article suggests pressing the TAB Key before each WPL entry line, and although there is nothing wrong with this method, it does use up a lot of memory since all those tabs are counted as characters. A single space produces the same effect.

I hope this information is useful to other Apple Writer /// users.

EDITOR'S NOTE: It's true, I do press the TAB key before each WPL program line. However, most of the time I've set up the tab stops so that the first TAB pressed in each line only gives a space. I find this is easier than pressing the space bar (habit!). ///

Program Listing: Continued from last page

```
          'W','w': If LAYOUT_READ Then Write_Kbd_File
                                    Else Error;
          'I','i': If LAYOUT_READ Then Install_Key_Table(KEYBOARD.KEY_TABLE)
                                    Else Error;
        END
      UNTIL COMMAND = Chr(27);
      Write(Chr(28)); {Clear screen}
    END.
```

KeyBoardLayout: Program Listing #2

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

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

;---------------------------------------------------------------------
;Pascal declaration Procedure Install_Key_Table(VAR TABLE: Layout_Tbl);
;---------------------------------------------------------------------

                .PROC           INSTALL_KEY_TABLE,1

RETURN          .EQU            0
SOURCE          .EQU            0E0
DEST            .EQU            2

BEGIN           POP             RETURN
                POP             SOURCE
                LDA             #0
                STA             DEST
                LDA             #17
                STA             DEST+1
                LDY             #0
LOOP            LDA             @SOURCE,Y
                STA             @DEST,Y
                INY
                BNE             LOOP
END             PUSH            RETURN
                RTS

                .END
```

SpreadSheeting From Basic

by Dean E. Russell

The following program is an example of spreadsheet type entry using Business Basic. The program is a demonstration that shows what can be done. Someone who would take the time to follow the program through and understand it could use a version of it with their own modifications in a Basic program.

The program allows for a window into a two dimensional array. you can think of it as rows and columns like a spreadsheet program. To understand how this program operates will take a little work, but after you understand how it works it will give you a powerful tool for text or data input and correction in Basic programs. Take out the window dressing, remove the REM statements, shorten variable names, and you will find that this program is not very long. You could use the cursor to designate the working cell instead of highlighting it and shorten the program further.

The display is a 6 column by 10 row segment of the array. The example has a column width of five characters. Remember this isn't Visicalc. The column width can be anything you want but you have to change the many things in the program that have to do with column width. I suggest that you take a piece of graph paper and draw the different overlapping windows on it in different colors so that you can get the feel of what is happening. I have used long variable names to help make the program easier to follow. The REM statements describe in a general way what the lines and statements that follow them do.

Let your imagination run wild. Add a dimension and call it page. Now you have sheets of information available limited only by the 64K byte size of an Apple /// array. This program can be developed so that the number of rows, columns, column widths, placement of windows on the screen, titles, etc., could be controlled by program commands. ///

SpreadSheeting: Program Listing

```
0    REM ************************************************************************
1    REM *                                                  ------------------- *
2    REM *  SpreadSheeting From Basic                       | Copyright 1983 by | *
3    REM *  -------------------------                       |  O N   T H R E E  | *
4    REM *  by Dean E. Russell                              |  June-July, 1983  | *
5    REM *                                                  ------------------- *
6    REM *  This is a good outline of a program that accepts Spreadsheet        *
7    REM *  type commands and processes them.  It isn't anywhere near           *
8    REM *  complete, but it is a good start.                                   *
9    REM *                                                                      *
10    REM ***********************************************************************
15    REM * An example of spreadsheet type input for   * BY: DEAN E. RUSSELL  *
20    REM * Apple /// Business Basic using the screen  * P. O. BOX 535        *
25    REM * control codes and window commands that are * EUREKA, KS. 67045    *
30    REM * available in the system.                   * PHONE 316-583/5512   *
35    REM ***********************************************************************
100    REM * Set array size variables, dimension array, set control string    *
105    REM * for cursor keys and escape key                                   *
110    columns=20:rows=15
115    DIM demoarray%(columns,rows)
120    control$=CHR$(8)+CHR$(21)+CHR$(10)+CHR$(11)+CHR$(27)
200    REM * Screen setup, has nothing to do with working part of program,    *
205    REM * sets up screen title and decribes workings of program            *
210    HOME:VPOS=24
215    PRINT"****************************************************************
       **************"
220    PRINT"*  AN EXAMPLE OF SPREADSHEET TYPE ENTRY USING BUSINESS BASIC ON
       THE APPLE /// *"
225    PRINT"****************************************************************
       **************"
230    interval=5:GOSUB 400
235    PRINT USING"79c";"SETTING UP PLEASE WAIT"
240    interval=8:GOSUB 400
245    PRINT"Example has a ";columns;" column by ";rows;" row array. The array
       is loaded to show operation":interval=1:GOSUB 400
250    PRINT"The last two digits in the number are the column and the first
       two are the row":interval=1:GOSUB 400
255    PRINT"You can: 1. Move around in the array with the arrow keys"
260    PRINT"          2. change the highlighted number to any positive integer
       up to 9999"
265    PRINT"          3. Exit program with the escape key";
300    REM * Fill demonstration array                                           *
305    FOR column=1 TO columns:FOR row=1 TO rows:demoarray%(column,row)=row*100
       +column:NEXT:NEXT
310    REM * Go to main program control                                         *
315    GOSUB 2000
320    REM * Clean up and exit program                                          *
325    NORMAL:TEXT:HOME:END
400    REM * Vertical scroll and delay subroutine for screen setup              *
405    FOR i=1 TO interval:PRINT CHR$(10);:FOR j=1 TO 5:NEXT:NEXT:RETURN
500    REM * Horizontal shift & control subroutine. If flag 1 is 1 then         *
505    REM * current window is for vertical control. Window and cursor is       *
510    REM * reset for horizontal control                                       *
515    IF flag1 THEN vhold= VPOS+1:hhold=hhold-2:flag1=0:WINDOW 25,5 TO 55,16:
       VPOS=vhold:HPOS=hhold
520    REM * Direction 1 sets left arrow control variables                      *
525    IF direction=1 THEN hboundry=1:w1=25:w2=29:hshift=5:ccounter=-1
530    REM * Direction 2 sets right arrow control variables                     *
535    IF direction=2 THEN hboundry=columns:w1=50:w2=54:hshift=251:ccounter=1
540    REM * Checks for right or left array column if true rings bell returns   *
545    IF column=hboundry THEN PRINT CHR$(7);:RETURN
550    REM * If not left column of window then move cell left                   *
555    IF direction=1 AND HPOS>5 THEN NORMAL:PRINT USING"5#";demoarray%(column,
       row);:HPOS= HPOS-2*( HPOS-hhold):hhold= HPOS:INVERSE:column=column+
       ccounter:PRINT USING"5#";demoarray%(column,row);:HPOS=hhold:RETURN
560    REM * If not right column of window then move cell right                 *
565    IF direction=2 AND HPOS<26 THEN NORMAL:PRINT USING"5#";demoarray%(column,
       row);:hhold= HPOS:INVERSE:column=column+ccounter:PRINT USING"5#";
       demoarray%(column,row);:HPOS=hhold:RETURN
570    REM * Select top row window variable and erase highlight                 *
575    toprow=row- VPOS+2:NORMAL:PRINT USING"5#";demoarray%(column,row);:HPOS=
       HPOS-5:hhold= HPOS
580    REM * Shift text left or right in window one column width and set        *
585    REM * window to print new column. Print new column                       *
590    PRINT CHR$(23);CHR$(hshift);:column=column+ccounter:WINDOW w1,5 TO w2,
       16:HOME:INVERSE:PRINT USING"3#,xx";column;:NORMAL:FOR rowpr=toprow TO
       toprow+9:PRINT USING"5#";demoarray%(column,rowpr);:NEXT
595    REM * Return to horizontal control window and highlight cell             *
600    PRINT CHR$(4);:INVERSE:PRINT USING"5#";demoarray%(column,row);:HPOS=
       hhold:RETURN
700    REM * Vertical shift & control subroutine. If flag 1 is 0 then current   *
705    REM * window is for horizontal control. Window and cursor is reset for   *
710    REM * vertical control.                                                  *
715    IF flag1=0 THEN vhold= VPOS-1:hhold=hhold+2:flag1=1:WINDOW 23,6 TO 55,
       15:VPOS=vhold:HPOS=hhold
720    REM * Direction 3 sets down arrow control variables                      *
725    IF direction=3 THEN vboundry=rows:vshift=10:rcounter=1
730    REM * Direction 4 sets up arrow control variables                        *
735    IF direction=4 THEN vboundry=1:vshift=11:rcounter=-1
740    REM * Checks for top or bottom array row if true rings bell & returns    *
745    IF row=vboundry THEN PRINT CHR$(7);:RETURN
750    REM * If not top or bottom row of window then move cell up or down       *
755    IF VPOS<10 AND VPOS>1 THEN NORMAL:PRINT USING"5#";demoarray%(column,row);
       :row=row+rcounter:HPOS=hhold:VPOS= VPOS+rcounter:INVERSE:PRINT USING"5#";
       demoarray%(column,row);:HPOS=hhold:RETURN
760    REM * Select left column window variable and erase highlight             *
765    presentcol=INT(( HPOS+5)/5):leftcol=column-presentcol+1:NORMAL:PRINT
       USING"5#";demoarray%(column,row);:HPOS=hhold
770    REM * Scroll to produce blank row. Print new row.                        *
775    PRINT CHR$(vshift);:row=row+rcounter:HPOS=1:INVERSE:PRINT USING"2#";row;
       :NORMAL:FOR columnpr=leftcol TO leftcol+5:PRINT USING"5#";demoarray%
       (columnpr,row);:NEXT
780    REM * Return to working cell and highlight.                              *
785    HPOS=hhold:INVERSE:PRINT USING"5#";demoarray%(column,row);:HPOS=hhold:
       RETURN
800    REM * Subroutine to stop main loop and end program                       *
805    repeat=1:POP:RETURN
900    REM * Change demonstration array variable up to four digits subroutine *
905    FOR enterlength=1 TO 4:work$=work$+enter$:demoarray%(column,row)=VAL
       (work$):PRINT USING"5#";demoarray%(column,row);:HPOS=hhold:GET enter$: IF
       ASC(enter$)<48 OR ASC(enter$)>57 THEN RETURN:REM change or enter variable
910      NEXT:PRINT CHR$(7);:RETURN
2000    REM * Main program control subroutine. Lines 2010 to 2040 set up the   *
2005    REM * spread sheet type display and loads the first display section    *
2010    REM * of the demonstration array.                                      *
2015    HPOS=0:VPOS=4:PRINT USING"79c";"HORIZONTAL ARRAY NO."
2020    WINDOW 19,6 TO 19,16:HOME:PRINT"VERTICAL"
2025    WINDOW 21,6 TO 21,16:HOME:PRINT"ARRAY NO."
2030    TEXT:INVERSE:VPOS=5:FOR m=1 TO 10:HPOS=23:VPOS= VPOS+1:PRINT USING"##";
        m;:NEXT
2035    VPOS=5:HPOS=25:FOR n=1 TO 6:PRINT USING"3#,xx";n;:NEXT:NORMAL
2040    WINDOW 25,6 TO 54,16:HOME:FOR row=1 TO 10:FOR column=1 TO 6:PRINT
        USING"5#";demoarray%(column,row);:NEXT:NEXT
2500    REM * Lines 2510 & 2515 set starting values for variables and the      *
2505    REM * vertical control window.                                         *
2510    row=1:column=1:flag1=1
2515    WINDOW 23,6 TO 55,15:VPOS=1:HPOS=3:work$=""
2520    REM * Highlight cell for row one, column one.                          *
2525    hhold= HPOS:INVERSE:PRINT USING"5#";demoarray%(column,row);:HPOS=hhold
2530    REM * Key input loop. Takes key input and sends it to the proper       *
2535    REM * subroutine for processing.                                       *
2540    FOR repeat=0 TO 1:repeat=-1:GET enter$
2545      key=ASC(enter$):IF key>47 AND key<58 THEN GOSUB 900
2550      direction=INSTR(control$,enter$):work$="":findsub1=INT((direction+
          1)/2)
2555      IF direction THEN ON findsub1 GOSUB 500,700,800
2560      NEXT:RETURN
```

Graphics Sketcher ///

by Stephen M. Dorman & L. A. Ditson

This program is written for the /// owner who is not necessarily an expert at programming. It is written for those who not only would like a useful graphics tool, but would also like a better understanding of Business Basic graphics programming. I also have taken steps to insure that the program is user friendly for the artists out there who might be more interested in the actual sketching than in programming.

With this program you can work with two buffers or "screens." The resolution of these two 'sketching pads' are selected at the start of the program. They are then set and cannot be changed unless you rerun the program. May I also suggest that you keep your graphics creations separated on different disks, one for each mode. This will prevent you from loading a picture of one resolution while the program mode is set for another. Other options selected by the user are the width of the sketching pen and the colors employed (if using color mode). You are provided with the ability to toggle between the two buffers, to save either picture, or to load an existing picture back into either buffer for further work. I have also added a small subroutine enabling the two buffers to be shown alternately at a speed determined by the user. If you would like to view two separate pictures together (in effect, making them transparent), simply set the time between buffers to 1. The two pictures will be toggled fast enough to create the illusion of overlaying. Unfortunately screen flicker at this speed will not yield a smooth transition between the two buffers, but it will do for comparison (lines at the same location, etc.)

There are a couple of details you need to take care of before running the program. They are called SOS.DRIVER and BGRAF.INV. You can find these two necessary modules on the Business Basic Boot Disc and can transfer them using the System Utilities Disc. SOS. DRIVER contains mini modules which allow you access to individual drivers. The one I use on line 15 is assigned reference number '#1' and is called .GRAFIX. This command is not essential to this program but would prove necessary if you should expand the program, for instance, printing text on the graphics screens. An example: PRINT #1; "A" will print the letter "A" at the current pen position. On line 30, BGRAF.INV is calling an assembly language routine which allows you the use of all 'PERFORM' commands. This, in a sense, is the workhorse of the program.

Line 49 begins the instructions of the pen movement. You will notice that numeric keypad on the /// lends itself well to easy pen movement. The number "5" will convert the pencolor to the background color (expressed as fillcolor in the program).

Line 310 is the command to SET the mode that you will be using. There are four modes (0 through 3). The details of these modes are in the program listings 170-200. There is one small problem however. If you are using MODE 1, you might find some distortions called 'Color Anomalies' occurring when you cross colors. (See page 92 of STANDARD DEVICE DRIVERS MANUAL)

Line 320 will FILL the PORT or erase the graphics screen if you will. When you "invoke" the graphics module (BGRAF.INV), SOS reclaims memory ordinarily used by program lines and reserves it for graphics. There is, however, data left behind that shows up as garbage on the graphics screens. The FILLPORT will fill the screens of both buffers with the color that you select as the fillcolor. Line 330 does the same to the buffer not being used at that moment and Line 340 will reset the graphics mode back to the buffer you chose.

Lines 420-620 are the main branching routines of the program, controlling either the menu selection, or the actual sketching on the screens. The control characters are for program control. They are self-explanatory. Make sure the CONTROL key is firmly down while typing the letter.

The first subroutine is found between lines 1000-1015 and provides for selection of normal or inverse for the black and white modes. Line 1005 is the error handling routine. This limits the choices to 1 and 2.

The second subroutine (lines 1500-1595) will allow for pencolor and fillcolor selection.

The third subroutine (lines 200-2110) is the MENU selection routine. This is self explanatory.

The subroutine from 2500-2530 simply turns the dot on and off. Here is an example of the SWAP command, a command which will SWAP two variables. I found my first use of this command while writing this program. It eliminates as extra step usually needed for switching variables, which, in this case, are the pencolor and fillcolor.

The ERASE routine is contained in lines 2600-2630. We meet up again with "FILLPORT" which, as you recall, clears the screen with the current fillcolor via the BGRAF.INV assembly module.

The page flipping routine is found from line 2800 to 2950. The j and t loops control the speed between flips. You will notice the use of "ON KBD". This allows the loop to stop by detecting that a key from the keyboard has been pressed.

The LOAD subroutine is used to load a picture from a disk. The pictures are stored as "FOTO" files. You must remember in which mode the picture was created. You may clue yourself by appending, for example, .m1 after the FOTO name (MYPICTURE.M1 for mode 1). But as I said before it is much better to keep them separate on an entire disk, with the disk labeled.

The SAVE routine (lines 3000-3040) will save the picture to disk. The size of the files that you SAVE will depend on the mode that you are using. Mode 0 files are 17 blocks, and other modes are 33 blocks. (That's quite a lot considering only 276 blocks are available on the disk). (Thanks to the editor for the four extra blocks.) If you will be using the SAVE and LOAD feature, may I suggest you use an entirely blank diskette in Drive 2. This will allow you to store your creations and leave the MASTER diskette free. If you have only one disk drive, you will change disks after selecting CONTROL-S. Change the disks, type SAVE pathname <RETURN> , and after the disk stops spinning, replace your SAVE disk with the program disk.

In the LOAD and SAVE subroutines, you will enter the PATHNAME of the file you wish to load or save: eg. .d2/MYPICTURE. If you make a mistake for any reason during these two routines, you will be returned to the graphics screen. If this happens, be sure your pathname is correct, and that your disk isn't full.

The QUIT routine (lines 3050-3080 and 3900-3940) is next. When you leave the program the computer has reserved 32K of space if you are using modes 1-3 for graphics. Also the BGRAF.INV takes up a lot of room. Therefore I have inserted the command PERFORM RELEASE to return this space back to you. "INVOKE" is similar. This releases space that the BGRAF.INV took up. If you exit the program illegally (eg. RESET-TEXT), and run the program again, you will eventually get some strange results: SOS CALL ERRORS, OUT OF MEMORY etc. This is because the space is not released before it is used again. Control-Q will allow you to QUIT formally, and free up this space.

The last routine which I will discuss is in LINES 3100-3250. This routine allows you to change the WIDTH of the pen strokes for the graphics display, and affects the axis of your choice (remember you have to use 0 width both vertically and horizontally to get the pen back to a dot).

Below is a synopsis of some of the commands in Graphics. (I have left out SETCTAB, and TRANSFER options), as I haven't had time to work with these enough to understand them. Preceed all commands with PERFORM.

(the % is a means of passing a variable to the subroutines. Do not confuse it with % for integer variables. Do not omit it when using these commands.)

(PERFORM) INITGRAFIX: not used in this program. This will reset the screen, and you will find yourself at 0,0 (lower left of the screen).

GRAFIXMODE: sets the graphics mode (0-3). You may use a variable or number: eg: PERFORM GRAFIXMODE (%1, %2) OR (%A, %B) The first number selects MODE, the second selects the buffer to use.

GRAFIXON: Turns ON the graphics screen

VIEWPORT: You can set the graphics screen to your own dimensions. (Viewport graphics screen in this case)

PENCOLOR: Set the color you are drawing with.

FILLCOLOR: Set the background color.

MOVETO: Will move the DOT to the coordinates you set.

MOVEREL: Will move a distance RELative to the current dot position.

DOTAT: Put a DOT were the cursor is. (Allow you to see where you are).

DOTREL: Put a DOT (%X, %Y) at a position from the current position.

LINETO: Draws a line from the current dot position to the coordinates you give it.

LINEREL: Draws a line RELative to the current dot position. (Use the change width of the pen in this program.)

FILLPORT: Erase the screen in the current FILLCOLOR

GSAVE.: Save the current picture on the current mode and buffer. Notice the "." after the GSAVE command. This command may be direct:)PERFORM GSAVE.".D2/MYPICTURE" or indirect; name $ ".d2/MYPICTURE": PERFORM GSAVE.name$

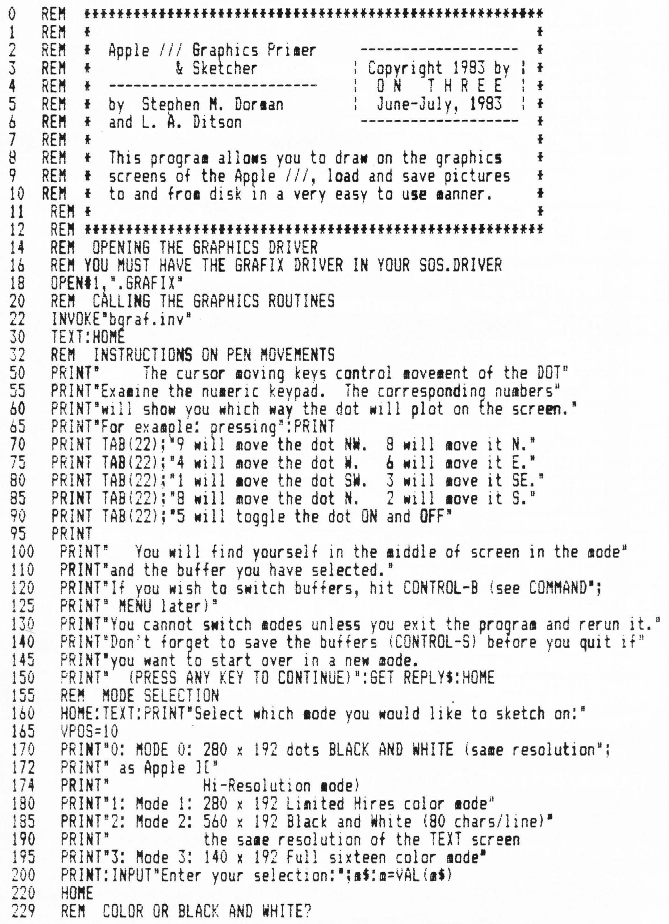
GLOAD.: The same rules apply to GLOAD as apply to GSAVE.

RELEASE: Releases space used by the Graphics routine.

So there is it. There is one note I would like to add. If you do not have an RGB color monitor, and do have a Video home recorder and color television, there are other ways to get color from the Apple ///. It will require that you have a DB-15 plug (ask your dealer) with pins 12 and 13 only. These pins may be connected to a two lead cable suitable to connect to a Video Tape Recorder Camera input. The VTR will then act as the RF modulator and you can see the color. This color is limited by the resolution of your television set, but still produces some nice results. This plyg is also useful for using the Apple /// in Emulation mode for COLOR programs. You will not, however, be able to read the 80 column screen using this technique, because the resolution of the television is the limiting factor. The characters will look fuzzy.

So happy sketching! (Now if I can only figure out how to dump these creations to my printer!) ///

Graphics Sketcher ///: Program Listing



```
0    REM *******************************************************
1    REM *                                                     *
2    REM *  Apple /// Graphics Primer         ------------------- *
3    REM *          & Sketcher               ! Copyright 1983 by ! *
4    REM *  -------------------------        !  O N   T H R E E  ! *
5    REM *  by  Stephen M. Dorman            !  June-July, 1983  ! *
6    REM *  and L. A. Ditson                  ------------------- *
7    REM *                                                     *
8    REM *  This program allows you to draw on the graphics    *
9    REM *  screens of the Apple ///, load and save pictures   *
10   REM *  to and from disk in a very easy to use manner.     *
11    REM *                                                    *
12    REM *****************************************************
14    REM  OPENING THE GRAPHICS DRIVER
16    REM YOU MUST HAVE THE GRAFIX DRIVER IN YOUR SOS.DRIVER
18    OPEN#1,".GRAFIX"
20    REM  CALLING THE GRAPHICS ROUTINES
22    INVOKE"bgraf.inv"
30    TEXT:HOME
32    REM  INSTRUCTIONS ON PEN MOVEMENTS
50    PRINT"     The cursor moving keys control movement of the DOT"
55    PRINT"Examine the numeric keypad.  The corresponding numbers"
60    PRINT"will show you which way the dot will plot on the screen."
65    PRINT"For example: pressing":PRINT
70    PRINT TAB(22);"9 will move the dot NW.  8 will move it N."
75    PRINT TAB(22);"4 will move the dot W.   6 will move it E."
80    PRINT TAB(22);"1 will move the dot SW.  3 will move it SE."
85    PRINT TAB(22);"8 will move the dot N.   2 will move it S."
90    PRINT TAB(22);"5 will toggle the dot ON and OFF"
95    PRINT
100    PRINT"   You will find yourself in the middle of screen in the mode"
110    PRINT"and the buffer you have selected."
120    PRINT"If you wish to switch buffers, hit CONTROL-B (see COMMAND";
125    PRINT" MENU later)"
130    PRINT"You cannot switch modes unless you exit the program and rerun it."
140    PRINT"Don't forget to save the buffers (CONTROL-S) before you quit if"
145    PRINT"you want to start over in a new mode.
150    PRINT"  (PRESS ANY KEY TO CONTINUE)":GET REPLY$:HOME
155    REM  MODE SELECTION
160    HOME:TEXT:PRINT"Select which mode you would like to sketch on:"
165    VPOS=10
170    PRINT"0: MODE 0: 280 x 192 dots BLACK AND WHITE (same resolution";
172    PRINT" as Apple ]["
174    PRINT"                Hi-Resolution mode)
180    PRINT"1: Mode 1: 280 x 192 Limited Hires color mode"
185    PRINT"2: Mode 2: 560 x 192 Black and White (80 chars/line)"
190    PRINT"                the same resolution of the TEXT screen
195    PRINT"3: Mode 3: 140 x 192 Full sixteen color mode"
200    PRINT:INPUT"Enter your selection:";m$:m=VAL(m$)
220    HOME
229    REM  COLOR OR BLACK AND WHITE?
```

```
230  IF(m=0) OR(m=2) THEN GOSUB 1000
240  IF(m=1) OR(m=3) THEN GOSUB 1500
250  IF(m<0) OR(m>3) THEN GOTO 160
260  HOME
265  REM  INITIAL BUFFER SELECTION
270  VPOS=10:HPOS=25:INPUT"Which buffer (1 or 2):";b
280  IF(b<1) OR(b>2) THEN GOTO 260
290  IF b=1 THEN a=2:ELSE a=1
300  REM  CLEARING BOTH BUFFERS
310  PERFORM grafixmode(%m,%b)
320  PERFORM fillport
330  PERFORM grafixmode(%m,%a):PERFORM fillport
340  PERFORM grafixmode(%m,%b)
350  HOME
355  REM  OPTIONS MENU
360  GOSUB 2000
365  REM  INITIAL PEN PLACEMENT
370  IF(m=0) OR(m=1) THEN y=96:x=140
380  IF m=2 THEN y=96:x=290
390  IF m=3 THEN y=96:x=70
395  ON ERR GOSUB 3960
400  PERFORM grafixon
410  PERFORM dotat(%x,%y):PERFORM LINETO(%(x+dx),%(y+dy))
420  REM  ACTUAL PLOTTING AND CONTROL OF PROGRAM ROUTINE
430  GET a$:PERFORM lineto(%(x+w),%y)
440  IF a$="9" THEN x=x+1:y=y+1:GOTO 410
450  IF a$="1" THEN x=x-1:y=y-1:GOTO 410
460  IF a$="7" THEN x=x-1:y=y+1:GOTO 410
470  IF a$="8" THEN y=y+1:GOTO 410
480  IF a$="4" THEN x=x-1:GOTO 410
490  IF a$="6" THEN x=x+1:GOTO 410
500  IF a$="2" THEN y=y-1:GOTO 410
510  IF a$="3" THEN x=x+1:y=y-1:GOTO 410
520  IF a$="5" THEN GOSUB 2500
525  IF a$=CHR$(2) THEN GOSUB 3200
530  IF a$=CHR$(3) THEN GOSUB 1500:PERFORM GRAFIXON:REM CONTROL-C
540  IF a$=CHR$(5) THEN GOSUB 2600:REM CONTROL-E
550  IF a$=CHR$(12) THEN GOSUB 2700:REM CONTROL-L
560  IF a$=CHR$(13) THEN GOSUB 2000:REM CONTROL-M
570  IF a$=CHR$(16) THEN GOSUB 2800:REM CONTROL-P
580  IF a$=CHR$(17) THEN GOSUB 3050:REM CONTROL-Q
590  IF a$=CHR$(19) THEN GOSUB 3000:REM CONTROL-S
600  IF a$=CHR$(23) THEN GOSUB 3100:REM CONTROL-W
610  GOTO 410
620  INVOKE:END
999  REM  BLACK AND WHITE PEN COLOR SELECTION ROUTINE
1000  HOME:VPOS=10
1010  PRINT"You have two choices in the  B&W modes:"
1020  PRINT"    1).  NORMAL  (white pen color/black fillcolor)"
1030  PRINT"    2).  INVERSE (black pen color/white fillcolor)"
1040  INPUT"    Select (1 or 2):";reply$
1050  IF(VAL(reply$)<1) OR(VAL(reply$)>2) THEN PRINT CHR$(7):GOTO 1001
1060  IF reply$="1" THEN f=0:p=15
1070  IF reply$="2" THEN f=15:p=0
1080  PERFORM fillcolor(%f):PERFORM pencolor(%p)
1090  PERFORM fillport
1100  RETURN
1499  REM  PEN COLOR SELECTION ROUTINE
1500  IF(m=0) OR(m=2) THEN RETURN
1504  HOME:TEXT:WINDOW 15,10 TO 80,24:REM SCREEN FORMAT
1505  HOME:PRINT"Here are the choices for pencolor and fillcolor"
1510  PRINT"0= Black    1=Magenta     2=Dark Blue     3=Purple"
1520  PRINT"4= D. Green 5=Grey 1      6=Med. Blue     7=Lt. Blue"
1530  PRINT"8= Brown    9=Orange      10=Grey 2       11=Pink"
1540  PRINT"12=Green    13=Yellow     14=Aqua         15=White"
1550  INPUT"Pencolor: ";p
1560  INPUT"Fillcolor: ";f
1565  IF(p>15) OR(f>15) THEN PRINT CHR$(7):GOTO 1500
1570  TEXT:HOME
1575  PERFORM fillcolor(%f)
1580  PERFORM pencolor(%p)
1600  RETURN
1999  REM  MAIN SELECTION MENU
2000  HOME:TEXT:PRINT"CONTROL SUMMARY"
2010  PRINT"CONTROL-B to CHANGE BUFFER BEING USED"
2020  IF(m=0) OR(m=2) THEN GOTO 2040
2030  PRINT"CONTROL-C to CHANGE PEN OR FILL COLORS"
2040  PRINT"CONTROL-E to ERASE THE BUFFER CURRENTLY IN USE"
2050  PRINT"CONTROL-L to LOAD A NEW PICTURE INTO PRESENT BUFFER"
2060  PRINT"CONTROL-M (or RETURN) to SEE THIS MENU AGAIN"
2070  PRINT"CONTROL-P to PAGE FLIP BETWEEN GRAPHICS BUFFERS (DISPLAY";
2075  PRINT" ALTERNATE  BUFFER"
2080  PRINT"CONTROL-Q to RETURN TO QUIT THE PROGRAM"
2090  PRINT"CONTROL-S to SAVE PICTURE TO DISK"
2100  PRINT"CONTROL-W to CHANGE WIDTH OF PLOTTER PEN"
2110  PRINT"Make sure you go back to the sketcher before using these";
2120  PRINT" commands."
2130  PRINT"Hit any key to go to the SKETCHER";:GET reply$
2140  IF(reply$<CHR$(32)) AND(reply$<>CHR$(13)) THEN GOTO 2000
2150  PERFORM grafixon
```

```
2200  RETURN
2499  REM   SWITCH THE PENCOLOR AND FILLCOLOR (WILL TURN DOT OFF & ON)
2500  SWAP p,f
2510  PERFORM pencolor(%p)
2520  PERFORM fillcolor(%f)
2530  RETURN
2599  REM  ERASE CURRENT BUFFER
2600  HOME:TEXT:PRINT"This will erase the picture currently in use.";
2610  INPUT" (CONTINUE? y/n)";reply$
2620  IF (reply$="y") OR (reply$="Y") THEN PERFORM fillport
2630  PERFORM grafixon
2640  RETURN
2699  REM  LOAD A PICTURE INTO THE BUFFER IN USE
2700  HOME:TEXT
2710  INPUT"Enter the file name of picture to load: ";name$
2720  IF LEN(name$)<1 THEN GOTO 2710
2730  PRINT"You were using buffer";
2740  PRINT b;" last"
2750  INPUT"which buffer 1 or 2 to load it into?:";c
2760  IF (c<1) or (c>2) THEN PRINT CHR$(7):GOTO 2750
2770  PERFORM grafixmode(%m,%c)
2780  PERFORM gload.name$
2790  PERFORM grafixmode(%m,%c)
2792  PERFORM grafixon
2795  RETURN
2799  REM  PAGE FLIPPING ROUTINE-- VARIABLE SPEED
2800  HOME:TEXT:PRINT"This subroutine will flip very quickly between the two"
2810  PRINT"buffers.  The speed of this flipping can be controlled, and"
2820  PRINT"may be fun to experiment with for basic animation"
2830  PRINT"The value you give will be the end of a do loop"
2840  PRINT"Try 125.  This is a good speed to allow flipping."
2850  PRINT"if you input 1, the flip will be so fast as to almost display"
2860  PRINT"both buffers simultaneously. Hit any key to stop the animation"
2870  INPUT"Enter value greater than 0: ";k
2880  IF k<1 THEN GOTO 2870
2890  PERFORM grafixon
2900  FOR i=1 TO 10000
2905    ON KBD i=10000:RETURN
2910    PERFORM grafixmode(%m,%a)
2915    PERFORM grafixon
2920    FOR j=1 TO k:NEXT j
2930    PERFORM grafixmode(%m,%b)
2935    PERFORM grafixon
2939    FOR t=1 TO k:NEXT t
2940    NEXT i
2950  RETURN
2999  REM  FOTO FILE SAVE ROUTINE-- ENTER PATHNAME
3000  HOME:TEXT
3010  INPUT"Enter name under which to save picture:";name$
3015  IF LEN(name$)<1 THEN GOTO 3010
3020  PERFORM gsave.name$
3030  PERFORM grafixon
3040  RETURN
3045  REM  QUITTING ROUTINE: THERE HAS TO BE A WAY OUT
3050  HOME:TEXT:PRINT"This will allow you to quit"
3060  PRINT"Do you want to quit without saving your picture? (y/n) ";
3065  get reply$
3070  IF reply$="Y" OR reply$="y" THEN GOSUB 3900
3075  PERFORM grafixon
3080  RETURN
3099  REM  PLOTTER PEN WIDTH CHANGING ROUTINE-- VERY INTERESTING EFFECTS!!
3100  HOME:TEXT
3101  INPUT"How wide would you like the pen to be: ";penwidth
3102  HOME:TEXT:PRINT"Do you wish to change the (H)orizontal or (V)ertical"
3103  PRINT"pen width or (B)oth";:GET reply$
3105  IF penwidth<0 THEN GOTO 3102
3107  HOME:PERFORM grafixon
3110  IF reply$="B" OR reply$="b" THEN dx=penwidth:dy=penwidth
3120  IF reply$="H" OR reply$="h" THEN dx=penwidth
3130  IF reply$="V" OR reply$="v" THEN dy=penwidth
3140  RETURN
3199  REM  BUFFER SWITCH ROUTINE
3200  SWAP a,b
3210  PERFORM grafixmode(%m,%b)
3230  PERFORM grafixon
3240  PERFORM dotat(%x,%y)
3250  RETURN
3899  REM  THIS QUITTING ROUTINE WILL FREE UP MEMORY USED BY GRAPHICS
3900  FOR I=1 TO 3
3905    PERFORM RELEASE:REM THIS FREES UP THE GRAPHICS BUFFERS
3910    NEXT I
3920  INVOKE:REM FREES UP SPACES USED BY THE 'BGRAF.INV'
3930  CLOSE#1:REM CLOSING GRAPHICS DRIVER.
3940  END:REM FINISHED!!
3960  TEXT:HOME:PRINT"ERROR #"; ERR; " IN LINE #"; ERRLIN
3970  FOR WASTE=0 TO 250
3980  REM DO NOTHING BUT WASTE SOME TIME AND THEN RETURN
3990  NEXT WASTE
3999  PERFORM GRAFIXON:RETURN
4000  REM  DAS ENDE
```

/// The Hard Way

Color and the Emulation Mode

by George Oetzel

There is a widespread rumor that the Apple /// does not support color in the Emulation Mode. Like many rumors, there is some truth to this one. The Apple /// doesn't generate the RGB color signals in Emulation Mode, so an RGB color monitor is a poor choice if you want to run Apple)(color programs. The somewhat cryptic note about this situation on page 144 of the Owner's Manual has been interpreted very commonly to mean that the Apple /// doesn't generate color in the Emulation Mode at all. That interpretation is incorrect.

You can use an NTSC color monitor or an RF modulator and a color TV set to obtain color images in Emulation Mode. The Amdek Color I is an example of an NTSC color monitor. Most of the NTSC monitors lack the resolution to be used regularly with 80-column-wide text displays, so you don't want to replace your green screen for normal use in the Apple /// native mode. An NTSC monitor and a B/W monitor together cost less than a good RGB monitor, and it seems that this is the best for the Apple ///. No sacrifice of Apple /// color is involved in the use of an NTSC monitor, because the Apple /// hi-res color mode is identical to the corresponding Apple)(mode.

If you purchased your Apple /// in 1982, there is a good chance that the color video signal has been routed to the RCA connector labelled "B/W VIDEO" on the back of the Apple ///. The Owner's Manual also calls the B/W Video Port and describes the video connection on page 135. The small change required to route the color signal to this connector apparently was made by Apple without fanfare or documentation changes.

If you find that your B/W Video Port is really restricted to black and white, or if you want to use two monitors, a connection to the Color Video port ranks as one of the world's simplest electronic construction projects. If you have an NTSC color monitor, it takes about six dollars worth of parts and ten minutes work. If you want to use your color TV, you need an RF modulator designed for the Apple)(. The construction work needed to attach the modulator to the Color Video Port is still in the ten-minute category. If you are uncomfortable with electronic projects that involve soldering, you don't have to worry that it is a major imposition to ask a friend for help with this one.

The connector for the color video signal is a 15-pin male "D" connector. It is usually listed in catalogs with the numerical designation DA15P, typically in addition to the distributor's catalog number. You want a connector that is intended for solder connections, not ribbon cable. At a local parts house in Palo Alto, the connector was $2.70.

The plastic hood that serves as a wire clamp and handle is usually called a DA15H, and cost $2.25 at the same place. If you live in a large city, you can probably obtain these parts at a local electronics supply shop. Several mail order houses advertising lower prices are in the back pages of BYTE, but shipping costs will add to the mail-order price.

If you have a color monitor that uses an RCA connector (like the B/W connector on the Apple /// and many hi-fi connectors), the only other item you need is an audio cable that has an RCA male connector on one end and bare wires on the other end. These are carried by many electronic and computer shops. For example, Radio Shack usually has them. A three foot cable at the same Palo Alto shop was $1.09. If the dealer is out of the cables with wires on one end, just buy one with connectors on both ends. Cut the connector off of one end. Cut an inch or so of the outer insulation and separate the shield wires into a bundle that will fit into one of the solder terminals on the connector. Strip some of the insulation from the center conductor. Trim the bare center wire so that it is just long enough to reach the bottom of the solder pins on the connector.

The picture on page 132 of the Apple /// Owner's Manual shows the numbering of the pins on the DA15P connector. The view shown matches what you see when you set the connector in position to do the soldering. If you have good eyes and good light, you will probably find that there are small, raised numbers in the plastic part of the connector. Pin 12 has the color video signal. Solder the center conductor of your cable to pin 12. Solder the shield to either pin 6 or pin 13. Both are grounds. Make sure that the shield can't touch any of the other pins. Remember Murphy's law: "If anything can go wrong, it will". If it is possible for that ground wire to touch one of the other pins, it's going to do it sooner or later. Plastic electrical tape is one way to prevent disaster. If you want to be a little fancier, and surer, perhaps your electronic supplier has some heat-shrinkable insulating tubing that you can shrink around the connection.

If you already have an RGB monitor, you may prefer a color TV for occasional use of Emulation Mode color. The most difficult part if you adopt this option comes at the very beginning. You have to find an RF modulator designed for the Apple)(. With falling prices and greater availability of color monitors, fewer people use the TV these days, so the local computer shop may be out of modulators. M & R Enterprises makes the "Sup 'R' Mod" RF modulator and you should be able to get one at your local computer store.

If you want to use the modulator with an Apple)(also, you will also have to pick up an in-line male connector to match the one on the modulator. They seem to come only in the chassis-mount variety, so the connection to the DA15P with short wires will have to be butressed with some sort of electrical tape. For a better Apple /// solution, just cut the little four-pin connector off the modulator wires. Before you cut, make sure that you can identify which wire to solder to which pin of the DA15P.

The table below identifies the modulator connections, as given by the Apple)(Reference Manual and the Sup 'R' Mod instruction sheet.

Apple)(Pin Number	Sup 'R' Mod Wire Color	DA15P Pin Number	Description
1	Black	6 or 13	Ground
2	Brown	12	Color Video
3	Red	7	-5 volts
4	Orange	8	+12 volts

Your modulator could have different wire colors. (The manufacturer may have gotten a bargain on a mile of purple wire.) Apple won't change their end, so you can still figure out the connections. The holes in the in-line connector are offset from the center, to prevent you from plugging it in backwards. Hold the connector with the thin side down, and pin 1 will be at the right-hand end. Strip a little insulation from the end of each wire and solder it to the approach pin of DA15P.

Follow these instructions, and you can have color displays with your favorite Apple)(programs, as well as in the Apple /// color modes. ///

JOYstick for][GAMES

by Elwynn Taylor

My children soon discovered that most Apple)(games would run on our Apple /// (and ON THREE tells us that the Micro-Sci Gameport /// will run them all). I was soon at the local dealership to obtain a "joystick" and returned home the proud owner of a "Cursor ///". I don't know if the serial number 104 on the device made me a pioneer or not, but there were certainly pioneer hardships in getting it to reach its potential of opening the world of Apple)(games in my household. On first plug-in, we found that one trigger worked and we could move the cursor up and down but not left and right. Next attempt: move the plug to joystick port B. Port B resulted in effective left and right movement but no vertical action. Clearly there was a basic incompatibility here: would we have to wait for Apple /// games or could we do something to get it to work?

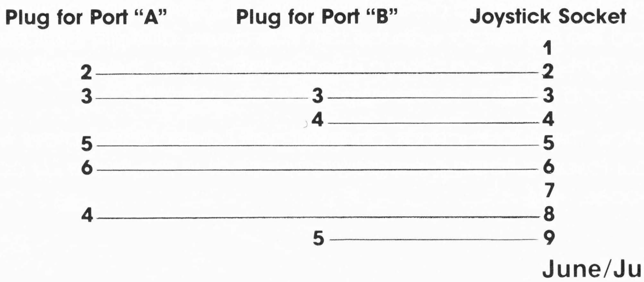
It became obvious that various games respond differently to our new joystick, but most seemed to have vertical action assigned to port A and horizontal action assigned to port B. So I breadboard-wired a splitter to plug my joystick to certain pins of ports A and B simultaneously. It WORKED!! We could play many games with our joystick. A few games required alternate connections and the breadboard made that easy (and also easy to misconnect wires)! There were a few games that just would not respond to any port configuration I could come up with. The bread-board remained important but a firm wired version has met the most popular needs of our household.

The hardware required to adapt the port plug consisted of TWO 9 pin plug (male) and ONE nine pin socket (female) and a foot (more if desired) of multi-conductor cable. I used 7 conductor but more would be OK. I used AMP 81-18 plugs and an AMP 81-12 socket (they were as rare as hen's teeth in our city but arrived within 10 days). Soldering together was not a great problem and we had a functional plug splitter/adapter.

The configuration I settled upon as most general, connected pins 2, 4, 3, 5, and 6 of the "A" port to socket pins 2, 3, 8, 5, and 6, respectively. The remaining male plug (port B) used pins 3, 4, and 5 connected to socket pins 3, 4, and 9, respectively.

I did find that the cursor would drift in a neutral position. This was corrected by removing the back from the "Cursor ///", loosening the set screws that connect the stick to potentiometers, and slightly adjusting them so that the ports do not have power applied in the neutral position. I'm told that the later models have an external adjustment. ///

Figure 1: Connections for a plug adapter that allows the "Cursor ///" joystick to function with numerous Apple)(games in Emulation Mode. The port assignments as given in the Apple /// Owner's Guide and the joystick sample circuit are used to make pin connection assignments.



Continued from page 28

```
680   DATA"065065065065073073085099065","067035050030012028054099065"
690   DATA"065099054028008008008008008","127062004008024048032126127"
700   DATA"028062099099099099099062028","012044076012012012012012127"
710   DATA"062099099003006012024024063","127006008016030003003031062"
720   DATA"012028052100127063004004004","031030032032062031067035030"
730   DATA"015029048032126099099035031","127063002006012012024048048"
740   DATA"030051051051030051051051030","060098099099063002006014124"
750   REM
760   REM ************************************************************
770   REM * READ IN LETTER/NUMBER RECIPES STARTING AT 64 TO 92; A TO Z,  *
780   REM * (A=ASCII 65, Z=ASCII 90) AND 48 TO 57; 1 TO 0.(1=ASCII 48,    *
790   REM * 0=ASCII 57)                                                   *
800   REM ************************************************************
810   REM
820   FOR I=1 TO 26
830     READ RECIPE$(I+64)
840     NEXT I
850   FOR I=1 TO 10
860     READ RECIPE$(I+47)
870     NEXT I
880   REM ************************************************************
890   REM * PRINT USER INSTRUCTIONS ETC...                                *
900   REM ************************************************************
910   HOME:VPOS=12
920   PRINT"      THIS PROGRAM IS CALLED HEADFIRST.  IT PRINTS A HEADING UPTO"
930   PRINT"13 CHARACTERS LONG ON A PRINTER(MX-100)."
940   PRINT
950   PRINT"ENTER A 1 TO END, PRESS ANY OTHER KEY TO CONTINUE"
960   GET ANS$
970   IF ANS$="" THEN 960
980   IF ANS$="1" THEN END
990   HOME:VPOS=12
1000   PRINT" PLEASE ENTER NAME FOR HEADING, LESS THAN 13 CHARACTERS."
1010   REM ****************************************************
1020   REM * INPUT NAME$ AND VALIDATE LENGTH AND STORE CHARACTERS.*
1030   REM ****************************************************
1040   INPUT NAME$
1050   STRL=LEN(NAME$)
1060   IF STRL>13 THEN 1000
1070   FOR I=1 TO STRL
1080     TMP$=MID$(NAME$,I,1)
1090     NUM=ASC(TMP$)
1100     REM ********************************************************
1110     REM * IF THIS CHARACTER IS BLANK ADD A SPACE TO THE PRINT ARRAY. *
1120     REM ********************************************************
1130     IF NUM<>32 THEN 1210
1140     FOR K=1 TO 9
1150       TDPA$(K)=TDPA$(K)+"        ": REM 8 Spaces between quotes
1160       NEXT K
1170     GOTO 1380
1180     REM *********************************************************
1190     REM * CHECK TO SEE IF CHARACTER IS LETTER OR NUMBER, IF NOT ERROR.*
1200     REM *********************************************************
1210     IF NUM<48 OR NUM>122 THEN 1680
1220     IF NUM<97 AND NUM>90 THEN 1680
1230     REM ***************************************
1240     REM *  CONVERT ALL CHARACTERS TO UPPERCASE.  *
1250     REM ***************************************
1260     IF NUM>96 THEN NUM=NUM-32
1270     REM ********************************************************
1280     REM * READ INSTRUCTION LIST, GET CHARACTER SLICE, AND FILL PRIN- *
1290     REM * TING ARRAY.                                                *
1300     REM ********************************************************
1310     N=0
1320     FOR J=1 TO 25 STEP 3
1330       TEMP$=MID$(RECIPE$(NUM),J,3)
1340       NVAL=VAL(TEMP$)
1350       N=N+1
1360       TDPA$(N)=TDPA$(N)+SLICE$(NVAL)+"      ": REM 6 spaces
1370       NEXT J
1380     NEXT I
1390   REM ********************************************************
1400   REM * OPEN PRINTER FILE FOR EPSON MX-100.                         *
1410   REM ********************************************************
1420   OPEN#1,".PRINTER"
1430   FOR I=1 TO 9
1440     PRINT#1TDPA$(I)
1450     NEXT I
1460   REM ********************************************************
1470   REM * PRINT 2 LINE FEEDS TO PREPARE FOR NEXT LINE.                 *
1480   REM ********************************************************
1490   PRINT#1;CHR$(10)
1500   PRINT#1;CHR$(10)
1510   REM ***********************
1520   REM * CLEAR PRINTING ARRAY *
1530   REM ***********************
1540   FOR I=1 TO 9
1550     TDPA$(I)="": REM No spaces between the quotes
1560     NEXT I
1570   REM *********************************
1580   REM * OFFER USER OPTION TO RUN AGAIN *
1590   REM *********************************
1600   HOME:VPOS=12
1610   PRINT"TO PRINT ANOTHER NAME ENTER A 1"
1620   GET ANS$
1630   IF ANS$="1" THEN 1000
1640   END
1650   REM **********************
1660   REM * PRINT ERROR MESSAGE *
1670   REM **********************
1680   HOME:VPOS=12
1690   PRINT"YOU'VE ENTERED A CHARACTER THAT THIS PROGRAM DOESN'T RECOGNIZE"
1700   PRINT" USE ONLY A-Z AND 0 TO 9"
1710   GOTO 1000
```

Headfirst

by Michael Gardner

I named this program HEADFIRST because it prints a heading for program listings, etc. and the heading is the first thing out of the printer, hence HEADFIRST. I wrote it for a different reason. I wanted to have some useful kind of program to show friends when they came over. Normally I would rave about my spreadsheet program and all of its applications. But I'd get mired down in explanations and unreal examples when it came time to demonstrate it. "Suppose I wanted to figure out the before tax annual internal rate of return on my toaster..."

I remembered a computer show I went to where a printer company was showing off all the features of their printers. The particular printer they had could print in block letters about a foot high. Now how do you suppose all of the people that stopped at the printer display felt when they left the show with their name printed in letters about a foot high? I've still got mine.

So if you're still looking for a program that is extremely visible and easy to use you might be interested in this one. Even if you're interested in practical applications, this is a useful utility program. HEADFIRST can be used to label program listings, make signs to tape on the fridge, and print embarrassing messages to first time computer users. The letters it prints are somewhat short of a foot high, but they are readable from across a room. See figure 1.

The program is written to run on an Apple /// with an Epson MX-100. The listing is also fluent with REM statements to guide the reader through it.

Overview

Input is examined one character at a time. As each character is examined the program looks through its data statements to find the instructions for printing a large version of that character. It then stores the large character in the printing array. Once the entire name is examined and stored in the printing array, the printing array is sent to the printer. After the name is printed the program will ask the user if he or she wants to print another name and either "loop" back to run again or end.

Nuts and Bolts

Each of the large letters printed is a 9 x 7 matrix made up of nine seven-character-long horizontal slices. The program prints different letters by combining different slices. So to print a particular letter, the program has to know which slices to put together to form that letter. That's why there are two distinct sets of data statements. One set contains all the different possible slices, and the other contains the "recipes" for putting those slices together to form a letter.

The slices themselves are combinations of printed and nonprinted spaces seven characters long. I chose the "@" for the printed space because it has more ink per area than most of the other characters, and consequently more contrast with the existing white space. As for the arrangement or sequence of the slices, I used the binary number system to organize them. In other words, if the character "@" was replaced by the number "1", and the empty space " " was replaced by the number "0", the first group of data statements would look like a listing of eight digit binary numbers from 0 to 127. Which is exactly what it is. By doing it this way, each slice has a label, the binary number it represents. This label also corresponds to the index of the array SLICE$(I) where the binary representation is stored. (lines 470-490)

Now that I have labelled the slices and I can tell them by number, I've got to know which ones to stack up to create a given letter. That's the purpose of the second set of data statements. Each element in the second set is a recipe for constructing a letter. The first element is the recipe for the letter "A", the second the letter "B", and so on. To identify the recipes, I've made them correspond to the ASCII value of the letter they represent. To do this they are read into the aray RECIPE$(I) where I is the ASCII value. The current set-up requires two "FOR-NEXT" loops to do this, the first incrementing from 65 to 90 (Ascii values for A to Z; lines 810-830), and the second incrementing from 48 to 57 (Likewise, Ascii values for 0 to 9; lines 840-860).

If you look closely, you'll see that each recipe is twenty-seven characters long. Each recipe states which nine slices are stacked on top of one another. Therefore nine identifying numbers, each three digits long (000 to 127) make up a recipe. The recipe for the letter "A", the first recipe in line 560 actually specifies the slices #31, #17, #17, #49, #127, #97, #97, #97, #97, and #97. However, to keep everything the same all numbers are written with three digits, 31 becomes 031 etc. The commas and number signs are omitted. Using them would make it easier to read the recipes, but the computer would have to read each number specifying a slice separately.

The program loads the slices for a given letter into a printing array one by one. The "FOR-NEXT" loop in lines 1250-1300 picks up the slices by reading three numbers at a time across a recipe. It then stores the slices in the printing array TDPA$(N). The loop increments by three to read three digits of a recipe at a time. To store characters into the printing array a different index is incremented by one during each loop iteration to allow sequential storage. This loop is used for each letter to be printed. The other "FOR-NEXT" loop lines 1060-1310 increments to select the next letter to be printed. You'll note the limit for this loop is the length of the string inputted to the program.

Figure 2 shows a summary of the steps taken to form the letter "A".

Limitations

Only letters and numbers are printed. The program will give an error message if the ASCII value of a character does not fall in the letter-number range (lines 1170-1180, and 1640-1670) I didn't enter recipes for printing punctuation. Consequently, if a punctuation mark is entered the user will get an error message (lines 1640-1670). The program prints only upper case letters. If lower case letters are entered they are converted to upper case. The program can handle blanks though, and it will add a 9 x 7 blank matrix into the print array to print the blank (lines 1120-1150). After one name is printed the printer is sent two "line-feed" characters to advance the paper enough to make the output legible if another line is printed (lines 1450-1460).

In addition to this remember that the 13 character length limit is a function of the printed character size, and the width of the printer carriage. My printer is an Epson MX-100 which is 136 columns wide. An MX-80 or another 80 column printer will print something less that 13 characters (about (80/136)*13, or 8 characters) You can squeeze a couple more characters out by reducing the size of the space between printed characters to something less than 5 spaces (line 1320).

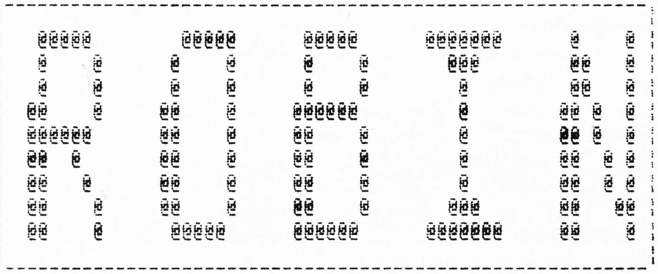
Variations and Improvements

HEADFIRST prints the letters shown in figure 1. The style is of my own creation, a sort of "computerese". If you don't particularly care for that style of type all you need do is change the numbers in the letter recipes. That means to change the style of type all you need do is change the nine sets of three digits in the 36 recipes used by this program to create "A" to "Z" and "0" to "9". While you're at it you may want to throw in some punctuation marks and other symbols. You'll need graph paper, a good knowledge of the binary number system, some imagination, and a little free time.

Possible improvements to this program are many. You could add a function to check the length of the name inputted and center it on the printed line if it is less than 13 characters in length. Another idea I have played with is changing the program to display rather than print. Then the output is on the crt. With the Apple /// this is as simple as changing the name of the device driver file in line number 1420. This is no major improvement until you make the program able to display a sentence or paragraph one word at a time and repeat it 100 times (Remember to change the length limitation to match your crt.) This would be a good one to leave on a machine at a computer store. They could make it say "BUY COMPUTERS AT JOES" etc. You or I could make it say "PLEASE UNPLUG ME BEFORE MY CORE MELTS AND RELEASES DEADLY RADIATION".

Whatever you decide to do with the program is up to you. Headfirst will allow you to show a visible use for your Apple /// and maybe let you have a little bit of fun with it. ///

FIGURE 1. Sample output from HEADFIRST.



```
1)  "A"=ASCII 65
2)   RECIPE$(65)="031017017049127097097097097
3)   SLICE$(031)=   @@@@@
     SLICE$(017)=   @   @
     SLICE$(017)=   @   @
     SLICE$(049)=  @@   @
     SLICE$(127)= @@@@@@@
     SLICE$(097)= @@    @
     SLICE$(097)= @@    @
     SLICE$(097)= @@    @
     SLICE$(097)= @@    @
```

FIGURE 2. Printing the letter "A". Step 1 — The ASCII value from the letter is obtained by the ASC function. Step 2 — the recipe for that character of that value is obtained from the array RECIPES$(I). Step 3 — three digit sections of the entry in the recipe array are used to call the necessary printing lines from the array SLICE$(I) to print the letter.

HEADFIRST: Program Listing

```
0    REM *******************************************************************
1    REM *                                                ------------------    *
2    REM *  HEADFIRST                                     | Copyright 1983 by |  *
3    REM *  ---------                                     |  O N   T H R E E  |  *
4    REM *  by Michael Gardner                            |   June-July 1983  |  *
5    REM *                                                ------------------    *
7    REM *                                                                      *
8    REM *  HeadFirst is a program that prints headings for '.Printer'          *
9    REM *  output etc...                                                       *
10   REM *                                                                      *
20   REM *  Writen for the Apple /// and Epson MX-100.                          *
30   REM *  March 1, 1982 -- Michael Gardner                                    *
40   REM *                                                                      *
50   REM * VARIABLE DICTIONARY                                                  *
60   REM * SLICE$(I)-ARRAY FOR DATA USED TO PRINT CHARACTER SLICES.             *
70   REM * RECIPE$(I)-ARRAY FOR RECIPES USED TO PRINT CHARACTERS.               *
80   REM * TDPA$(I)-ARRAY USED TO PRINT OUTPUT                                  *
90   REM * NAME$-USER INPUTTED NAME TO BE  PRINTED                              *
100   REM * NUM-ASCII VALUE OF A CHARACTER IN NAME$                             *
110   REM * TEMP$-A TEMPORARY STORAGE VARIABLE                                  *
120   REM * STRL-A NUMERIC VALUE EQUAL TO THE LENGTH OF NAME$                   *
130   REM$ PROGRAM BEGINS BY DIMENSIONING ARRAYS...                             *
140   REM *******************************************************************
150   REM
160   DIM SLICE$(127),RECIPE$(127),TDPA$(9)
170   REM
180   REM *******************************************************
190   REM * DATA USED TO PRINT LETTERS ONE SLICE AT A TIME.      *
200   REM *******************************************************
210   DATA"       ","      @","     @ ","     @@","    @  ","    @ @"
220   DATA"    @@ ","    @@@","   @   ","   @  @","   @ @ ","   @ @@"
230   DATA"   @@  ","   @@ @","   @@@ ","   @@@@","  @    ","  @   @"
240   DATA"  @  @ ","  @  @@","  @ @  ","  @ @ @","  @ @@ ","  @ @@@"
250   DATA"  @@   ","  @@  @","  @@ @ ","  @@ @@","  @@@  ","  @@@ @"
260   DATA"  @@@@ ","  @@@@@"," @     "," @    @"," @   @ "," @   @@"
270   DATA" @  @  "," @  @ @"," @  @@ "," @  @@@"," @ @   "," @ @  @"
280   DATA" @ @ @ "," @ @ @@"," @ @@  "," @ @@ @"," @ @@@ "," @ @@@@"
290   DATA" @@    "," @@   @"," @@  @ "," @@  @@"," @@ @  "," @@ @ @"
300   DATA" @@ @@ "," @@ @@@"," @@@   "," @@@  @"," @@@ @ "," @@@ @@"
310   DATA" @@@@  "," @@@@ @"," @@@@@ "," @@@@@@","@      ","@     @"
320   DATA"@    @ ","@    @@","@   @  ","@   @ @","@   @@ ","@   @@@"
330   DATA"@  @   ","@  @  @","@  @ @ ","@  @ @@","@  @@  ","@  @@ @"
340   DATA"@  @@@ ","@  @@@@","@ @    ","@ @   @","@ @  @ "
350   DATA"@ @  @@","@ @ @  ","@ @ @ @","@ @ @@ ","@ @ @@@","@ @@   "
360   DATA"@ @@  @","@ @@ @ ","@ @@ @@","@ @@@  ","@ @@@ @","@ @@@@ "
370   DATA"@ @@@@@","@@     ","@@    @","@@   @ ","@@   @@","@@  @  "
380   DATA"@@  @ @","@@  @@ ","@@  @@@","@@ @   ","@@ @  @","@@ @ @ "
390   DATA"@@ @ @@","@@ @@  ","@@ @@ @","@@ @@@ ","@@ @@@@"
400   DATA"@@@    ","@@@   @","@@@  @ ","@@@  @@","@@@ @  ","@@@ @ @"
410   DATA"@@@ @@ ","@@@ @@@","@@@@   ","@@@@  @","@@@@ @ ",@@@@@ @@"
420   DATA"@@@@@  ","@@@@@ @","@@@@@@ ","@@@@@@@"
430   REM
440   REM *******************************************************************
450   REM * READ IN DATA USED TO PRINT CHARACTER SLICES.                       *
460   REM *******************************************************************
470   REM
480   FOR I=0 TO 127
490     READ SLICE$(I)
500     NEXT I
510   REM
520   REM *******************************************************************
530   REM * RECIPES USED TO DESIGNATE WHICH SLICES ARE USED TO CONSTRUCT       *
540   REM * A GIVEN CHARACTER LISTED---A TO Z, AND 0 TO 9.                     *
550   REM *******************************************************************
560   REM
570   DATA"031017017049127097097097097","062033033126097097097097126"
580   DATA"031063096096096096096063031","060062035099099099099126124"
590   DATA"060032032096124096096096127","060032032096124096096096096"
600   DATA"031035032096103103101049031","033033033097127097097097097"
610   DATA"127028008008008008008028127","126002003003003003099098062"
620   DATA"034036040104120104104100098","032032032096096096097097127"
630   DATA"065099085073073065065065065","033049049105105101101099097"
640   DATA"031033033097097097097097062","062033033097126096096096096"
650   DATA"060034033097097097111034029","062033033097126100098097097"
660   DATA"031063065064125001065126124","127072008024024024024024024"
670   DATA"033033033097097097097063031","033033033033097097099118028"
```

Program Listing Continued on Page 26

WPL Revisited

by Bob Consorti

Welcome back to the column that teaches you just what the mysterious WPL is all about. Before we get into this month's column I think we had better correct two mistakes in the last one.

The seventh paragraph of the right hand column of page 20 is a short paragraph that the user is supposed to type in so that the WPL program can later use it when printing out letters. On one of the lines it says that each printing will set the left margin 5 spaces greater than the last printing. This is incorrect and should say that the left margin will be set 10 spaces greater than the last time.

On page 21, the right hand column, the fifth paragraph has a small mistake. The text '0 through 6' should read '0 through 60 (counting by 10's)'. Many readers caught this and understood what was meant by it, but I thought I'd better warn you before we go on.

As promised last time, we are going to learn just how the WPL program that was listed in the last issue works. Briefly recapping what we did, that program (listed below) was intended to be the 'Better, Faster, Smarter' version of another WPL program I presented. That program printed out a letter seven times to the console. After each printing, the left margin was increased by ten spaces. It showed off how easy it is to set up multiple printings from within an Apple Writer /// WPL program.

```
Start    PGO Begin
PrintIt  PPR
         PLM+10
         PNP
         PIN $A
         PRT
Begin    NY
         PND
         PPD.CONSOLE
         L.D2/LETTER
         PLMO
         PLM-10
         PAS Press RETURN to Continue =$A
         PSX 6
Loop     PSR PrintIt
         PSX -1
         PGO Loop
         PAS Done!! Press RETURN To Go Back To The Editor =$A
         PSR PrintIt
         PQT
```

Remember to type CONTROL OPEN APPLE Back-Slash after the last 'R' in the line 'PrintIt PPR'. This will clear the screen when the program gets here.

The above program does the same thing as our first one, only using a more 'Structured-Approach'. For those of you who know Basic or Pascal you should catch on quickly to what the program does and how it works.

WPL has something called variables that enables a WPL program to use numbers (0, 44, 9393...) and strings (any textual information — words, letters...) within the program to do repetitive tasks very quickly. If you turn to page 78 of your Apple Writer /// manual you will see just how to use numbers (numeric variables) in your program. Try out the example on that page and experiment a bit.

After some studying you will see that in our above program we use the variable "X". Now, what we do with it is something completely different. In our program we use the variable "X" to act as a counter in seeing just how many times we have printed out the letter. Our first version of the program has a number of very similar statements that did (mainly) the same thing.

If you changed your mind and wanted 50 letters printed out, each with a left margin only one space greater than the rest, the first method we presented would take up a lot of space and more importantly a lot of time. Using our updated program, very little needs to be changed if you want to print out 50 or even 500 letters.

Our new program uses what is called a 'SUBROUTINE'. This is just a group of statements that are used many times throughout a program. Notice how our first program had a number of similar lines. By using a subroutine in our new program we only have to type in the statements once. Whenever the program needed to carry out those similar statements it would call the subroutine with the command 'PSR label', where 'label' indicates where in the WPL program the subroutine is.

For our example program, the Subroutine is named 'PrintIt' and it is five lines long. The first line 'PrintIt PPR' shows that this is the start of the subroutine and the first statement is 'PPR', which prints out a control-backslash to the screen. This will clear the screen whenever we reach this statement. The next line 'PLM +10' sets the left margin 10 spaces greater than it currently is. The printing of the document occurs in the next line 'PNP' which simply prints out the current document in memory. We will skip what the line 'PIN $A' does for a minute and go to the last line 'PRT'. This statement says in effect "We're done so let's go back to the statement right after the one that called us". These commands are documented on page 90 of the Apple Writer /// manual. If you are still unsure about just what is happening, look over that page and by all means try the examples.

Getting back to the line we left out, for our Subroutine to do the same thing as the groups of statements in the original program, it must print out the line 'Press RETURN To Continue' after each printing except after the last one, when it will print out 'Done!! Press RETURN To Go Back To The Editor'. Since our subroutine has to print out two different things we used string variables.

These are described on pages 88 and 89 of the manual. In our program we assign the string $A with the two lines of text that we need printed. First it is set to 'Press RETURN To Continue'. When the subroutine 'PrintIt' is executed, we eventually get to the line 'PIN $A'. We have seen how the PIN function works, so the only strange thing about this line is the $A.

If you remember about PIN you should remember that it will print out the line of text immediately following PIN. Putting two and two together you should see that our program line will print out whatever is contained in the string $A and then wait for the user to press the 'RETURN' key. This is how we get the same subroutine to do a number of different things.

Now that we know what the individual parts of the program do, let's look at the program as a whole and see what is happening. The first line, 'Start PGO Begin' just does a jump to the main part of the program. Why not have the start at the start? Good question.!

We do this so that the program can run faster. You see, whenever a subroutine is called (in our case, the routine starting with the line 'Loop PSR PrintIt') Apple Writer /// scans through the entire program looking for the subroutine's label (PrintIt for us). Putting the subroutine at a place nearer to the start speeds up the program because Apple Writer /// doesn't have to look as far for the label.

Getting back to the Begin!, we see that the first line of the main part of the program is 'Begin NY'. The first part of this line is the label that shows the very first program line where to jump to. The second part of this line clears out memory. The next line 'PND' shuts off the screen display during printing and was discussed last time.

The line 'PPD.CONSOLE' sets the printing output device to be the console. This saves paper but can be changed if you want a hard copy of what is printed on the screen. The next line 'L.D2/LETTER' loads the sample letter that we will use during the printing. If you've been following along, it was printed in the last issue. It is reproduced below for those of you who can't find the disk or subdirectory you put it on.

This is a short line of text to illustrate one of the many uses of WPL in letter and general document preparation. When the WPL program is executed, each time you press RETURN the document will be printed with the left margin 10 spaces greater than the last time.

After you type the above line, save it with the name '.D2/LETTER'. If you want to put it somewhere else, just change the line that loads it into memory to point to wherever the file is and the program will work just fine.

The next two lines are related because together they set the left margin for the first and subsequent printings. The first of these, 'PLMO' sets the left margin to the leftmost column and the next line PLM-10' sets the margin to 10 spaces left of the first column.

Right about now you're probably asking why do that? The answer is because when we go to the Subroutine 'PrintIt', the left margin is incremented by 10 spaces so that each new printing is 10 spaces to the right of the last one. Since we want the first printing to start in the leftmost column on the screen, we initially set the left margin to be 10 spaces left of the leftmost column!

The next two lines set-up various other things for the programs use. The line 'PAS Press RETURN To Continue $A' sets-up the string $A to contain the line of text 'Press RETURN Continue'. This is the line that is shown after each printing except the last one.

The line 'PSX 6' sets the variable "X" to be the number of times you want to print the letter minus one. Thus if you wanted to print the letter 4 times, change this line to 'PSX 3'. The actual control of the printing is in the next five lines. The program will go to the Subroutine 'PrintIt' and when 'PrintIt' finishes printing the letter it will then go to the line 'PSX -1'. This line decrements the variable "X" (subtracts 1 from it).

All this looping around is nice but how do we get out of here? Well, in WPL there is a range of values that the numeric variable "X" can take on. It goes from 0 to 65535. There is also something called error flagging and detection that is discussed (partially) on page 82 of the Apple Writer /// manual. Whenever the value of "X" becomes zero, the very next statement is skipped and the program goes to the second line past the one that made "X" zero.

That's probably a bit tough to follow so let's look at our sample program and follow through what happens when it's time to finish up. As stated before, the WPL program will print our letter seven times, the last time through it will display the line 'Done!! Press RETURN To Go Back To The Editor'.

In our program what happens is this: The letter will be printed "X" times (6 for now) by the main loop. The sixth (and last) time through, the value of "X" will be zero. This tells the computer that an error has occurred. From what has been said before, Apple Writer /// will skip the very next line 'PGO Loop' and go the line following that one.

Here the closing message is assigned to the string $A and the letter is printed one last time. When 'PrintIt' finishes printing, this message is displayed and we return and go to the line 'NY' which clears out any text in memory. Finally, the line 'PQT' tells Apple Writer /// that the WPL program is finished and we are done!

Help Screen HELP!

At one time or another all of us have forgotten exactly how to use one or more of the many features of Apple Writer ///. Since many of us aren't yet blessed with the index to the Apple Writer /// manual, we sometimes use the help screen feature of the program to help us remember what to do.

Quite a few of you have hard disks and are no doubt impatient over the time it takes to load the help files from the built in floppy disk. If you haven't already bought one of the 'Interpreter-Switcher' programs, like the very fine Catalyst utility from Quark, that allows you to put all of your programs (protected ones too!) on your hard disk, you will be interested in the next section. We will put all of the help screens on the hard disk to get a much quicker response from the Help Screen Menu.

The following WPL program doesn't limit you to working with a ProFile hard disk, you can put help screens on any disk device — your Davong, Xcomp or even Micro-Sci high capacity floppy will work just fine! To start you must first use the System Utilities diskette to create a couple of subdirectories on your hard disk. From this point on I will use the term 'hard disk' to refer to any external disk drive, so if you aren't working with a ProFile don't worry.

After booting up your System Utilities diskette, type 'F' to go to the File Handling Commands Menu. Next type 'M' to create a new subdirectory. Create a subdirectory with a name in the format '.drive/AW3'. If you are using a ProFile enter '.PROFILE/AW3'. Next, create a subdirectory with the name in the format '.drive/AW3/HS'. Again, for the ProFile enter '.PROFILE/AW3/HS' at this point.

Press the ESCAPE key to go back to the File Handling Commands Menu and type 'C' to copy files. Put your Apple Writer /// master diskette in the built in drive and enter '.D1/HS/ ' when it prompts you for what files to copy. Then enter '.PROFILE/AW3/HS/ ' when it prompts you for where to copy the files to. After you press the RETURN key it will take a minute or so to do the actual copying.

Once all the copying is done, press the ESCAPE key to go back to the File Handling Commands Menu and type 'W' to change the write-protection. We are going to change the 'HELP' file on the Apple Writer /// boot disk and we must therefore unlock it first, so when the prompt appears enter '.D1/HELP' and then type 'N' to unlock it.

Once you are finished with that you can boot the Apple Writer /// and type in the following WPL program that will make the necessary changes. Don't forget the CONTROL OPEN APPLE Back-Slash right after the PPR of the third line. This will clear the screen before the program really starts.

Help Screen Move Utility
by Bob Consorti

```
Start
     PND
     PPR
     PPR     Help Screen Move Utility
     PPR     by Bob Consorti
     PPR
     PPR
     PPR     Enter the pathname of where you put the help screens.
     PPR     If you have followed instructions, this should be
     PPR     the directory. .PROFILE/AW3/HS. Press RETURN to accept
     PPR     the default, or enter the full pathname          =$A
     PCS     /$A//
     PGO     Default
     PGO     Next
Default
     PAS .PROFILE/AW3/HS  $A
Next PPR     Make sure your Apple Writer /// Boot disk is in the
     PIN     built in drive and then press the RETURN key.
     NY
     L.D1/HELP
     B
     F!.D1/HS!$A!a
     S
     NY
     PPR
     PPR     The Help file on the Apple Writer /// boot disk has
     PPR     been updated. Now updating the Help Screen files.
     PPR
     PPR     This will take a little while...
     L$A/HSCOMMANDS
     F$d1/hs!$A!a
     S
     NY
     L$A/HSPRINT
     F!.d1/hs!$A!a
     S
     PPR
     PPR     End of Help Screen Move Utility. Files have been updated.
     PIN               (Press RETURN) to go back to the Editor.
     NY
     PQT
```

When you have finished typing it in, save it to a disk with the name 'HSM.UTIL' for Help Screen Move Utility. Once it is on disk you can execute the WPL program by typing '(P)DO HSM.UTIL'. Follow the instructions on the screen and the program will automatically update the files in a minute or so.

Once it is finished type OPEN APPLE ? and then another ? to go to the help screen menu. From here type in a number from 1 to 10 and press the RETURN key. While before, the help screen files were loaded (rather slowly) from the internal disk, they will now come off of the hard disk.

This has distinct advantages since now only the file 'HELP' is needed on the boot disk (or rather — the disk in the internal drive) for help screen operations. If you're like me you probably don't leave your boot disks in the internal drive. With Apple Writer /// this has always been a problem because even though I don't mind putting a normal disk in the internal disk drive and leaving it there, I don't want to waste the disk space necessary for all the help screens and subdirectories. With this routine I only use up three or four blocks on a normal diskette (for the HELP file) and after booting up I can put it in the internal drive and forget about it.

The program is pretty straight forward and you should be able to get through it easily with what you've learned so far. Next time we will study this program in more detail, but to learn WPL you must do the examples I give and make up your own if you want to become proficient.

Until next time, a question: Using Apple Writer /// Version 1.0 can you use a partial prefix? That is to say, if you've set the prefix to '.PROFILE/AW3' and you want to load the file '.PROFILE/AW3/LETTERS/MAMIE', what's the best way to do it? How about if you've set the prefix to '.PROFILE/AW3' and you want to catalog the files on the directory '.PROFILE/AW3/LETTERS' because you can't remember when you last wrote Mamie? ///

Continued from page 43

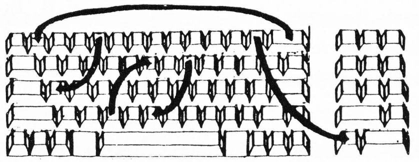
```
        Bottom Prompt;
        EXIT (Do_Option)
      END;
    IF (Disk [1] <> '.') THEN
      BEGIN
        GOTOXY (0, 7);
        WRITE (CHR (Bell), 'The Device name must be ".D1 - .D4"');
        Bottom Prompt;
        EXIT (Do_Option)
      END;
    IF (POS (Disk, Valid_Disk) = 0) THEN
      BEGIN
        GOTOXY (0, 7);
        WRITE (CHR (Bell), 'Cannot format ', Disk);
        Bottom Prompt;
        EXIT (Do_Option)
      END;
  ($IOCHECK-}
    RESET (File_ID, Disk);
  ($IOCHECK+}
    IF (IORESULT = 0) THEN
      Disk Formatted;
    CLOSE (File ID)
  END; { Of PROCEDURE Test_Disk_Name }


  PROCEDURE Bad_Name;
  BEGIN
    WRITELN;
    WRITELN (CHR (Bell), 'Bad volume name!');
    WRITELN;
    WRITE ('It must begin with a letter (A-Z), then the rest can be');
    WRITE (' any combination of      letters and periods (".").  It must');
    WRITELN (' also be less than 16 characters in length.');
    Bottom Prompt;
    EXIT (Do Option)
  END; { Of PROCEDURE Bad_Name }


  PROCEDURE Test Name;          { Check to see if it's a valid volume name }
  VAR counter: INTEGER;

  BEGIN
    IF ((LENGTH (New_Name) = 0) OR (LENGTH (New_Name) > 15)) THEN
      Bad name;
    IF (NOT (New_Name [1] IN A_to_Z)) THEN
      Bad name;
    FOR counter := 1 TO LENGTH (New Name) DO
      IF (NOT (New_Name [counter] IN A_to_Z_and_Special)) THEN
        Bad Name
  END; { Of PROCEDURE Test_Name }
```

Program Listing Continued on page 47

/// to the Max

by Al Evans

The Key to the Keyboard



As promised in our last column, we will now change the Apple ///'s keyboard map to match our personal tastes. You VisiCalc users can put the "*" and "+" on keys that are easier to reach. You Pascal programmers can remove CONTROL-BACKSLASH from the map for an easy way to keep users from BREAKing out of your programs. And just for a laugh, you crazies can attain new levels of system security by changing the map so that only you know which keys produce which letters.

To complete this lesson, you'll need your trusty Standard Device Drivers Guide and the Pascal system. If you don't have the Pascal system, go to the nearest Apple store and buy it — you CANNOT have intimate relations with an Apple /// without it. For those who have a disk zapper-type program, such as my own BLOKACCESS, we'll give a quick and dirty method of making a few changes directly to a keyboard layout file on disk.

An Apple /// keyboard layout file is a two-block file produced by the Apple /// Pascal assembler. The first block is a header, and is irrelevant for our purposes. However, the System Configuration Program presumably expects it, and you should probably leave it alone. The second block has two parts: a 256-byte lookup table, and 256 bytes of mostly zeros with a little linker data thrown in. We are interested only in the first half of the second block; bytes 0 to 255 (See Figure 1).

There are two levels of understanding: knowing how something works and knowing why it works. In this case, I have not made the second level. I know that the first 188 bytes of this second block are the keyboard layout. But I don't know: (1) What the last 68 bytes are for. My contacts at Apple tell me they've got to be there, they never change, and their purpose is a secret. (2) Why all characters except the lower case letters are represented in positive ASCII (MSB clear), but the lower case letters are represented in negative ASCII (MSB set). The first person to send me an imaginative and plausible answer to either of these questions will receive an autographed copy of InVisiCalc, the financial management program that doesn't exist.

For now, let's ignore all that, and look at the first 188 bytes. This table contains one 4-byte entry for each of the 47 "standard" keys (see pages 135-137 in the SDD Guide). Each group of 4 bytes represents the ASCII characters which can be produced by the key alone, the key with the SHIFT key held down, the key with the CONTROL held down, and the key with both SHIFT and CONTROL held down. As mentioned above, the lower-case letters in this table have their most significant set, so the numbers in the table for these letters are the ASCII codes for the letters plus 128 ($80).

And that's all you need to know to change a keyboard layout with a disk-zapper. Just make a copy of the file you want to change, read the second block of that copy, and change the key assignments as desired. Remember to add 128 ($80) to the ASCII codes for any lower-case letters you put in the table. When you've finished making changes, write the block back to the same spot on the disk. Now you can use the System Configuration Program to install your new keyboard layout in the SOS.DRIVER file of your favorite boot disk, and the keys you desire will forever be at your fingertips.

If you want to make more extensive changes, or if you become addicted to custom keyboard layouts, you'll probably want to use the program shown in listings 1 and 2. Listing 1 is a Pascal program which allows you to read a keyboard layout file, modify it as desired, and write the modified file to a block storage device in such a form that the System Configuration Program will accept it. Listing 2 is a short external assembly language routine which lets you install your modified layout in the system immediately, if you wish.

Enter the Pascal program in listing 1, save it as KBD.LAYOUT, and C(ompile it to a codefile of the same name. Enter the assembly-language program in listing 2, save it as INSTL.AL, and A(ssemble it to a codefile of the same name. Invoke the linker, enter KBD.LAYOUT as the host file, and enter INSTL.AL as the library file. Press <RETURN> when asked for the next library file. Press <RETURN> when asked for a map file (or enter .CONSOLE, if you like arcane feedback). Finally, enter KBD.LNK when asked for a destination file. Back at the command line, eX(ecute KBD.LNK.

The program won't let you do anything until you R(ead a keyboard layout file. Put your system utilities data disk online, press "R", and enter /UTILITIES.DATA/KEYBOARD.LAYOUT/SHOLES for a file pathname. You should now be back at the program's main command line. Press "E" to E(dit the keyboard layout.

You may have observed by now that the key numbers shown on page 135 of the SDD Guide do not agree with those shown on pages 136-137. This program uses the numbers from the diagram on page 135. For example, let's replace the accent aigu on key 25 with a "+" sign. Enter "25" as the key number to edit. The display will show the ASCII codes presently assigned to key 25, and ask for new ASCII codes for this key in all its possible conditions (see p. 136 of the SDD Guide, entry for key 26). Enter "43" (the ASCII code for the "+" sign) as "New ASCII code for <KEY> ALONE". Re-enter the same numbers presently shown in the display for the other conditions, since we don't want to change anything else. Then press "Y" to accept the changes. The program will ask for a new key number to edit; enter "—1" to return to the main command line, "I" to install the new layout table in your system, then press <ESCAPE> to exit the program.

Now go to the Pascal editor and test your new "+" key. While you're there, make sure the others keys are still working properly, too. If they're not, or if you can't get to the editor, you've made a mistake in entering the assembly-language routine; you'll have to re-boot, fix the mistake, re-assemble the routine, and link it to the codefile again. This is why we used KBD.LNK as the name for the final product — if we'd used KBD.LAYOUT again, we'd have to recompile the Pascal program before we could relink. After everything is working right, use the filer to change KBD.LINK.CODE to KBD.LAYOUT.CODE, since you no longer need the unlinked version.

That about covers making a keyboard layout editor and using it. Now we'll talk about how it works. And pontificate a bit about Pascal.

Other than initializing global variables, the main program just acts as a command processor and ensures that a keyboard layout file is read before it is edited, written, or installed. Each command is processed by a separate procedure.

The one external assembly-language procedure moves the applicable portion of the KEYBOARD record (i.e. KEYBOARD.KEY_TABLE) to system memory. Thanks to Dr. John Jeppson, one of our foremost Apple /// explorers, we know that the keyboard layout table resides at $1700-$17FF in system blank memory. Although this action is Not According to SOS, all we have to do to change the keyboard layout "on the fly" is to download the right 256 bytes to this area.

The Pascal procedures were written for simplicity and brevity, and include no error-trapping — feel free to expand them if you like. In particular, the procedure Edit_Layout has its own main commdnd-processing loop and several sub-procedures. You could, for example, add a command to display a complete keyboard layout. Or modify the sub-procedure Show_Line to display the actual characters as well as the numeric ASCII codes, when applicable. The best way to understand how a program functions is to change it to work like YOU want it to.

Which brings me to the subject of Pascal style. It seems to me that entirely too many data structures in entirely too many Pascal programs are defined simply as "Packed Array(0..n) of 0..255", i.e. as strings of bytes. Sometimes this is necessary. For example, a SOS directory entry will not fit directly into a Pascal record structure simply because it contains an odd number of bytes and Pascal data comes in 2-byte words.

But whenever possible, define your data so that you can refer to its components by NAME! This is one of the major features of Pascal, and is responsible for much of the power of the language.

For example, I could have defined KEYBOARD as a Packed Array(0..1023) of 0..255. However, this would force me to address individual elements of that table by offsets instead of by name. For example, a particular entry in the layout table can be addressed as:

KEYBOARD.KEY_TABLE.KEY(NUM)(COND)

If KEYBOARD were defined as a packed array, we would have to say

KEYBOARD(512+KEY_NUM*4+COND)

The procedure call

Install (KEYBOARD.KEY_TABLE)

would become

Install (KEYBOARD(512))

The assignment statement

THIS_KEY: KEYBOARD.KEY_TABLE.KEY(NUM)

would be

Moveleft (KEYBOARD(512+4*NUM,THIS_KEY,4)

Either way works fine, but I think you'll agree that the Pascal record format makes it much more obvious precisely what is happening. It's also better for purposes of remembering what the program does a year or so from now. Use records whenever you can!

That's about it for this column. Next time we'll move away from serious applications and talk about making funny noises. Incidentally, the game I mentioned in the last column, (Cap'n Magneto, for you late-comers) is now up to 3700 lines of Pascal and about 5K of assembly language. It's totally different from any game you've ever seen, runs only on the Apple ///, and should be available around October. Meanwhile, write me and tell me what you're doing with your ///. Even better, tell me what you WANT to do — how YOU want to push YOUR /// to the Max. ///

Figure 1. Layout table for the Sholes keyboard.

```
$00: 31 21 31 21 32 40 32 00 33 23 33 23 34 24 34 24    1!1!2@2.3#3#4$4$
$10: 35 25 35 25 36 5E 36 1E 37 26 37 26 38 2A 38 2A    5%5%6^6.7&7&8*8*
$20: 39 28 39 28 30 29 30 29 2D 5F 2D 1F 3D 2B 3D 2B    9(9(0)0)-_-.=+=+
$30: 5C 7C 1C 7F F1 51 11 11 F7 57 17 17 E5 45 05 05    .|q@..wW..eE..
$40: F2 52 12 12 F4 54 14 14 F9 59 19 19 F5 55 15 15    rR..tT..yY..uU..
$50: E9 49 09 09 EF 4F 0F 0F F0 50 10 10 5B 7B 1B 1B    iI..oO..pP..[{..
$60: 5D 7D 1D 1D 60 7E 60 7E E1 41 01 01 F3 53 13 13    ]}..`~`~aA..sS..
$70: E4 44 04 04 E6 46 06 06 E7 47 07 07 E8 48 08 08    dD..fF..gG..hH..
$80: EA 4A 0A 0A EB 4B 0B 0B EC 4C 0C 0C 3B 3A 3B 3A    jJ..kK..lL..;:;:
$90: 27 22 27 22 FA 5A 1A 1A F8 58 18 18 E3 43 03 03    '"'"zZ..xX..cC..
$A0: F6 56 16 16 E2 42 02 02 EE 4E 0E 0E ED 4D 0D 0D    vV..bB..nN..mM..
$B0: 2C 3C 2C 3C 2E 3E 2E 3E 2F 3F 2F 3F 00 00 64 30    ,<,<.>.>/?/?..d0
$C0: 04 68 A4 9C 70 3C 74 78 7C 50 80 84 88 AC A8 54    .h..p<tx|P.....T
$D0: 58 34 40 6C 44 4C A0 38 98 48 94 5C 30 60 14 28    X4@lDL.8.H..0`.(
$E0: 64 00 90 08 0C 10 18 90 20 24 1C 2C B0 28 B4 B8    d....... $.,.(..
$F0: 24 00 04 08 0C 10 14 18 1C 20 8C 8C B0 2C B4 B8    $........ ...,..
```

KeyBoardLayout: Program Listing #1

```
PROGRAM EDIT_KBD_LAYOUT;

{ ****************************************************************** }
{ *                                             --------------------- * }
{ *  Keyboard Layout Editor                    | Copyright 1983 by | * }
{ *  ----------------------                    |  O N   T H R E E  | * }
{ *  by Al Evans                               |  June-July, 1983  | * }
{ *                                             --------------------- * }
{ *  This program lets you edit and change the keyboard layout.  You   * }
{ *  can now change the keyboard positions to suit your liking.  Change * }
{ *  the '+' and '*' keys to an easier to reach position for Visicalc  * }
{ *  users.  Move around anything you like, or even remove certain keys * }
{ *  from your keyboard!                                                * }
{ *                                                                     * }
{ ****************************************************************** }


TYPE Byte = 0..255;
     Key_Number = 0..46;
     Key_Cond = (ALONE, SHIFT, CONTROL, BOTH);
     Key_Desc = Packed Array[Key_Cond] of Byte;
     Layout_Tbl = Packed Record
                    KEY: Packed Array[Key_Number] of Key_Desc;
                    MYSTERY: Packed Array[0..67] of Byte
                  END;
     Kbd_Layout = Packed Record
                   HEADER: Packed Array[0..511] of Byte;
                   KEY_TABLE: Layout_Tbl;
                   TRAILER: Packed Array[0..255] of Byte
                 END;

VAR KEYBOARD: Kbd_Layout;
    KBD_FILE: File of Kbd_Layout;
    COMMAND: Char;
    LAYOUT_READ: Boolean;

Procedure Read_Kbd_File;
VAR PATHNAME: String[255];
BEGIN
  Write(Chr(28)); {Clear screen}
  Writeln('READ KEYBOARD LAYOUT FILE');
  Write('Pathname of keyboard layout file: ');
  Readln(PATHNAME);
  Reset(KBD_FILE, PATHNAME);
  KEYBOARD:= KBD_FILE^;
  Close(KBD_FILE);
  LAYOUT_READ:= TRUE;
END;

Procedure Write_Kbd_File;
VAR PATHNAME: String[255];
BEGIN
  Write(Chr(28)); {Clear screen}
  Writeln('WRITE KEYBOARD LAYOUT FILE');
  Write('Pathname for new keyboard layout file: ');
  Readln(PATHNAME);
  Rewrite(KBD_FILE, PATHNAME);
  KBD_FILE^:= KEYBOARD;
  Put(KBD_FILE);
  Close(KBD_FILE, LOCK)
END;
```

Program Listing Continued on Page 19

Assembling (ON) the ///

by Martin Nichols

Once again, I'm back to confuse you with little tidbits of information. We will get to what I promised last time in a minute or so but I'd like to get on my soapbox for now and say some things I've wanted to say for years.

How long have YOU been waiting for the SOS Reference Manuals? I was looking for a copy over two years ago! Just a short while back I finaly got a set. The /// has been around for a few years now and Apple is just starting to provide the information needed to program on it. There should be no wonder why there is very little in the way of software for the beast.

The most interesting and useful Apple)(programs came not from giant multi-national corporations, but from dedicated hackers. Apple made available to everyone (and not just licensed developers!) all the needed information for the)(but chose a different route with the ///.

With the release of the //e, Apple concurrently released all the new technical manuals for that machine. Why did it take over three years to do the same thing for the ///? You tell me. Well, that's the end (almost) of my griping! The SOS manuals are, for the most part, great.

One thing I can't figure out is why Apple left out some things. Two (that I know of) SOS calls were left out of those new manuals. They are the device calls D_READ and D_WRITE. They allow the programmer to read or write a block of data from any logical block on a disk device. Pascal can't live without them and many programs already use them extensively.

The only word I can get out of Apple is that they are afraid that a poor programmer could seriously damage his program and/or data disks if given this information. What kind of BS is that? If they want to protect the world from bad programs, why didn't they just not release any language system for the ///. Apple could keep everyone's data safe from us bad programmers if there was no Business Basic or Pascal development tools.

If Apple)(programmers had that same 'support' from Apple I would bet you anything that the)(would have done just as poorly as the /// is now doing. Since I think it's clear that there is a problem, the only thing we need to talk about is a solution. What can be done? Open up the machine and release everything that licensed developers have access to and more! This is in Apple's best interests and I hope they see that.

Formatting Disks

That's enough of that! Let's get down to the business at hand, which is the business of formatting disks. Everyone should be familiar with this, without formatted disks your machine can't store any data.

Until now, when you ran out of room on your disks, you usually had to leave the program you are working on, run out to your local computer store, buy a new box of disks, run home and boot up the System Utilities Disk. Now you used that program to format the new disks so you could store information on them.

At this point in time only one or two programs out there let you format a disk from within a program, rather than having you boot up the Utilities disk to do it. I think the main reason is lack of knowledge of HOW to format a disk. Since I am going to show everyone just how (and provide examples to do this from both Basic and Pascal) hopefully in the near future all programs will take advantage of this feature.

The key to our method of formatting disks is the assembly language routine 'Formatting Utility' (Program Listing #1). It contains an invokable (or linkable) procedure that allow Basic and Pascal programs to very easily format a disk. See the listings of the demonstration programs to check exactly how to use this procedure from within your Basic or Pascal programs.

A key note that you must remember is that this routine needs the device drivers '.FMTDx' . Without them it will not work and an appropriate error code will be returned. The entire invokable module takes up only 2.5 K of memory in Business Basic and a similarly negligible amount in Pascal.

Program Listing #1 is a dual purpose program that will assemble to a Pascal or Basic version of the formatting routine. As it is listed, you will get a Basic version when it is assembled. To create a Pascal version, change the value of the label 'Pseudo' to be 'OEO'. This is the only thing that you need to change. The June-July Disk of the Month will contain both versions of this routine.

Once you have entered Program Listing #1, save it with the name 'FORMAT.TEXT'. Assemble it and then enter the Filer to change the name of the code file from 'FORMAT.CODE' to 'FORMAT.INV'. Next, boot up a Basic disk and enter the Basic Documentation and Test Program (Listing #2). If you aren't using a Catalyst system you will probably have to add the device driver '.FMTDx' to your Basic disk, since it isn't normally there. Once you have it all typed in RUN it and you will be able to format diskettes from Basic!

To do the Pascal version, remember to change the value of the label 'Pseudo' to 'OEO'. Next save this test and assemble the routine with the name 'P.FORMAT'. Before you can do the Pascal test program you have to type in and assemble Program Listing #3. This is an assembly language routine to rename a disk and is needed since Pascal has no 'RENAME' command as Business Basic does.

Type it in and save it with the name 'PASCAL.RNM'. Assemble this routine and we can now write the Pascal test program. Enter the test of Program Listing #4. This is the Pascal test program and you should save it with the name 'PASCAL.FOR' Compile it and we are almost done.

We now have to link together the assembly language routines and the Pascal test program. Invoke the linker by typing 'L' at the Pascal Main Command Line. When it prompts you, enter the name 'PASCAL.FOR', or whatever name you saved the Pascal test program under. Next enter 'PASCAL.RNM' and press RETURN. Now type in 'P.FORMAT' and press RETURN. Hit RETURN twice again and then enter 'PASCAL.FOR' as the output file.

If you followed the instructions you should now have a working copy of the Pascal test program. To test it out, execute the file 'PASCAL.FOR' and try out the formatting routine.

Closing Notes

These programs will do the same general thing that the System Utilities Format option does, and some thing that it doesn't. If you use Basic or Apple Writer /// and 'Catalog' a disk you probably have noticed that after the name of the directory you're listing there is a date inside parenthesis and then a 'V' followed by a number.

The System Utilities Disk fills these fields with zeroes when it formats a disk. They are supposed to hold the date and time that the disk was formatted and something called a volume number. The volume number is insignificant and can be left zero. The date created is a bit more important.

Since diskettes are fragile things, I like to periodically make backups. After I have used a diskette a certain amount of time I retire it from service and go on by using the Backup copy and so on. Since the System Utilities Format Option does not put the current time on the disk when it formats it (yes, even systems with working clock kits!) I have to write the day it was formatted on the disk lael.

This routine will stamp the time and date information of the disk when it formats it. By doing this it makes our lives a bit easier because we don't have to remember as much — just catalog the disk and see when it was created!

That's it for now. To help me to better help you, write me care of ON THREE and tell me what you want to see. I didn't go into any detail of how the assembly language routines and test programs in this issue work because I don't know if people want that. Please say so! That way we can help each other. Next time? Who knows? Well, let's just say it will be interesting. ///

Assembling (ON) the ///: Program Listing #1

```
; *********************************************************************
; *                                                -------------------  *
; *  Formatting Utility                            | Copyright 1983 by |  *
; *  ------------------                            | O N   T H R E E  |  *
; *  by Martin Nichols                             |  June-July, 1983  |  *
; *                                                -------------------  *
; *                                                                     *
; *  This assembly language routine will enable your Basic or Pascal   *
; *  programs to format a diskette.  You will never again have to force *
; *  the user to boot the System Utilities Disk to format a disk.      *
; *                                                                     *
; *  This routine will work equally well with Apple 140K disk drives   *
; *  and any of the Micro-Sci drives.                                  *
; *                                                                     *
; *  Just invoke (or Link) this routine and your programs can easily   *
; *  format diskettes.  A note of caution: The drivers 'FMTDx' must be *
; *  in your boot disks 'SOS.DRIVER' file.  Without them, the routine  *
; *  will not work and will return an appropriate error message.       *
; *                                                                     *
; *  See the Pascal and Basic test programs for examples of how to     *
; *  use this routine.                                                 *
; *                                                                     *
; *  To create a routine that can be used from Basic, set the label    *
; *  'Pseudo' to 0E8.  If you want to use it from Pascal set the label *
; *  'Pseudo' to 0E0.  The listing below is set up to assemble a Basic *
; *  version of the routine.                                           *
; *                                                                     *
; *********************************************************************


          .MACRO  POP                 ; Pull a word from the stack
          PLA
          STA     %1
          PLA
          STA     %1+1
          .ENDM

          .MACRO  PUSH                ; Push it back on
          LDA     %1+1
          PHA
          LDA     %1
          PHA
          .ENDM

          .MACRO  SOS                 ; Macro def for SOS call block
          BRK                         ; Begin SOS call block
          .BYTE   %1                  ; call_num
          .WORD   %2                  ; parameter_list pointer
          .ENDM                       ; end of macro definition
```

```
Get_Time        .EQU    63                    ; call_num for GET_TIME
D_Write         .EQU    81                    ; call_num for D_WRITE
D_Control       .EQU    83                    ; call_num for D_CONTROL
Get_Dev_Num     .EQU    84                    ; call_num for GET_DEV_NUM
D_Info          .EQU    85                    ; call_num for D_INFO

Pseudo          .EQU    0E8                   ; Set for Basic.  Change to 0E0 for
                                              ; a Pascal assembly language routine.

;
; Use the following notation to call this routine:
;
; From Basic use:  PERFORM FormatDisk (%disk.num, %vol.num, @er%)
;
; From Pascal use: Format_Disk (disk_num, vol_num, error);
;


                .PROC   Format_Disk, 3  ; Three Parameters for this routine

Begin           JMP     Start           ; Go around all the buffers

Block_Buf       .BLOCK  200, 00

BootBuf         .EQU    *


Directory       .BYTE   00, 00, 03, 00          ; The directory info
NameLen         .BYTE   0FA
Name            .ASCII  "BLANK.DISK"
                .BYTE   00, 00, 00, 00, 00
                .BYTE   75, 00, 00, 00, 00, 00, 00, 00
TimeType        .BYTE   00, 00, 00, 00
Volume          .BYTE   00
                .BYTE   00, 0C3, 27, 0D, 00, 00, 06, 00
Blocks          .WORD   00

Bytes_To_Write  .BYTE   00

Formatter       .BYTE   06
                .ASCII  ".FMTD"               ; disk formatter
Suffix          .BYTE   00                    ; suffix of the formatter

GD_List         .BYTE   02
                .WORD   Formatter
Dev_ref         .BYTE   00                    ; device number goes here

D_Ptr           .BYTE   03
                .ASCII  ".D"
D_Suffix        .BYTE   00                    ; suffix of the disk drive (1-4)

D_List          .BYTE   02
                .WORD   D_Ptr
Dev_num         .BYTE   00

Info_Options    .BYTE   00, 00, 00, 00, 00
Total_Blocks    .WORD   00                    ; Here is the number of blocks
                                              ; that the disk drive can hold.
Info_List       .BYTE   04
Info_num        .BYTE   00                    ; Put the device number here
                .WORD   D_Ptr
                .WORD   Info_Options
                .BYTE   07

Ctrl_List       .BYTE   03
Ctrl_ref        .BYTE   00                    ; Here will be the Ref_Num
                .BYTE   0FE                   ; code to format the disk
                .WORD   Block_Buf             ; Pointer to page that will be
                                              ; repeated on all the blocks

Write_List      .BYTE   04                    ; Here is the list to write one
D_num           .BYTE   00                    ; block of data to the disk
Buf_Ptr         .WORD   0000
                .WORD   0200                  ; 1 block (512 bytes)
Block_num       .WORD   0000                  ; The block number will go here

Boot_Ptr        .WORD   BootBuf
Block_Ptr       .WORD   Block_Buf

Error_Code      .BYTE   00                    ; Use this location to store
                                              ; any SOS errors.
Return          .WORD   00                    ; Put the return address here

Start
                POP     Return                ; Save return address
                PLA                           ; Get the address of the
                STA     Pseudo                ; error variable.  Use this
                PLA                           ; to store any error that is
                STA     Pseudo+1              ; generated by the routine.
                PLA                           ; Get the volume number and
                STA     Volume                ; store it in the right place.
                PLA                           ; Discard the MSB
                PLA                           ; Get the disk number to format
                ADC     #30                   ; Convert to ascii
                STA     Suffix                ; Put it in the proper places
                STA     D_Suffix
                PLA                           ; Remove the MSB and discard it
                LDA     #00                   ; Zero the error code
                STA     Error_Code
                JSR     Do_Dev_Nums           ; Get the device numbers
```

```
             LDA    #00
             CMP    Error_Code
             BEQ    $1
             JMP    S_Finish          ; Leave if an error occurs
$1           JSR    Do_D_Info         ; Get the device info (# of blocks)
             JSR    Format            ; Try and format the disk
             LDX    #0FF
             LDA    #00
Clear        STA    Block_Buf,X       ; After formatting, clear off
             DEX                      ; the page that was used as the
             BNE    Clear             ; repeating buffer for the format.
             STA    Block_Buf
             LDA    #00
             CMP    Error_Code
             BEQ    $2
             JMP    S_Finish
$2           JSR    Write_Blocks      ; Write out the neccessary blocks

S_Finish
             LDA    #00
             TAY
             STA    @Pseudo,Y         ; Save the error code
             LDA    Error_Code        ; (If any!)
             INY
             STA    @Pseudo,Y
             PUSH   Return            ; Restore return address
             RTS                      ; and come back

Do_D_Info                             ; Find the number of blocks
             SOS    D_Info, Info_List ; blocks that this device
             LDA    Total_Blocks      ; has and compute the right
             CMP    #18               ; bit_map type.
             BNE    $1
             LDA    #22               ; This is for a standard
             STA    Bytes_To_Write    ; 280 block disk.
             LDA    #18
             STA    Blocks
             LDA    #01
             STA    Blocks+1
             JMP    $4
$1           LDA    Total_Blocks
             CMP    #30
             BNE    $2
             LDA    #45               ; This is for a Micro-Sci
             STA    Bytes_To_Write    ; 560 block disk.
             LDA    #30
             STA    Blocks
             LDA    #02
             STA    Blocks+1
             JMP    $4
$2           LDA    Total_Blocks
             CMP    #60
             BNE    $3
             LDA    #8B               ; This is for a Micro-Sci
             STA    Bytes_To_Write    ; 1120 block disk.
             LDA    #60
             STA    Blocks
             LDA    #04
             STA    Blocks+1
             JMP    $4
$3           LDA    #00               ; If we ever get here, set-up
             STA    Bytes_To_Write    ; for a disk of zero blocks
             STA    Blocks            ; and adjust everything
             STA    Blocks+1          ; accordingly.
$4           RTS

Do_Dev_Nums                           ; Find the device numbers
             SOS    Get_Dev_Num, GD_List
             BNE    Err1
             LDA    Dev_ref
             STA    Ctrl_ref
             SOS    Get_Dev_Num, D_List
             BNE    Err1
             LDA    Dev_num
             STA    D_num
             STA    Info_num
             RTS

Format                                       ; Format the disk
             LDX    #0FF                     ; Set up the page that
             LDA    #0EE                     ; will be repeated on
Again        STA    Block_Buf,X              ; all blocks of the disk.
             DEX

             BNE    Again
             STA    Block_Buf
             SOS    D_Control, Ctrl_List
             BNE    Err1
             RTS

Err1                                         ; If there is an error,
             STA    Error_Code               ; store the error code
             RTS                             ; and return.

Write_Blocks
             LDA    Boot_Ptr
             STA    Buf_Ptr
             LDA    Boot_Ptr+1
             STA    Buf_Ptr+1
             LDA    #00
             STA    Block_num
```

```
             STA    Block_num+1
             SOS    D_Write, Write_List    ; Write Block 0
             BNE    Err1

             LDA    Block_Ptr
             STA    Buf_Ptr
             LDA    Block_Ptr+1
             STA    Buf_Ptr+1              ; Setup new buffer

             LDX    #2A
             LDA    #00
Next         STA    Block_Buf,X
             DEX
             BNE    Next                   ; Zero the directory info

             LDA    #02
             STA    Block_num
             LDA    #00
             STA    Block_num+1            ; Setup to write block 2

             LDA    #00
             STA    Block_Buf
             LDA    #03
             STA    Block_Buf+2

             JSR    Find_Time              ; Insert the time info

             LDX    #2A
Next2        LDA    Directory,X
             STA    Block_Buf,X
             DEX
             BNE    Next2                  ; Insert directory info

             SOS    D_Write, Write_List    ; Write Block 2
             BNE    Err1

             LDX    #2A
             LDA    #00
Next3        STA    Block_Buf4,X
             DEX
             BNE    Next3                  ; Zero the directory info

             LDA    #03
             STA    Block_num
             LDA    #00
             STA    Block_num+1
             LDA    #02
             STA    Block_Buf
             LDA    #04
             STA    Block_Buf+2            ; Setup to write block 3
             SOS    D_Write, Write_List    ; Write Block 3
             BNE    Err1

             LDA    #04
             STA    Block_num
             LDA    #00
             STA    Block_num+1
             LDA    #03
             STA    Block_Buf
             LDA    #05
             STA    Block_Buf+2            ; Setup to write block 4
             SOS    D_Write, Write_List    ; Write Block 4
             BNE    Err2

             LDA    #05
             STA    Block_num
             LDA    #00
             STA    Block_num+1
             LDA    #04
             STA    Block_Buf
             LDA    #00
             STA    Block_Buf+2            ; Setup to write block 5
             SOS    D_Write, Write_List    ; Write Block 5
             BNE    Err2

             LDX    Bytes_To_Write
             LDA    #0FF
Next4        STA    Block_Buf,X
             DEX
             BNE    Next4                  ; Setup the bitmap
             LDA    #01
             STA    Block_Buf

             LDA    #06
             STA    Block_num
             LDA    #00
             STA    Block_num+1

             SOS    D_Write, Write_List    ; Write Block 6
             BNE    Err2
             RTS

Err2                                       ; Again, if there is an
             STA    Error_Code             ; error, store the code
             RTS                           ; and return.

Pointer      .BYTE  00, 00                 ; Paramter list for the
Year10       .BYTE  00                     ; SOS GET_TIME call below.
Year1        .BYTE  00
Month10      .BYTE  00
Month1       .BYTE  00
```

```
Day10           .BYTE   00
Day1            .BYTE   00
Ignored1        .BYTE   00
Hour10          .BYTE   00
Hour1           .BYTE   00
Minute10        .BYTE   00
Minute1         .BYTE   00
Ignored2        .BYTE   00, 00, 00, 00, 00

Time_List       .BYTE   01
                .WORD   Pointer

TempYear        .BYTE   00                      ; Temporary locations used
TempMonth       .BYTE   00                      ; in the routine 'Do_Times'.
TempDay         .BYTE   00
TempHour        .BYTE   00
TempMinute      .BYTE   00

Find_Time       SOS     Get_Time, Time_List
                SEC
                LDX     #17.                    ; Convert from ASCII to
Loop            LDA     Pointer,X               ; decimal by subtracting
                SBC     #30                     ; $30.
                STA     Pointer,X
                DEX
                BPL     Loop
                JSR     Do_Times
                RTS

Do_Times
                LDA     Year10                  ; Multiply the 10s position
                ASL     A                       ; by 10 using the formula -
                STA     TempYear                ; 10x = 2x + 8x
                ASL     A                       ; The ASL command multiplies
                ASL     A                       ; by 2 so by doing it a few
                CLC                             ; times we get the result.
                ADC     TempYear
                ADC     Year1                   ; Add the 1s position to get
                STA     TempYear                ; the number, and store it.

                LDA     Month10                 ; Repeat for the month, day
                ASL     A                       ; hour and minute as they
                STA     TempMonth               ; all contain 2 digits.
                ASL     A
                ASL     A
                CLC
                ADC     TempMonth
                ADC     Month1
                STA     TempMonth
                LDA     Day10
                ASL     A
                STA     TempDay
                ASL     A
                ASL     A
                CLC
                ADC     TempDay
                ADC     Day1
                STA     TempDay

                LDA     Hour10
                ASL     A
                STA     TempHour
                ASL     A
                ASL     A
                CLC
                ADC     TempHour
                ADC     Hour1
                STA     TimeType+3

                LDA     Minute10
                ASL     A
                STA     TempMinute
                ASL     A
                ASL     A
                CLC
                ADC     TempMinute
                ADC     Minute1
                STA     TimeType+2

                LDA     TempMonth               ; Convert the 3 byte Day,
                ROL     A                       ; Month, and Year information
                ROL     A                       ; into 2 bytes and store it.
                ROL     A
                ROL     A
                ROL     A
                ORA     TempDay
                STA     TimeType+0
                LDA     #00
                ROL     A
                STA     TimeType+1

                LDA     TempYear
                ASL     A
                ADC     TimeType+1
                STA     TimeType+1

                RTS

                .END                    ; Of Assembly
```

Assembling (ON) the ///: Program Listing #2

```
10   REM *****************************************************************
20   REM *  Formatting Utility: Documentation and      -------------------  *
30   REM *                      Test Program           | Copyright 1983 by |  *
40   REM *  by Martin Nichols   (Basic Version)        |  O N   T H R E E  |  *
50   REM *  ------------------------------------       |  June-July, 1983  |  *
60   REM *                                             -------------------  *
70   REM *  This program demonstrates how easy it is to use the 'FORMAT'    *
80   REM *  invokable module from Business Basic.                           *
90   REM *                                                                  *
100   REM *  Your Basic programs can now format disks whenever needed and   *
110   REM *  thus will not force the user to boot the System Utilities      *
120   REM *  diskette to perform this function.                             *
130   REM ****************************************************************
150   a.to.z$="ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz"
160   a.to.z.and.special$=a.to.z$+"1234567890."
170   valid.disks$=".D1.D2.D3.D4.d1.d2.d3.d4"
180   INVOKE"FORMAT.INV":REM Load the Invokable Module
200   TEXT:Screen.Off$=CHR$(14):Clear.Screen=29
210   Title$=Screen.Off$+"FORMAT INVOKABLE MODULE Documentation & Test"
220   GOSUB 510:VPOS=4
230   PRINT"Before any Invokable Module can be used, it must be loaded";
240   PRINT" into the system by   the following Command Format:"
250   PRINT:PRINT")INVOKE FORMAT.INV":PRINT:PRINT"where FORMAT.INV can be";
260   PRINT" the name of this or any other Invokable Module.":PRINT:GOSUB 500
290   Title$=Screen.Off$+"FORMAT INVOKABLE MODULE Documentation & Test"
300   GOSUB 510:VPOS=4
310   PRINT:PRINT TAB(8);"Select one of the following options:":PRINT
320   PRINT TAB(10);"1.  Format a disk"
330   PRINT TAB(10);"2.  Documentation"
340   PRINT TAB(10);"3.  Catalog a disk"
350   PRINT TAB(10);"4.  Quit":PRINT
360   PRINT TAB(8);"Which option ";:INPUT a$:x=CONV(LEFT$(a$,2))
370   IF x<0 THEN x=0
400   ON x GOTO 1000,2000,3000,420
410   PRINT TAB(8);"Please enter 1, 2, 3 or 4":VPOS= VPOS-2:GOTO 360
420   HOME:PRINT"Bye!":PRINT:END
500   VPOS=24:PRINT USING"79c";"Press any key to Continue";:GET a$:RETURN
510   HOME:PRINT USING"79c";Title$:PRINT:RETURN
1000   REM --- Format a disk ---
1010   Title$=Screen.Off$+"-- Format a Disk --":GOSUB 510
1020   VPOS=4:PRINT"Enter the device whose disk you want to format:"
1030   PRINT"(.D1 - .D4) or RETURN to exit ";:INPUT disk$
1040   IF LEN(disk$)=0 THEN GOSUB 500:GOTO 290
1050   IF INSTR(valid.disks$,disk$)=0 THEN VPOS=8:PRINT CHR$(7);"Cannot
       format ";disk$:GOSUB 500:GOTO 290
1060   ON ERR GOTO 1100
1070   OPEN#1,disk$
1080   GOSUB 1300
1100   OFF ERR:CLOSE
1110   PRINT:PRINT"Enter the new name of the disk ";:INPUT name$
1120   IF LEN(name$)>15 THEN GOTO 1399
1130   IF INSTR(a.to.z$,LEFT$(name$,1)) THEN 1140:ELSE GOTO 1399
1140   FOR a=1 TO LEN(name$):char$=MID$(name$,a,1)
1150     IF INSTR(a.to.z.and.special$,char$) THEN NEXT:ELSE GOTO 1399
1160   PRINT:PRINT"Enter the volume number (0-255) ";:INPUT volume.num%
1170   disk.num%=CONV(RIGHT$(disk$,1))
1180   VPOS=16:PRINT"Formatting..."
1190   PERFORM FormatDisk(%disk.num%,%volume.num%,@er%)
1200   IF er%=0 THEN RENAME disk$,"/"+name$:GOTO 1299
1210   VPOS=16:PRINT CHR$(7);"Error while formatting, ";
1220   IF er%=16 THEN PRINT"device ";disk$;" or .FMTD";d.num;" not configured
       into your system."
1230   IF er%=39 THEN PRINT"no diskette in the device ";disk$
1240   IF er%=43 THEN PRINT"diskette in device ";disk$;" write protected."
1250   IF er%<>16 AND er%<>39 AND er%<>43 THEN PRINT"unspecified type -
       error #";er%
1260   GOSUB 500:GOTO 290:REM Go back to the Menu
1299   VPOS=16:HPOS=11:PRINT" successful":GOSUB 500:GOTO 290
1300   PRINT:PRINT CHR$(7);"This disk is already formatted!"
1310   PRINT:PRINT"Do you really want to format it (Y/N) ";:GET a$:PRINT a$
1320   IF INSTR("Yy",a$) THEN RETURN
1330   POP:GOSUB 500:GOTO 290:REM Go back to the Main Menu
1399   GOSUB 1400:GOSUB 500:GOTO 290:REM Go back to the Main Menu
1400   PRINT:PRINT CHR$(7);"Bad volume name!"
1410   PRINT:PRINT"It must begin with a letter (A-Z), then the rest can be";
1420   PRINT" any combination of      letters and periods ('.').  It must";
1430   PRINT" also be less than 16 characters in length."
1440   RETURN
2000   REM --- Documentation ---
2010   Title$=Screen.Off$+"-- Documentation --":GOSUB 510
2020   VPOS=4:PRINT"The FORMAT invokable module allows all Basic programs";
2030   PRINT" to format a disk.  You    will never again have to force the";
2040   PRINT" user to boot the System Utilities Diskette   to perform this";
2045   PRINT" function"
2050   PRINT:PRINT"To use, enter the following statement:":PRINT
2060   PRINT")PERFORM FORMATDISK (%disk.num%,%volume.num%,@er%)":PRINT
2070   PRINT"The variable 'disk.num%' refers to the number of the floppy";
2080   PRINT" disk that you want  to format (1 - 4 for .D1 - .D4).":PRINT
2090   PRINT"The variable 'volume.num%' is the volume number that you want";
2100   PRINT" on the disk you   are formatting.  It will appear on the";
2110   PRINT" directory listings from Business Basic   and Apple Writer ///."
2120   PRINT:PRINT"The variable 'er%' is used to return the number of any";
2130   PRINT" error that occurs during the formatting process."
2199   GOSUB 500:REM Get keypress to go to the next page
2200   Title$=Screen.Off$+"-- Documentation --":GOSUB 510
```

```
2210   VPOS=4:PRINT"There are only a few error codes that will ever be";
2220   PRINT" returned by the routine."
2230   PRINT:PRINT"Error #16: This means that either the disk (.D2 - .D4) or";
2240   PRINT" the formatting device"; SPC(12);"driver (.FMTD1 - .FMTD4) is";
2250   PRINT" not configured into the SOS.DRIVER file"; SPC(13);"on the";
2260   PRINT TAB(13);" boot disk.":PRINT
2270   PRINT"Error #39: This means that an I/O error has occured during";
2280   PRINT" the formatting":PRINT TAB(12);"process.  Possible causes";
2290   PRINT" include a bad diskette, no diskette in the"
2300   PRINT TAB(12);"disk drive, or the door of the drive is open.":PRINT
2310   PRINT"Error #43: This means that the disk in the disk drive is write-";
2320   PRINT"protected and can"; SPC(11);"not be formatted."
2330   PRINT:PRINT"As soon as you call the routine, formatting begins.  Thus";
2340   PRINT" check to see if the   disk is already formatted BEFORE";
2350   PRINT" performing the routine.  Otherwise you could   erase a disk";
2360   PRINT" with important data on it."
2399   GOSUB 500:REM Get keypress to go to the next page
2400   Title$=Screen.Off$+"-- Documentation --":GOSUB 510
2410   VPOS=4:PRINT"If the formatting process encounters no errors, the disk";
2420   PRINT" that was formatted willhave the name 'BLANK.DISK'.  Since you";
2430   PRINT" may want it to have a different name, usethe RENAME command";
2440   PRINT" of Business Basic like this:":PRINT
2450   PRINT")RENAME .Dx, /NEW.DISK.NAME":PRINT
2460   PRINT"Where '.Dx' is the name of the disk drive (.D1 - .D4) that";
2470   PRINT" holds the disk you   want to rename, and '/NEW.DISK.NAME' is";
2480   PRINT" the new name.  Remember to add the '/'  before the name or";
2490   PRINT" the RENAME command will not work.":PRINT
2500   PRINT"This routine will also correctly format all of the Micro-Sci";
2510   PRINT" disk drives,        including the A73 and A143 high density";
2520   PRINT" disk drives."
2999   GOSUB 500:GOTO 290:REM Go back to the Menu
3000   REM --- Catalog a disk ---
3010   Title$=Screen.Off$+"-- Catalog a Disk --":GOSUB 510
3020   VPOS=4:PRINT"Enter the directory you want to catalog ";:INPUT path$
3030   PRINT:ON ERR GOTO 3199
3040   OPEN#1 AS INPUT,path$
3050   ON EOF#1 GOTO 3200
3060   INPUT#1;line$
3070   PRINT line$
3080   IF VPOS=23 THEN GOSUB 500:VPOS=5:PRINT CHR$(Clear.Screen)
3090   GOTO 3060
3199   OFF ERR:PRINT CHR$(7);"Error in cataloging ";path$;" - bad pathname."
3200   CLOSE:GOSUB 500:GOTO 290:REM Go back to the Menu
```

Assembling (ON) the ///: Program Listing #3

```
;
; Rename a Volume Utility
;
; -----------------------
;
; This routine enables a Pascal program to rename the disk in the
; drives .D1 - .D4.  It is used by the Pascal Formatting Utility
; to put the correct name on the diskette that was just formatted.
; NOTE: This routine will NOT work in Basic, only Pascal.
;

                .MACRO  POP                ; Pull a word from the stack
                PLA
                STA     %1
                PLA
                STA     %1+1
                .ENDM

                .MACRO  PUSH               ; Push it back on
                LDA     %1+1
                PHA
                LDA     %1
                PHA
                .ENDM

                .MACRO  SOS                ; Macro def for SOS call block
                BRK                        ; Begin SOS call block
                .BYTE   %1                 ; call_num
                .WORD   %2                 ; parameter_list pointer
                .ENDM                      ; end of macro definition

Rename          .EQU    0C2                ; call_num for RENAME

Pseudo          .EQU    0E0

;
; Use the following notation to call this routine from Pascal only!
;
; Rename_Volume (Disk_num, New_Name);
;
; Where 'Disk_num' is the number of the disk drive (1-4 for .D1-.D4),
; and New_Name is a string of up to 15 characters.
;

                .PROC   Rename_Volume,2 ; Two Parameters for this routine

                JMP     Start              ; Go around the buffers

Old_Name        .BYTE   03
                .ASCII  ".D"
Old_Name_Suffix .BYTE   00

New_Name        .BYTE   00                 ; Leave room for a string
                .BLOCK  15.,00             ; of up to 15 characters.

Rename_List     .BYTE   02
                .WORD   Old_Name
                .WORD   New_Name

Return          .WORD   00                 ; Put the return address here


Start           POP     Return             ; Save return address
                PLA                        ; Get address of the new volume name
                STA     Pseudo             ; and store it in the right locations
                PLA
                STA     Pseudo+1
                PLA                        ; Get the disk number to rename
                ADC     #30                ; Convert to ascii
                STA     Old_Name_Suffix ; Put it in the proper place
                PLA                        ; Remove the MSB and discard it

                LDY     #0
Again           LDA     @Pseudo,Y
                STA     New_Name,Y
                INY
                CPY     #16.
                BCC     Again

                SOS     Rename,Rename_List
                PUSH    Return
                RTS                        ; Come on back

                .END
```

Assembling (ON) the ///: Program Listing #4

```
PROGRAM Format_Test;

{ ******************************************************************** }
{ *  Formatting Utility:  Test Program           -----------------  * }
{ *                       (Pascal Version)     | Copyright 1983 by | * }
{ *  by Martin Nichols                         |  O N   T H R E E  | * }
{ *  -----------------------------------       |  June-July, 1983   | * }
{ *                                              -----------------  * }
{ *  This program demonstrates how easy it is to use the 'FORMAT'   * }
{ *  assembly language routine from Pascal.                         * }
{ *                                                                  * }
{ *  Your Pascal programs can now format disks whenever needed and  * }
{ *  thus will not force the user to boot the System Utilities      * }
{ *  diskette to perform this function.                              * }
{ ******************************************************************** }

CONST Bell = 7;
      Clear_viewport = 28;

VAR Response: CHAR;
    A_to_Z: SET OF CHAR;
    A_to_Z_and_Special: SET OF CHAR;
    Valid_Nums: SET OF CHAR;
    Valid_Disks : STRING;

PROCEDURE Format_Disk (Disk_num, Volume_num: INTEGER; VAR error: INTEGER);
                    EXTERNAL;

PROCEDURE Rename_Volume (old_disk: INTEGER; VAR New_Name: STRING);
                      EXTERNAL;

PROCEDURE Set_Up;
BEGIN
  Response := ' ';
  A_to_Z := ['A'..'Z', 'a'..'z'];
  A_to_Z_and_Special := ['A'..'Z', 'a'..'z', '0'..'9', '.'];
  Valid_Nums := ['1', '2'];
  Valid_Disks := '.D1.D2.D3.D4.d1.d2.d3.d4'
END; { Of PROCEDURE Set_Up }


PROCEDURE Do_Menu;
BEGIN
  WRITE (CHR (Clear_viewport));
  GOTOXY (24, 0);
  WRITE ('Formatting Utility Test Program');
  GOTOXY (33, 1);
  WRITE ('Pascal Version');
  GOTOXY (7, 3);
  WRITE ('Select one of the following options:');
  GOTOXY (9, 5);
  WRITE ('1.  Format a disk');
  GOTOXY (9, 6);
  WRITE ('2.  Quit')
END; { Of PROCEDURE Do_Menu }


PROCEDURE Get_Option;
BEGIN
  GOTOXY (7, 8);
  WRITE ('Which option');
  REPEAT
    GOTOXY (20, 8);
    UNITREAD (2, Response, 1,, 12)       { * Read one character * }
  UNTIL (Response IN Valid_Nums)         { * from the keyboard. * }
END; { Of PROCEDURE Get_Option }
```

Program Listing Continued on Page 43

Basic — The Easy Way

by Earl Curlson

It's back to work again! I hope that during my lengthy absence you practiced some of the things we were talking about, because we have got a long way to go! Each section (Strings and Input/Output) of this installment of the tutorial shouldn't take more than an hour or so to read through, so take some time and follow along!

Strings

We have already talked somewhat about string variables, but we are now really going to learn how to use them. As always, the only way to follow along is by sitting down at the computer and doing all the examples I give, and more. So boot up your Business Basic disk and read on.

As I stated before, Basic can represent what are called 'Strings' by enclosing the text you want to manipulate within double quote marks. Thus, the statement 'A$ = "This is a test string"' tells the computer to find room in memory for the variable with the name 'A' and the type string. Then the computer assigns the words 'This is a test string' to that variable.

I showed last month all of the different things you can do with the numeric variables. This time we will look at the string functions which in one way or another manipulate strings.

Just as you can add two numbers together, two or more strings can be added to give a new string. Type in 'A$ = "Start"', and 'C$ = "End"'. If you then type 'PRINT A$' and then 'PRINT C$', you will see 'Start' and 'End' on two different lines. Now type 'RESULTS$ = A$ + C$'. What do you think you just did? Test out your hunch by now typing 'PRINT RESULT$'.

You should get 'StartEnd'. The computer simply added the two strings together and stored them in the variable 'RESULT'. Just as you can add more than two numbers together at once, any number of strings can be added together to form a new string.

Type in 'B$ = "Middle"'. This assigns to the string variable 'B' the text 'Middle'. To make sure, type 'PRINT B$' and you should see 'Middle' printed on your screen. Now type in 'RESULT$ = A$ + B$ + C$'. If you did everything right when you type 'PRINT RESULT$' the line of text 'StartMiddleEnd' should appear on your screen.

Adding is nice but how about when you want to subtract one string from another? Say you wanted to delete the text 'End' off of the string 'RESULT'. How do you do it? Well, since you added numbers together with the '+' and subtracted them with the '-', and since we add strings together with the '+', would the '-' subtract strings from strings? Let's find out!

Type 'PRINT RESULT$ - "End" and what do you get? Ouch!?! A Syntax Error! Keeping away from a detailed explanation of why it doesn't work, suffice it to say that it doesn't. You can only add strings together, the '-*/' operators have no meaning when working with strings and therefore don't work.

If all you could do is add one string to another, things would get boring quickly. If you remember back that far, last time I showed you all sorts of functions for numeric variables that expanded on just adding, subtracting etc. Strings have a set of their own functions which I will attempt to explain in the next few paragraphs.

The Reference Manual tells about String Functions on pages 57-65. The material presented below is not so much a duplication as it is an expansion of those items.

If you are working with strings it sometimes becomes necessary to know just how long a particular string is — how many characters it contains. The string function 'LEN' does this for us. It simply returns the number of characters of the string given as an argument.

Type in 'PRINT LEN (RESULT$)' and if you haven't changed the string 'RESULT' from the last time you used it, you should get a 14 printed on your screen, as the string is 14 characters long ('StartMiddleEnd').

A similar example is shown on page 57 of the Basic Manual. One of the interesting things about strings in Business Basic is that they can't ever become more than 225 characters long. If you want to waste the time, test it out by typing in the following program and then run it. Remember to type 'NEW' before you start entering this program to erase any program that is already in memory.

```
10  A$ = "*"
20  FOR counter = 1 TO 256
30     B$ = B$ + A$
40     PRINT LEN(B$); " ";
50     NEXT counter
```

Once it is run, it will display a bunch of numbers on the screen. These indicate the length of the string as it gets bigger and bigger inside the program loop. It will stop when the length of B$ + A$ gets to be 256 characters because it can't handle that big a string.

Starting at the top, the program assigns the character '*' to the string variable 'A'. It then goes through a loop 256 times and does something interesting. In line #30 the computer adds the string B$ to the string A$ and assigns it to the string B$. Since A$ contains the '*' character and B$ is initially empty, the string B$ is first set to '*' and then the next time through the loop it becomes '**' and so on.

The 256th time through the string is 255 characters long and the program tries to add one more character to it. Since it can't do that it gives you an error message and stops.

Enough of all this lengthy business, let's get back to the good stuff! Since we are all now well versed in the 'LEN' function, we can now move on to some more string functions. Every once in a while you will need to take certain characters from one string and put them in another one. Business Basic has three functions that help you do just this.

The string function 'LEFT$' will return a string that consists of a certain number of the leftmost characters of the given string. The number of characters that are returned is another argument to this function. Shown on page 60 of the manual, you can use it like this:

```
)PRINT LEFT$ ("Appleskin", 5)
Apple
)PRINT LEFT$ ("Sparkling", 3)
```

```
    Spa
    )
```

As you can see, the function in the first line takes the leftmost 5 characters from the string "Appleskin" and prints them. Likewise the next statement takes the leftmost 3 characters from "Sparkling" and prints them.

You can also use this statement to assign the leftmost 'n' characters of one string to another string. Type in RESULT$ LEFT$ ("Bill Jones", 4) and then 'PRINT RESULT$' will get you a line of 'Bill'.

Very similar to the LEFT$ function is RIGHT$. It will return a string composed of a certain number of rightmost characters of the given string. The Basic manual gives examples on page 61. It isn't that much different from the LEFT$ function so I won't delve any deeper into this function.

Perhaps the most useful of these functions is MID$. While the other two only allow you to get a string from either the left or right of another string you will see that there will be a problem if you need to extract certain characters from the middle of the string.

MID$ allows you to take a substring out of a given string at any position within that string. All you have to do is specify where in the string the function should take the substring from. This isn't the easiest thing to explain so let's go to an example.

Type 'PRINT MID$ ("The sky is blue today", 12)' and the text 'blue today' will be printed on your screen. If you only wanted to extract the words 'blue' you can do that too! Just type 'PRINT MID$ ("The sky is blue today", 12, 4)'. Just as with the other string functions the strings returned by 'MID$' can be assigned to a string variable. Thus you could have 'RESULT$ MID$ ("The sky is blue today", 12, 4)' and if you then typed 'PRINT RESULT$', the line 'blue' would appear.

Now that we know how to get information out of strings it becomes even more important to be able to find information within strings. Apple /// Business Basic has a function that does this and it is called 'INSTR'.

This function will return the position within a string of another string. For example, remember how we did the example on the 'MID$' function above? Well, we just looked at the string for the test 'blue' and found that it started at the twelfth position within the string. Since computers can't really see that well it couldn't do the same thing.

However, with 'INSTR' it can perform the same function. Type in 'PRINT INSTR ("The sky is blue today", "blue")' and the computer will respond with the number 12. You can then use that number to extract the section of text you want from the string.

Now try 'PRINT INSTR ("The sky is blue today", "black")' and the computer should respond with a zero. This indicates that the computer could not find the text "black" in the other string. This is a very powerful function indeed!

Isn't this great? We now know almost everything about strings and string functions! Just a couple more to go, and one of them is the 'SUB$' function. It allows you to substitute a portion of one string with another. The Basic manual has many examples of this routine on pages 64 and 65. It is also very powerful and useful in many respects.

Another very useful routine is the 'VAL' function. It converts a string value into the numeric equivalent. For example type in 'PRINT VAL ("3.21")' and the string within the quotes "3.21" will be converted to a number and printed on the screen. You can also use it in an assignment statement like this: 'num VAL ("3.21")' and then if you type 'PRINT num', 3.21 will be displayed.

After we learn a little bit about the Business Basic I/O (input/output) we will use the 'VAL' function in a program (next time!). Until then remember it is a very helpful tool for programmers, which we will use later on.

Input/Output

This section of the tutorial will concern itself with Apple /// Business Basic I/O, which is a fairly complicated subject. Believe it or not, you have already learned one important aspect of I/O in Business Basic. Remember the 'PRINT' statement? The 'PRINT' statement is used to display text and numbers on the display screen. We have already done this in past lessons so you may want to skip the next few paragraphs.

Pages 74 through 77 of the Basic manual describe the 'PRINT' statement and some of the things you can do with it. You can type in 'A$ = "This is some text"' and then 'PRINT A$' and the computer will respond with the line 'This is some text'. Likewise enter 'A = 3.14159' and the 'PRINT A' and it will come back with '3.14159'. You can also print out integers by typing 'A% —502' and then 'PRINT A%'.

The semicolon and the comma are used by the 'PRINT' statement to format the way things appear on the screen. If items in a 'PRINT' statement are separated by a semicolon, no spaces will be printed between them when they are printed on the screen. For example, type in 'PRINT "Here is a number"; 11'. You will get the line of text 'Here is a number11' on your screen because the semicolon told the computer to just print the 11 directly after the line of text.

Now try 'PRINT "Here is a number", 11'. The 11 should now be 16 spaces to the right of the 'r' in 'number'. If you read over page 74 you will see that the text screen is divided (in Basic) into five 16-character-wide tab fields. To illustrate this, type in 'PRINT 1, 2, 3, 4, 5'. The numbers 1 through 5 neatly fill the screen, but it looks like there may be room for one number at the right of the screen. Try typing 'PRINT 1, 2, 3, 4, 5, 6'. The 6 appears in the first column of the second row of printed numbers, so using numbers with a comma separating them will give you a maximum of 5 columns per line.

As we just saw, the line 'PRINT "Here is a number", 11' prints out the line of text and then 16 spaces before the number 11 is printed. This is because the text 'Here is a number' is exactly 16 characters long and as such fills up the entire first tab field. Type in 'PRINT "Here is a numbe", 11' and you will get the text 'Here is a numbe 11' printed on the screen. This is because the text 'Here is a numbe' didn't completely fill up the first tab field so the comma caused the 11 to be printed at the first position of the second tab field — one space after 'numbe'.

If you remember back a ways, the 'PRINT' statement can print out strings too. Perhaps you want to print out a bunch of numbers on the screen and don't want to have only 5 items per line using the comma as a separator. What's a poor programmer to do? That's right, just print a space between each number! Try the following line 'PRINT 1;" ";2;" ";3;" ";4;" ";5;' and you should get the numbers 1 through 5 printed on your screen with a single space between them.

Say you needed some numbers printed out at column positions that are different from the pre-defined tab positions that Business Basic gives you. For example, you want to print out numbers in 9 columns across the screen. How do you do it? Or what if you wanted to print out 25 spaces between one item and another?

Page 77 of the Basic Business Manual very briefly describes the way to do these things. You can move to any column on the line you are printing on with the 'TAB' specification of the 'PRINT' statement. For example, type in 'PRINT "Where is the number?"; TAB(80);5' and what do you get? If you typed in everything right you should get the line of text and the number 5 printed in the rightmost column of the screen.

The TAB specification allows you to move to any column on the text screen and print from there with just one limitation. If you typed in 'PRINT "Here it is"; TAB(5);11' you will get one line 'Here it is11' printed on the screen. This is because you told the computer to tab to column number 5 after it had printed out 'Here it is'. Since it had already gone past that column, it couldn't go back so it printed out the 11 directly after the text it had printed.

Thus, you can use the TAB spec. to go to different columns on the screen and print from there. The example we gave earlier on wanting to print in 9 columns across the screen is now easy to do. Type

'PRINT 1;TAB(10);2;TAB(20);3;TAB(30);4;TAB(40);5;TAB(50);6; TAB(60);7;TAB(70);8;TAB(80);9'.

Well, maybe no so easy — but it can be done! You know enough now to try to write a program that does this in an easier to read manner. Go ahead and try to write one. A hint: Remember FOR..NEXT loops?

By now it should be fairly clear on just how to print out numbers (or text) in different columns across a line. There was just one more question we had about the 'PRINT' statement and that was how do we print out say 25 spaces between one item and another without printing out a string of 25 spaces?

The answer is quite simple, use the 'SPC' specification of the 'PRINT' statement. 'SPC' will insert a number of spaces between one item and another. Type in 'PRINT "Here is some text"; SPC(10);"now some more..."' and 'Here is some text' will be printed on the screen, followed by 10 spaces and then the text 'and now some more...'.

That's enough of 'PRINT' for now, since we know quite a bit about 'Output', let's now look at how to get information into our program — this is the 'input' part. Look at the following program:

```
 10  years = 36
 50  days = years * 365
100  PRINT "You have been around about ";days;" days"
```

If the user changes line 10 to indicate how many years old he or she was, the program (once RUN) would tell about how many days they have been around. Since changing the program every time another person wanted to use it would be time consuming, we need to find another way to set the value of the variable 'years'.

The 'INPUT' statement is just what we're looking for. It will take a number that you type in and assign it to a variable. Try the following updated program:

```
 10  PRINT 'Enter your age in years"
 20  INPUT years
 50  days = years * 365
100  PRINT "You have been around about ";days;" days"
```

As explained on pages 78 through 81 of the Basic manual, the 'INPUT' statement will display a question mark on the screen and wait for you to enter something. If you put a semicolon after the end of the 'PRINT' statement in line 10, when the program is RUN the question mark will follow the first line and will not be on another line. Look over the following program:

```
 10  INPUT "Enter your age in years"; years
 50  days = years * 365
100  PRINT "You have been around about ";days;" days"
```

This shows that a string may be included in the 'INPUT' statement, and it will be printed out before it prompts you with the question mark for a number. As shown on those pages of the manual, there are many combinations of things that you can do with the 'INPUT' statement. Practice some of these to become more proficient in the use of getting information into your programs.

One thing that the manuals don't cover well enough is using the 'INPUT' statement to enter strings. It really isn't different from using the 'INPUT' statement to enter numbers so the following program should answer your questions. There are a couple of strange things that you have to watch for and these are explained on pages 80 and 81 of the Basic manual. Remember to type 'NEW' before you enter this program to erase the last program you wrote.

```
10  INPUT "What is your name "; name$
20  length = LEN (name$)
30  PRINT name$;" your name is ";length;" letters long."
```

When programming it is sometimes best to input just a single character from the user, as in the case of answering a Yes or No question. The 'GET' statement allows you to do this. It is described on pages 81 and 82. The following program uses it.

```
 10  PRINT "Are you alive (answer Y or N) ?";
 20  GET life$: PRINT life$
 30  IF life$= "y" OR life$= "Y" THEN 100
 40  IF life$= "n" OR life$= "N" THEN 200
 50  GOTO 10
100  PRINT "That's good!"
110  END
200  PRINT "How are you typing then!?!"
210  END
```

Well we're doing pretty good now, we've learned quite a bit about both input and output in Apple /// Business Basic. Just a little more and we will have a good idea of the fundamental aspects of Basic on the ///. Remember why we learned how to input information using the 'INPUT' and 'GET' statements? That's right, so we didn't have to type it into the program each time it changed.

While those statements are good, it is sometimes necessary to store information within a program so that it can later use it. The statements 'DATA', 'READ', and 'RESTORE' allow you to do this. Described on pages 82 through 86 they provide the programmer with another way of getting data into their programs.

Simply put, the 'DATA' statements hold a bunch of pre-defined pieces of information that can be inputted into a program with the 'READ' statement. Thus, the 'READ' statement is just like the 'INPUT' statement, except it takes information from 'DATA' statements instead of from what the user types in. Look over the

Continued on page 43

Products Received

The products outlined below have been received by ON THREE for the purpose of review. Some have been reviewed in the past and many will be reviewed in the future. The products have all been given the ON THREE 'stamp of approval'. This is only an indication that a product works as advertised and is not an endorsement of the product by ON THREE.

SOS Block Access Utility & System Character Set Editor

Al Evans (contributing author of ON THREE) has created a couple of gems with the release of these two utility programs. The SOS Block Access Utility allows you to access any block structured device (ie. disk drive) and to read, display, print, edit and write any given block on that device. A very nice on-line help screen system is always there when you need a question answered.

The System Character Set Editor is one of the finest on the market. It allows you to create and modify the characters that are used by the Apple /// text screen. Up to four fonts may be in memory at any one time, and each can be edited independently. Fonts that you create can be saved on disk or installed in the system font directly. This program also has an on-line help screen system that aids you in the mastery of the program.

Both these products are menu driven (similar to the System Utilities Program) and very easy to use. These programs are written in Pascal and for those of you who want to make any modifications, fully commented source listings are provided with each package.

Both products require Pascal and are priced at $40 each. They may be ordered directly from PowerTools, 1206 Karen Avenue, Austin, Texas 78757.

MICROTEK Dumpling-GX & Dumpling Spooler & Alpine Printer Driver

The MICROTEK printer interface cards used with the Alpine printer driver allows the Apple /// to connect with a variety of parallel printers. In particular, the MICROTEK Dumpling Spooler interface allows you to dump vast quantities of data into the interface buffer for later printing.

Combined with the Alpine Apple /// interface software, the MICROTEK interface cards allow the printing of text, graphics and screen dumps on your parallel printer. It even works in emulation mode.

MICROTEK, Inc., 9514 Chesapeake Drive, San Diego, California 92123. (619) 569-0900. Alpine Computing, Inc., 851 North Main, Logan, Utah 84321. (801) 752-6432.

Programmer's Power Tools ///

PPT /// is a powerful set of utilities for the programmer that works in Business Basic. When incorporated into your programs, PPT /// gives you instant access to routines for sorting, searching, and sophisticated data entry. Each PPT /// command accomplishes in a single program statement what would require many lines of code in Business Basic. All of these commands operate in machine language for the additional benefit of speed.

With the invokable modules supplied with this package, your Basic programs will have access to 13 new routines that will make your program development much easier. The ability to do very fast searches and sorts through large string and numeric arrays, locking-out and unlocking the RESET key and formatting disks are now within reach of the average programmer.

This is a very nice utility package for Apple /// Business Basic programmers as it cuts down program development time. CE Software will let you integrate these routines into the programs you sell for a small licensing fee. Source listings of the routines may also be purchased.

Programmer's Power Tools /// has a suggested list price of $79.95 and can be ordered from any dealer that carries the CE Software product line.

CE Software, 801 73rd Street, Des Moines, Iowa 50312. (515) 224-1992.

STOCK PORTFOLIO SYSTEM

The STOCK PORTFOLIO SYSTEM by Smith Micro Software is a personal investment accounting, record keeping and control system designed to provide you with the facts you need to make informed investment decisions. It covers a wide range of investments, including stocks, bonds, options, multiple CD, bank, credit union and money market accounts.

It is menu driven and very flexible. You can let the STOCK PORTFOLIO SYSTEM access the Dow/Jones Retrieval service to get stock quotes. The terminal mode allows you to reach out beyond your Apple for direct access to the nation's financial news and other services.

The STOCK PORTFOLIO SYSTEM has a suggested retail price of $185 and is available from Smith Micro Software.

Smith Micro Software, P.O. Box 604, Sunset Beach, California 90742. (213) 592-1032.

VersaForm

VersaForm is the Business Form Processor. Utilizing the familiarity and structure of existing paper forms, VersaForm accelerates the speed and accuracy of processing information. Because it emulates current paper handling procedures, VersaForm is very easy to tailor to specific uses.

This package makes it possible for you to do your business date handling tasks on a computer, using what you already know. Instead of requiring that you enter the world of the programmer, VersaForm works in the familiar world of business forms and deals with your data just they way you do now — on paper, but with the speed and power of a computer.

A complicated system, the VersaForm package takes a while to get to know, but it is worth it.

Applied Software Technology, 14125 Capri Drive, Suite 4, Los Gatos, California 95030. (408) 370-2662.

PFS: FILE & REPORT

With PFS: File you can create a file, search and update any item of group of items in the file, and print sorted information. Information management at its best, these programs are extremely easy to use.

All PFS products are designed so that a novice can master them in less than an hour. Reviewed in the January issue of ON THREE, these programs received an A- and a B- respectively.

Available from most authorized Apple dealers, these programs are made by Software Publishing Corporation and are priced at $175 and $125 respectively, for the Apple ///.

Software Publishing Corporation, 1901 Landings Drive, Mountain View, California 94043. (415) 962-8910.

QUICK & EASY DATA MASTER

Quick & Easy Data Master is a program that creates custom applications software and report forms designed to your specifications. This package creates an unprotected Business Basic data base program as per your specifications.

The ideal data base program is one that you can design exactly the way you want it: prompts, edits, error messages, headers, titles, computed data, interactive files, report forms, etc. to your specifications. You design it and Quick & Easy will create it for you.

Intended for the more serious computer user who knows how to program in Basic, this package is not very hard to use, but it does require some thought. Sold by Advanced Software Technology, Inc., it is priced at $69.95. Reviewed in the April-May issue, it gets a C.

Advanced Software Technology, Inc., 7899 Mastin Drive, Overland Park, Kansas 66204. (913) 648-4442.

CRITICAL PATH SCHEDULING

If you are involved in project management and tired of the hassels of project scheduling, the Critical Path Scheduling System is for you! It is a management tool for defining and analyzing the overall concepts of a project and provides a powerful method for scheduling the many tasks necessary to complete the project ON TIME AT THE LOWEST POSSIBLE COST.

Armed with the information that this system provides, the manager is better prepared to make decisions regarding the impact any tasks will have on the project and permits him to be instrumental in guiding the project rather than just monitoring its progress.

This is a very 'User Friendly' system, and it has a good tutorial/user manual. Comprehensive reports make a manager's life a lot easier. Developed by Great Divide Software, it has a suggested retail price of $495. Reviewed in the April-May issue, CRITICAL PATH SCHEDULING gets a B.

GL-PLUS

To many managers, accounting and the preparation of financial reports are time consuming chores that have to be struggled through. But now, at last, accounting can be simplified.

GL-PLUS is an accounting system designed for the Apple /// computer. It is a flexible, easy to use, journal-based General Ledger system. The computer and GL-PLUS combine to provide you with a tool. A tool to make your accounting chores easier. GL-PLUS automatically guides you through entries and then automatically sorts and posts them.

Report preparation is a "snap" with GL-PLUS. You select the report you wish and the rest is done automatically. GL-PLUS includes a PLUS. The PLUS is a built-in accounts receivable and accounts payable capability that can be implemented anytime you desire.

Another 'User Friendly' system, flexible reporting and ease of use make an excellent accounting package. Developed by Great Divide Software, it has a suggested retail price of $495.

Great Divide Software, Inc., 8060 West Woodard Drive, Lakewood, Colorado 80227. (303) 337-0383.

PKASO ///

The PKASO /// printer interface system is a hardware and software device that allows the Apple /// to operate with just about any dot matrix printer in both native and emulation mode. It gives the user the option of printing text, graphics or even screen dumps on your parallel printer.

A very nice printer interface, the system has a suggested retail price of $205. Reviewed in the April-May issue, the PKASO /// printer interface gets an A.

Interactive Structures, Inc., P.O. Box 404, Bala Cynwyd, Pennsylvania 19004. (215) 667-1713.

Continued From Page 41

following example program and you should start to see what I mean:

```
10   DATA PI    , 3.14159, EXP (1) = , 2.1728
20   READ a$, a, b$, b
30   PRINT a$; a, b$; b
```

The first and third elements of the DATA' statement in line 10 are 'READ' by the program as strings and are assigned to the variables a$ and b$. The second and fourth elements are 'READ' as real numbers and are stored in the variables a and b. The program is fairly simple yet it shows just what a powerful tool these two statements can be.

The 'RESTORE' statement is described on page 86 of the manual and it allows you to read the same data more than once. Since we aren't going to delve very deeply into 'DATA' and 'READ' we won't be mentioning 'RESTORE' for now. But you never know what we are going to do in the future, so practice!

The Future

We're done for now! Next time we will do that 'HELLO' program I've been promising for some time and even take a look at files and what they can do for you. If you want to work ahead, glance over the parts of the manual concerning File I/O. We won't be doing all of chapter 5 next time, but if you like you can go through it all. ///

Continued from page 38

```
PROCEDURE Bottom_Prompt;
VAR ch: CHAR;

BEGIN
  GOTOXY (28, 22);
  WRITE ('Press any key to Continue');
  READ (ch)
END; { Of PROCEDURE Bottom_Prompt }


PROCEDURE Do_Option;
VAR New_Name, Disk: STRING;
    Disk_num, volume_num, error: INTEGER;


  PROCEDURE Disk_Formatted;
  VAR ch: CHAR;

  BEGIN
    WRITELN;
    WRITELN (CHR (Bell), 'This disk is already formatted!');
    WRITELN;
    WRITE ('Do you really want to format it (Y/N) ');
    READ (ch);
    WRITELN;
    IF (ch IN ['Y', 'y']) THEN
      EXIT (Disk_Formatted)
    ELSE
      EXIT (Do_Option)
  END; { Of PROCEDURE Disk_Formatted }


  PROCEDURE Test_Disk_Name;      { Check to see if it's a valid device }
  VAR File_ID: FILE OF CHAR;

  BEGIN
    IF (LENGTH (Disk) = 0) THEN
      BEGIN
```

Program Listing Continued on Page 31

ON THREE Author Guidelines

Due to popular demand we are including our Author Guidelines in this issue. If you have been putting off sending in an article, please wait no more! We need all the help we can get to go to a monthly format, so send it in today. The Author Guidelines follow in their original unadulterated form.

ON THREE Author Guidelines

ON THREE is the only information source that deals exclusively with the Apple ///. As such we must be a 'Jack Of All Trades' when it comes to article submissions. What it boils down to is this - We will accept and print almost any material submitted that is connected with the ///.

Our pay rates are quite high - $75 per printed page, with an additional royalty from the sales of the disk of the month. This royalty will be determined using a weighted scale dependent on the number and quality of the programs on that months disk.

For example, if your article contains a program that is included on that months diskette, and if it was the feature article, you could expect a royalty of about 20% of the net sales of that disk.

Exceptional programs with commercial applications will be considered on a case by case basis, so if you want to market a software package - come to us. It may pay you to try us, advertising rates being what they are today. If you do choose us to market your software, we will not only help you with final program design and user interface, we will make sure your instruction booklet can be read (and understood) by human beings!

Our active participation in the design and documentary work of your software package will enable us to uphold the high quality of software that the /// needs - and its users demand.

Content

As I said before, we will accept most anything that you can come up with. What we don't want are articles telling how some 'Acne-faced' executive in his early twenties made his first million. We will leave those articles for magazines like Softalk. Our readers want content!

Since one of our major goals is to instruct the reader in the art of using their computer, any material submitted MUST instruct the user in one way or another.

Now, the manner of this instruction is very important. Droll, highly technical work may merit publication, but a manuscript containing the same technical material presented in a way that the lay person can understand will be given preference.

In any magazine, at times, some comic or colorful relief is needed. Thus we will always find room for a few short programs without an accompanying article, such as graphic demos. Upon acceptance, these will be paid a flat rate of $25.

Submission Requirements

To borrow a phrase, 'accuracy is the keyword'. If you write an article, telling of some neat new 'Better-Faster-Smarter' way of doing things, it had better work. All programs must be completely debugged and idiot-proofed. Please don't force the user to remember something like - by pressing 'ESCAPE', and then 'CONTROL C' you can catalog a disk! If the user needs to learn special key sequences to operate your program, the least you can do is to include some sort of on-line menu or 'help' screen display.

Material should be submitted typewritten and double spaced to allow for easy editing. Dot-matrix printer output is acceptable. Any diagrams or figures that accompany your article should be in black ink, with typed or typeset labels. We can reduce your drawings, but remember that as a picture shrinks - clarity is lost, so if at all possible send us exactly what needs to be printed.

Programs longer that a page should be accompanied with a diskette. We will soon be able to send copy directly from our computer to the typesetter, so for a longer article we would appreciate you sending it on a disk. Any standard Apple /// textfile will do. Articles written on the Pascal editor are also acceptable. Diskettes will not be returned unless a stamped self-addressed envelope is enclosed.

With your submission please include a cover letter with your name, address, telephone number(s) and time of day that you can be reached. Please include a statement saying that the material submitted is an original work, has not been previously published, and is not currently under consideration by any other publication.

All material should be directed to the above address, marked ATTENTION: EDITOR. If the article you envision is quite large, or if you have dreams of creating a program that does everything but walk the dog, contact me beforehand. Call or write, but please don't put me in the position of turning down your work after you have finished it, I wouldn't want to waste your time.

Acceptance - Publication

Initial acknowledgement will usually be made by form letter. This letter will not normally indicate acceptance or non-acceptance. The decision whether or not to publish will be determined by myself after careful review.

Once that decision is made, we will contact you advising of acceptance or declination. If favorable, at that time you will be given an issue date for possible publication. This date will not be set in stone, due to the complex factors involved in publishing, but we will keep you advised of any changes.

Payment

Payments are based on a flat rate per page, exclusive of advertising space, but including all artwork, titles, program listings etc. This payment also includes reprint rights for any and all special compilations of ON THREE articles. For these publications, each author will be contacted and given an additional payment of approximately 25% of the normal page rate.

Payment will be made upon publishing, and will be mailed along with a copy of the issue in which your article appears. Authors who want additional copies of that issue, may order them for $1.25 each.

Program and Article Tips

With the Apple ///, we are blessed with having the ability to send the output of a program to almost any device that is configured into the system. Therefore, please make your programs as general as possible. Let the user decide if he wants the output sent to the printer, console or even a disk file. Sticking to this suggestion will increase your potential audience and your chances of publication.

Business Basic (and Pascal) allow you to use long, descriptive variable names - so use them! The easier your program is to read, the quicker we will be able to educate those unfortunate souls who don't know 37 different computer languages.

I have done some timings of loops and subroutines in Basic and have found a negligible difference in execution time using long variable names, so there is no excuse not to!

Whether you use Basic or Pascal, use a 'top-down' structured approach in writing your programs. Break the program up into easily understood procedures and functions. This will not only help the reader understand your work, it will aid you in debugging and testing.

Use GOTO's spareingly at most. With the advanced features of the ///, there is no excuse for writing 'spaghetti' code, whose logic and general design is hard to follow.

Document your program with comments. There are times when even the 'greenest' novice will understand a liberally commented program, so do it!

Please write your articles in a clear and concise manner. If you need, include graphs or tables to help make your point. You should decide what type of audience you will be aiming for - before you write the article. Please remember that not everyone is well versed in computer languages and therefore, they may require a lot of explanations before they understand.

That's all folks. I hope you could get enough information out of this sheet to satisfy your curiosity for a while. If you have any other questions, please don't hesitate in giving us a call. We're here to help you, so if you think of something you would like to see, please feel free to write or telephone for additional information.

Apple /// User Groups

If you know of an Apple /// club or user's group that isn't listed here, please write to ON THREE care of this column and we will promptly publish the notice.

TAU (Third Apple Users) Group
T.A.U. is an independent, non-profit organization dedicated to the accumulation and dissemination of information about the Apple ///. Membership fees are currently $12 a year, with a $5 initiation fee. Located in Illinois, T.A.U. has about fifty members and they meet the last Wednesday of each month. They also publish a newsletter containing information about the ///. For those interested in further information, T.A.U.'s address is P.O. Box 72187, Roselle, Illinois 60172.

Original Apple ///rs
The Original Apple ///rs is a user's group in the San Francisco area that holds regular meetings the third Wednesday of each month. Annual membership dues are $25 and may be mailed to: Original Apple ///rs, P.O. Box 813, San Francisco, CA 94101. The Open Apple Gazette newsletter is sent out as part of the membership dues.

Call THREE: Hot Line

As I mentioned in my editorial this month, ON THREE is setting up a Hot Line service whereby Apple /// users with problems can call a relatively local number to get help. Since very few of the dealers are set up to support the ///, we are going to have to help ourselves.

If you know enough about your machine to answer questions, please give me a call or write me a letter, everyone needs your help! If you can answer some questions, in the letter you send me please state what areas of expertise you have and the hours during the week you would be willing to take calls.

We currently can't offer monetary compensation to those of you willing to take calls (it's very hard to verify a call that you RECEIVE), but we can offer advance copies of our future products in a beta test site arrangement.

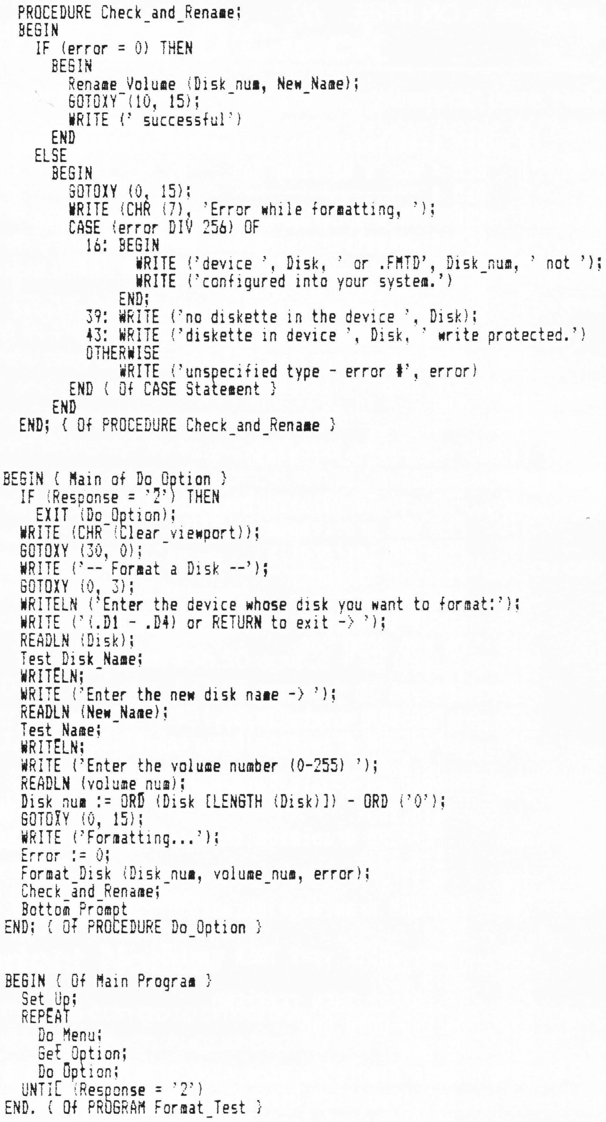
If this interests you please drop us a note. If you've already talked to me over the phone, please write me a note again telling the hours you are available and the topics that you are fluent in. Come October I'd like to fill this entire page with names and phone numbers of people willing to take calls.

Errata: Oops, another mistrake!

Page 11 of the February-March issue of ON THREE contains a small typesetting error that has caused a few headaches. Paragraph four of that page has a segment of text, '00060C00W'. This should read '0006 < 0C00W'. The program method presented in that article will not work without the above correction.

Page 30 of the same issue also contains a mistake. The program line 'PROGRAM Addition Test;' should read 'PROGRAM Addition _Test;' Note the underscore character ('_') between the 'Addition' and the 'Test'. The same type of error is on page 33, and twice on page 35. The Pascal programs will NOT work without these changes.

Continued from page 31



```
  PROCEDURE Check_and_Rename;
  BEGIN
    IF (error = 0) THEN
      BEGIN
        Rename_Volume (Disk_num, New_Name);
        GOTOXY (10, 15);
        WRITE (' successful')
      END
    ELSE
      BEGIN
        GOTOXY (0, 15);
        WRITE (CHR (7), 'Error while formatting, ');
        CASE (error DIV 256) OF
          16: BEGIN
                WRITE ('device ', Disk, ' or .FMTD', Disk_num, ' not ');
                WRITE ('configured into your system.')
              END;
          39: WRITE ('no diskette in the device ', Disk);
          43: WRITE ('diskette in device ', Disk, ' write protected.')
          OTHERWISE
              WRITE ('unspecified type - error #', error)
        END { Of CASE Statement }
      END
  END; { Of PROCEDURE Check_and_Rename }


BEGIN { Main of Do_Option }
  IF (Response = '2') THEN
    EXIT (Do_Option);
  WRITE (CHR (Clear_viewport));
  GOTOXY (30, 0);
  WRITE ('-- Format a Disk --');
  GOTOXY (0, 3);
  WRITELN ('Enter the device whose disk you want to format:');
  WRITE ('(.D1 - .D4) or RETURN to exit -> ');
  READLN (Disk);
  Test_Disk_Name;
  WRITELN;
  WRITE ('Enter the new disk name -> ');
  READLN (New_Name);
  Test_Name;
  WRITELN;
  WRITE ('Enter the volume number (0-255) ');
  READLN (volume_num);
  Disk_num := ORD (Disk [LENGTH (Disk)]) - ORD ('0');
  GOTOXY (0, 15);
  WRITE ('Formatting...');
  Error := 0;
  Format_Disk (Disk_num, volume_num, error);
  Check_and_Rename;
  Bottom_Prompt
END; { Of PROCEDURE Do_Option }


BEGIN { Of Main Program }
  Set_Up;
  REPEAT
    Do_Menu;
    Get_Option;
    Do_Option;
  UNTIL (Response = '2')
END. { Of PROGRAM Format_Test }
```

Three Shorts - Fini!

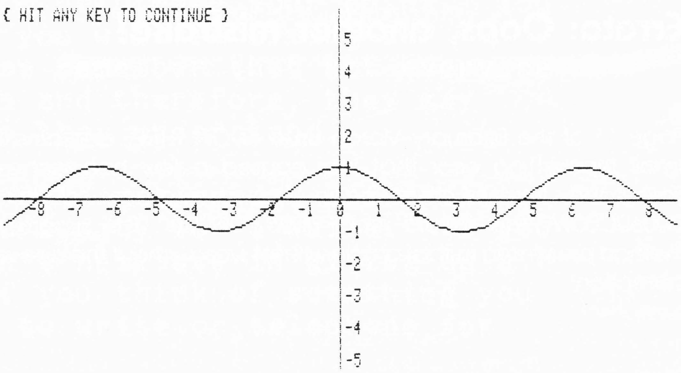
Well, we didn't get three this time, but the two programs below are very interesting and will keep you busy for a little while.

As usual, to use just type them in and save them on disk. Make sure that the '/BASIC' disk is on-line and type "RUN". The '/BASIC' disk is needed because the file "BGRAF.INV" is used. If the programs can't find that file they will hang and you will probably have to reboot.

ON THREE will pay $25 for any short demonstration program used in this space, so send in your favorite today, and we will see you next time in ON THREE. ///

```
0     REM ****************************************
1     REM * Ron's Walking Man -- by Ron Puckett *
2     REM * -----------------                   *
3     REM * This program shows how to do simple *
4     REM * animation using the DRAWIMAGE pro-  *
5     REM * cedure of BGRAF.INV and the TEN()   *
6     REM * function of BASIC.                  *
7     REM ****************************************
10     ON ERR INVOKE"/BASIC/BGRAF.INV"
100    PERFORM initgrafix:OFF ERR
110    DIM source%(68),blank%(22)
120    FOR x=0 TO 68
130      READ a$
140      source%(x)=TEN(a$)
150      NEXT x
160    FOR x=0 TO 22:blank%(x)=0:NEXT
170    PERFORM grafixmode(%1,%1)
180    PERFORM initgrafix
190    PERFORM fillport
200    PERFORM grafixon
205    GOSUB 10000:REM Draw Backround lines
210    ON KBD GOTO 500
215    yy=191
220    FOR x=0 TO 269 STEP 5
230      PERFORM moveto(%x,%yy)
240      PERFORM drawimage(@source%(0),%2,%0,%0,%
         16,%23)
250      PERFORM drawimage(@source%(23),%2,%0,%0,
         %16,%23)
260      PERFORM drawimage(@source%(46),%2,%0,%0,
         %16,%23)
270      NEXT x
280    PERFORM drawimage(@blank%(0),%2,%0,%0,%16,
       %23)
285    yy=yy-26:IF yy=9 THEN yy=191
290    GOTO 220
500    IF KBD=27 THEN TEXT:END
510    ON KBD GOTO 500
520    RETURN
1000    DATA 0180,03c0,03c0,03c0,0180,0380,07c0,0
        fc0,1fe0,1fe0,0ff0,07b0,07d8,07cc,07c4,06
        c0,06c0,0ec0,1cc0,38c0,70c0,30c0,10e0
1020    DATA 00c0,01e0,01e0,01e0,00c0,01c0,07e0,0
        ff0,1ff8,1bcc,1bc6,19c3,1bc1,1bc0,03e0,01
        e0,00e0,01c0,03c0,0fc0,0cc0,0cc0,00e0
1040    DATA 0060,00f0,00f0,00f0,0060,00e0,01f0,0
        1f0,01f0,01f0,01f8,00fc,00f6,00f2,00f0,00
        f8,00c8,00cc,00cc,00ce,00c6,00c6,00e7
10000    FOR a=167 TO 0 STEP-26
10010      PERFORM moveto(%0,%a):PERFORM lineto(%
           279,%a)
10020      NEXT a
10030    RETURN
```

```
0     REM ****************************************
1     REM * Graphing Demo -- by Brenda Shaw      *
2     REM * -------------                        *
3     REM * This program graphs functions of     *
4     REM * your choice on the GRAPHICS screen.  *
5     REM * See the program for details.         *
6     REM ****************************************
10     ON ERR INVOKE"/BASIC/BGRAF.INV"
100    PERFORM initgrafix:OFF ERR
110    REM * Initialize Graphics Parameters      *
120    OPEN#1,".GRAFIX"
130    PERFORM Grafixmode(%2,%1)
140    PERFORM Fillcolor(%15)
150    PERFORM Pencolor(%0)
160    REM * Draw the x and y axis for the graph *
170    HOME:PERFORM Fillport
180    PERFORM Moveto(%280,%0)
190    PERFORM Lineto(%280,%192)
200    PERFORM Moveto(%0,%96)
210    PERFORM Lineto(%560,%96)
220    REM * Put the x and y coordinates on the axis *
230    FOR x=-8 TO 8
240      y=94:z=277+32*x
250      PERFORM Moveto(%z,%y)
260      PRINT#1;x
270      NEXT x
280    FOR y=-5 TO 5
290      IF y=0 THEN 310:ELSE z=284:q=99+16*y
300      PERFORM Moveto(%z,%q)
305      PRINT#1;y
310      NEXT y
320    REM * Define the current function to be graphed *
330    VPOS=10:PRINT"FN T(A) as shown below is the ";
340    PRINT"function currently in the program."
350    PRINT:PRINT"To change the function, retype ";
360    PRINT"the - ENTIRE - line with the function ";
370    PRINT"you wish":PRINT"to graph, enter 'RUN', ";
380    PRINT"and press 'RETURN'.":PRINT
390    PRINT"If it is the function you wish, then ";
400    PRINT" enter 'CONT' and press 'RETURN'."
410    LIST 420:END
420    DEF FN T(A)=COS(A)
430    REM * Graph the current function *
440    FOR x=-9 TO 9 STEP .1
450      y= FN T(x):z=280+32*x:q=96+16*y
460      IF f=0 THEN f=1:PERFORM Dotat(%z,%q):ELSE PERFORM
         Lineto(%z,%q)
470      PERFORM Grafixon
480      NEXT x
490    PERFORM Moveto(%0,%191)
500    PRINT#1;"{ HIT ANY KEY TO CONTINUE }"
510    GET a$
520    HOME:TEXT:GOTO 330
```



```
{ HIT ANY KEY TO CONTINUE }
```

Lazarus /// (Undelete your deleted files!)

How much are your important data files worth? $100, $1000? Even if you back up your files regularly, the one file you accidentally delete will be the one you haven't ever backed up.

Wouldn't it be great if you could somehow regain those files you deleted? Well, with **Lazarus** ///- you can! Very easy to use, just insert the diskette with the files you deleted and **Lazarus** /// will recover it. Completely user friendly, this program has on-line help and tutorial screens to aid in the use of the program. It even works with ProFile and other disk drives!

If you order this program today, you can get it for the pre-introductory price of $24.95. Orders postmarked after Sept. 30, 1983 will be sold at the full price of $29.95. This program will be shipped on Sept. 30, so place your order today for the best Apple /// utility in town. Please add $1.50 for shipping and handling.

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 buy the disk!

DOM #1 - Extra Disk Space Plus!

This disk contains all the programs contained in the January and February-March issues of ON THREE. Included are Disk Pak1, which will give you four extra blocks of disk space on all your data disks (a very handy feature for those budget conscious people 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) 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 now 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 an Apple][DOS diskette 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!

For only $9.95 (plus $1.50 for postage and handling) you can get any of these great packages. If you want to order all three you can get them for the extra low price of $22.50 (plus $2.00 for postage and handling). Order today!

For only $9.95 (plus $1.50 for postage and handling) you can get either of these great packages. If you want to order both you can get them for the extra low price of $15.00 (plus $2.00 for postage and handling). Order today!

Group rates are as follows:

2-9 disks: **$7.50** apiece + $2 total shipping
10-24 disks: **$7.00** apiece + $3 total shipping
over 24 disks: **$6.50** apiece + $4 total shipping

Group rates 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

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, this kit comes complete with a plug in clock chip with a battery backup.

With 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 date and time 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 to get that deal anywhere else!

What's the best part? - The price! While others are selling theirs for $60 and up, we have broken the $50 barrier. Heck, we broke the $40 barrier!

For only $39.95 (plus $2.50 for postage and handling) you can get the best little clock in town!

Group rates are as follows:

2-9 clock sets: **$36.50** apiece + $5 total shipping
10-24 clock sets: **$33.25** apiece + $7 total shipping
over 24 clock sets: **$31.00** apiece + $9 total shipping

Group rates 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

Important Notice

In order that we can go to a monthly format starting in October, we will not publish an August-September issue of ON THREE. In early October all subscribers will receive the October issue of ON THREE and we will publish on a regular monthly basis from that point on.

As a reiteration of our policy, the subscription fee is $30 for twelve issues, not a year. The double issues (this being the last one) count as one towards the total of twelve that subscribers will receive.

Once again, I have to thank all of you who have sent in article submissions. This is the reason we are going to a monthly format and I'm sure each and everyone out there will appreciate a more regular magazine.