

TM

ON THREE

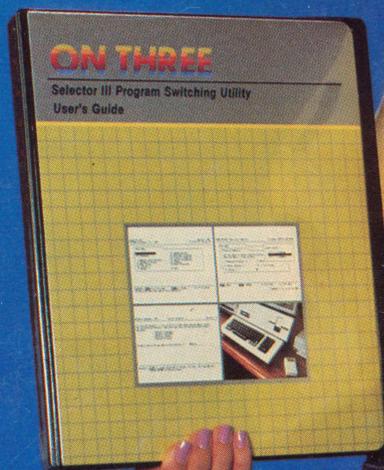
The Magazine For Apple III Owners and Users

Volume 3 - Number 1

January 1986
\$4.00 U.S.

REVIEW: Word Juggler

About Basic



PLUS:

- Graphics
- Sorting
- Emulation
- ...and more



Apple III 512K Memory Upgrade



= Increased Productivity Through Your More Powerful 512K Apple III!

ON THREE Presents. . .

For everyone who has wished for more memory in their **Apple III** or **Apple III plus** - here it is!

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

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

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

Have you ever had problems putting some programs on your ProFile hard disk while running under Catalyst? Since Catalyst takes up a good portion of memory, there are times when there isn't enough memory to go around. Many programs will not run at all under Catalyst because of memory limitations. Some programs will run only on minimal Catalyst systems. That means no spooling, no special purpose utilities like ONTIME

or the Calendar-Pak, and a lot of dynamic driver loading. Who needs problems like this? With the **512K Memory Upgrade**, your problems with programs not fitting while running under Catalyst will be solved. You will be able to put Lazarus and Draw ON under Catalyst and have your Draw ON pictures being printed out while you undelete a file with Lazarus or even work on your word processing with Applewriter or Word Juggler. The BPI system and all other large programs such as Keystroke will now fit easily under Catalyst.

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

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

The ON THREE 512K Memory Upgrade

is specially priced at only **\$399***†

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

- 1) An **Apple III 512K Memory Board** using state of the art 256K memory chips.
- 2) The **Confidence Memory Program** which will ensure that your **512K Memory Board** is working correctly.
- 3) The RAM ultra-fast **RamDisk** drive with demonstration programs.
- 4) The **Upgrade to 512K Utility** that will update all of your disks so that they will work with the memory upgrade, and the **updated** version (1.2) of the **System Utilities** program that permits larger SOS DRIVER files.
- 5) **Complete** instructions for installation and use.
- 6) A full **90-day Warranty**

*The **ON THREE 512K Memory Upgrade** can only be used on a 256K **Apple III** or an **Apple III Plus**. If you have an old 128K **Apple III**, the upgrade costs **\$449*** + shipping (there is no rebate), and it must be installed by a dealer or by **ON THREE**. We offer same day turnaround on 256K or 512K memory upgrades and charge only \$50 plus shipping for the installation.

†The purchase price of the **512K Memory Upgrade** is \$449* plus shipping. After installing the new **512K Memory Board**, return your old 256K memory board to us and you will get a rebate of \$50.

‡Price effective through 1/31/86 only.

Ordering Information

Item:	Price:	Shipping:
512K Memory Upgrade	\$ 449	+ \$10.00
A143 560K Floppy Disk	\$ 399	+ \$ 6.50
A3 143K Disk Drive	\$ 299	+ \$ 6.50
Trustor 10H Hard Disk (Main Drive)	\$ 999	+ \$35.00
Trustor 10H Hard Disk (Second Drive)	\$ 899	+ \$35.00
Trustor 10H Combo (Main and Second)	\$1848	+ \$70.00
Trustor 30 Hard Disk	\$1999	+ \$35.00
Draw ON III Graphics Tool	\$ 179	+ \$ 5.00
(Specify which printer and interface card you are using.)		
Apple III Fruit Machine	\$ 19.95	+ \$ 2.00
Apple III Card Machine	\$ 24.95	+ \$ 2.00
Game Combo (Fruit & Card Machine)	\$ 39.95	+ \$ 3.00
Lazarus III File Restoration	\$ 49.95	+ \$ 2.00
ON THREE O'Clock	\$ 49.95	+ \$ 3.00
ONTIME Utility Package	\$ 39.95	+ \$ 2.00
ON THREE O'Clock - ONTIME Combo	\$ 79.90	+ \$ 4.00

ON THREE

4478 Market Street, Suite 701-702
Ventura, California 93003

To place an order over the phone, call **(805) 644-3514** direct.
California residents add 6% sales tax for products (not shipping).

We accept Visa, Mastercard and American Express.†

†Per our usual policy we will give schools, government agencies and Fortune 500 companies NET-15 terms. We do not extend credit to individuals or companies.

‡3% surcharge on AE.

ON THREE T.M.

The Magazine For Apple III Owners and Users

Publisher:

Bob Consorti

Editor:

Val J. Golding

Comptroller:

Joseph Consorti

Logistics:

Janet Schanz

Interior Art:

Virginia Carol

Technical Support:

Rob Turner

Order Dept.:

Claudia White

Typography:

West Coast Typography

Van Nuys, CA

Printing:

California Offset Printers, Inc.

Glendale, CA

ON THREE™—The Reference Source for the Apple /// is published monthly by ON THREE, 4478 Market Street, Suite 701, Ventura, CA 93003. (P.O. Box 3825 Ventura, CA 93006.)

For a copy of author guidelines, send a stamped, self-addressed envelope to the above address. Current page rate: \$35 per printed page.

Return postage must accompany all manuscripts, diskettes and drawings submitted if they are to be returned. All manuscripts longer than 500 words or program listings longer than one half page must be accompanied by a diskette if they are to be considered. No responsibility can be assumed for unsolicited materials.

All letters received by ON THREE unless specifically marked to the contrary, will be considered as unconditionally assigned for publication and are subject to ON THREE's right to edit and comment editorially.

Subscription information:

U.S.: \$40 for 12 issues.

For First Class service, remit an extra \$12

Foreign subscription information:

Canada, Mexico, APO, FPO \$20 additional, \$60 total.

So. America, Europe: \$24 additional, \$64 total.

Pac. Islands, Asia, Australia, Mid-east: \$29 additional, \$69 total.

Back Issues:

Available at \$5 each. Please indicate issues desired on the enclosed order form.

Postage for ON THREE products:

U.S.: listed on enclosed product price sheet.

Canada and Mexico: double listed US prices.

Outside North America: triple listed US prices.

All funds must be remitted in U.S. Dollars drawn on a U.S. bank or funds in your native currency at the current exchange rate drawn on a bank in your country.

Group purchases must have one common mailing address. Please contact ON THREE at the above address for information on current volume pricing and terms.

ON THREE is a registered trademark of ON THREE. Apple, Apple II, Apple II plus, Apple III, Apple III plus, Applesoft, Apple Business Basic, Disk II, Disk III, Lisa, LisaDraw, Macintosh, MacPaint, MacWrite and ProFile are registered trademarks of Apple Computer, Inc. Micro-Sci and Gameport III are registered trademarks of Standum Controls, Inc. Selector III is a registered trademark of Sabre Software, Inc. ON THREE O'Clock, Lazarus III, Draw On and ONTIME are registered trademarks of ON THREE, the Apple III magazine.

Opinions expressed in this magazine are those of individuals authors and staff and not necessarily those of ON THREE.

Entire contents copyright ©1985 by ON THREE. All rights reserved.

Volume 3, No. 1

Table of Contents

January, 1986

FEATURES

Review ON: Word Juggler Word Processing at its Finest 7	Titan /// Plus // 25
Arthur A. Schumer from the wilds of north dakota, a thorough and positive look at quark's apple /// word processor	Don Russell now you can emulate the apple II language card and soon the //e
Painless Programming: Now About This Basic Business... 13	Technique: Sort of a Heap 27
Val J. Golding why boot /// e-z pieces to add a few figures ... programming in simple terms	Brenda Shaw sometimes the tried and true bubble sort just doesn't hack it... here's another way
ImageHelper: The Easy Image Maker 17	Tricky Quickie: Flasher 32
Rick Gauger shortcuts to fancy graphics programming from business basic	Sharon Webb bet you thought the /// couldn't flash characters

DEPARTMENTS

The Editor Dishes it Out: Apple.Sauce 3	De-Classified 29
Val J. Golding touring the unexplored pages of on three	anyone can advertise here
Notes from the Publisher: Block Write 4	Call Three: Hot Line 30
Bob Consorti words of introduction and a host of new products	consultants help with your problems
Reading the Mail: Three Questions 21	Apple /// User Groups 30
where the reader can get a word in edgewise	coming soon to an area near you...

ON: The Cover

This Month the ON THREE cover features lovely Carolyn Consorti modeling ON THREE's new "I Love My ///" t-shirt (only \$1 1.96!) For a back view, see the back cover. (Where else?)

Photography by David Ojerholm, H & O Studios, 614 E. Main St., Ventura, CA 93001

I Love My ???

Some time back, Apple had a campaign called "Apple III forever." Our viewpoint, of course, is *Apple /// forever* to create and continue interest in one of the world's finest computers. To this end, we have designed *I Love My Apple ///* t-shirts and sweat shirts. You can help us promote the *///* by ordering now. The t-shirts are only \$11.95 and the sweat shirts are \$18.95 each plus \$3 shipping and handling. Both come in sizes Sm, Med, Lrg and X-lrg. The t-shirts are available in yellow, beige, white and blue, while the color choices in sweat shirts are silver, white, blue and yellow. Order one of each now and demonstrate your support for the *Apple ///*.

LAZARUS /// FILE RESTORATION

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

Introducing Lazarus III...

LAZARUS /// U2: 8 22 Jan 84 5:39:06 PM
© 1982, 1983 ON THREE Prefix is: 01

```

Main Menu
Welcome to LAZARUS /// - The easy to use file restoration
utility from the folks who bring you ON THREE Magazine.
- - - - -
[ ] Set The Prefix          [ ] Undelete Files
[ ] ON THREE Information   [ ] List Files
[ ] Quit The Program
    
```

Press: [LEFT] for prior menu, [RIGHT] to select, * * * * * to move.
You may also type the name of your selection. [CTRL]-, - or = for HELP.

If you have ever accidentally erased a file from one of your disks and have wished for a little magic to bring it back. **Why Don't You Have A Copy Of Lazarus?** This extremely easy to use utility will restore any file that have accidentally deleted from any of your disks. It will work with *ProFile*, *The Micro-Sci A3, A143*, our *Trustor* hard disks and every other disk drive that you can connect to your *Apple III*. If you can save a file on it - or delete a file from it, *Lazarus* can restore your little accidents.

To restore (undelete) one or more files, you simply tell *Lazarus* where to look and press **RETURN**. *Lazarus* will look for deleted files on that disk and show you a window filled with the files that you can undelete. Just select the one you want, press **RETURN** and *Lazarus* does the rest!

LAZARUS /// U2: 8 22 Jan 84 8:02:37 PM
© 1982, 1983 ON THREE

```

Main Menu
Undelete Files
Use this option to undelete files that you have del
enter the directory that holds the file you want to
then press [RETURN]. A window will open on the right
screen displaying all of the deleted files on that
select the file you want to restore and press [RETURN]
Enter the directory which holds the file you want t
[ ] PROFILE
    
```

Press: [LEFT] for prior menu, [RIGHT] to select, * * * * * to move.
You may also type the name of your selection. [CTRL]-, - or = for HELP.

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

Lazarus III requires an *Apple III* or *Apple III Plus* with a minimum of 256K and is available for only \$49.95 + \$2 for shipping and handling. *Lazarus* is not copy-protected and may be installed under *Selector III* and *Catalyst*.

Other Products From ON THREE...

This flyer is much too small to show you all of our software and hardware products for the *Apple III* and *Apple III Plus*. Please call or write for complete product information and pricing. Some of the specials we are currently offering are as follows.

Apple III Software and Hardware from ON THREE	Price	Shipping
Used 256 K <i>Apple III</i> 's with Monitor III's	\$1050.	+ \$50.
<i>Apple III</i> Business Basic (version 1.23) with Manuals and Disk	\$99.00	+ \$5.00
<i>Apple III</i> Fortran 77 With Reference Manual and Compiler Disk	\$99.00	+ \$5.00
256K Memory Upgrade (for a 128K <i>Apple III</i>)	\$250	+ \$10.00
256K or 512K Memory Board Installation (by appointment only)	\$50.00	+ Call
Micro-Sci A3 140K Disk Drive (Disk III replacement)	\$299.00	+ \$6.50
Game Card III (allows you to connect <i>Apple II</i> game paddles or joysticks to your <i>Apple III</i> and use <i>Apple III</i> games)	\$59.95	+ \$2.00
3M Dbl. Sided, Dbl. Density Disks (10) for use in the A143	\$39.95	+ \$3.00
Disk of the Month #'s 1-6 (each - call for quantity discounts)	\$14.95	+ \$2.00
CROSSWORD—SCRAMBLER Crossword Game	\$19.95	+ \$2.00
Dust Cover for <i>Apple III</i> with Monitor III	\$11.95	+ \$2.00
Dust Cover for <i>Apple III</i> with Monitor III and <i>ProFile</i>	\$12.95	+ \$2.00
I LOVE MY APPLE III T-Shirts - Sm, Med, Lrg, X-Lrg (Color choices: Yellow, beige, white, blue)	\$11.50	+ \$2.00
I LOVE MY APPLE III Sweat-Shirts - Sm, Med, Lrg, X-Lrg (Color choices: Yellow, silver, white, blue)	\$18.95	+ \$2.00
I LOVE MY APPLE III CAPS	\$5.98	+ \$2.00

ON THREE O'CLOCK

Calling all you time conscious *Apple III* owners out there! How would you like a working **clock/calendar** for your *Apple III*? Did you know that your *Apple III* has provisions for a built-in clock/calendar? With a clock, your *Apple III* can function more efficiently and make your life easier. Just as it was originally intended (and built-into the *Apple III Plus*, the **ON THREE O'Clock** kit comes complete with a plug-in clock chip with battery backup and easy to follow instructions.

With an **ON THREE O'Clock** installed, whenever 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 will automatically use the *Apple III* built-in time and date routines to give you an up to the second read-out of what time it is.

Extremely easy to install and adjust, it is completely compatible with **SOS** and doesn't use up a slot! This is the one you have been waiting for! The package contains comprehensive instructions and a **Six Month Warranty**. Try to get that deal anywhere else! For only \$49.95 (plus \$3.00 for postage and handling) you can get the best little clock in town. We offer quantity discounts, so please call before placing a volume order.

ON TIME

The **ONTIME** package from *ON THREE* is an incredible new enhancement to your *Apple III* system (with an **ON THREE O'Clock**) or your *Apple III Plus*. While you don't need **ONTIME** to use your *Apple III* clock, as soon as you have it you won't know how you were able to live without it. Before **ONTIME** the only way you could find out the time from within most programs is if you looked on your watch!

ONTIME, the incredible new product from *ON THREE*, will conquer this problem by giving your *Apple III* many great new features: With the **ONTIME** driver installed, you can display the Time & Date whenever **YOU** want it. A simple keystroke combination will turn the clock display on, and another will turn it off. One more turns on a stopwatch that can count down by tenths of a second! This is very useful in monitoring how long you've been on the telephone, or using a computer service like the Source.

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

Coming soon are add-on utilities to **ONTIME** that will give you a **Calendar, Calculator, Notepad, Communication Package, a Spelling Checker for III E-Z Pieces** and more. All of these add-on utilities will also work in the background. You will be able to press a button to jot down some notes to yourself while you are working with some other program like *Visicalc*. You will also be able to use the **Calendar** to set your week's schedule of meetings. When the time comes for that meeting, the computer will prompt you, from within whatever program you're working in. With the communications utility you will be able to send files using a modem from within the program you are using!

All of these utilities will be shortly available for the **ONTIME** package. **ONTIME** is currently priced at only \$39.95 (plus \$2.00 for postage and handling). **ONTIME** requires an **ON THREE O'Clock** or equivalent to work. If your *Apple III* doesn't already have a clock, you can purchase both the **ON THREE O'Clock** and **ONTIME** combination for only \$79.90 (plus \$4.00 for postage and handling). That's ten dollars off! If you have been waiting for a reason to get a clock - **ONTIME** is it!

Please call or write for availability of the **ONTIME Version 2** which has a built-in **Calendar, Calculator, Notepad, Communications Package, a Spelling Checker for III E-Z Pieces** and more. Current users of **ONTIME** will be able to directly upgrade to Version 2 at low cost.

ON THREE Magazine

ON THREE is far more than the largest producer and distributor of *Apple III* software and hardware in the US, it is also the *Apple III* support group and publisher of *ON THREE* Magazine. If you are an *ON THREE* subscriber, it's pretty hard to ignore the fact that we have not published for the last year. This has been a matter of business priorities, and **Bob Consorti** found out the hard way that his energy could be stretched only so far. A solution has been found in hiring **Val Golding** (former editor and founder of *Call-A-P.P.L.E.*) to take over the chores of producing a monthly *ON THREE* magazine. As a consequence, those of you who subscribe will be receiving a new issue in less than 30 days, packed with new information, techniques and reviews that as an *Apple III* user you will not want to miss. And if you are not already a subscriber, here is your chance, a 12-month subscription for \$36.50 (regular rate \$40). You must mention this mailer when remitting. We also need your input in the form of letters or articles (we pay \$35 per page), so share your *Apple III* experiences with us.

The Editor Dishes it Out:

Apple.Sauce

val j. golding

Onward Three

Welcome back to the world of *ON THREE*. We'd like to thank those of you have stuck around and with *ON THREE* through its various stages of publication and non-publication for their patience. Alas, as we found out some years ago, there isn't always time to do what you want, and when you are operating a business, the business usually must come first. **Bob Consorti** will vouch for that. Our role then is to assist Bob and guide the magazine.

Our introduction to the /// was via A.P.P.L.E. co-founder **Bob Huelsdonk** who took delivery of one of the very first ///'s delivered to Seattle. Bob was, and still is, very proud of his /// and he always delighted in showing it off to us and to anyone within earshot. It is truly a powerful machine, every bit as innovative as was Woz's original Apple One back in 1976. We have followed the (mis-) adventures of the /// ever since, cried a bit when we heard that the company that gave birth to such a fine machine also attempted to snuff out its life. But you owners out there—our friends—knew better. That's why *ON THREE* is still here.

Currently there are some 100,000 plus machines out there. How many of you remember that, although the design guidelines provided for far more, that the first Apple /// was only 96K. Now, through *ON THREE*, you can upgrade to 512K. Not too shabby for a five-year old machine, huh? As we followed the progress of the ///, we were surprised to learn that there are still a few new products being produced for the ///, and occasional upgrades as well.

One new product we've just recently been made aware of is a real hush-hush earth-shaker. We'd like to tell you more, but we also like our job. Nevertheless, you'll read about it here first, within the pages of *ON THREE* in our *Apple Sauce* column. Each month we'll fill you in on new products when we find 'em and re-acquaint you with some of the existing ones. In addition, we'll take the time to point out some of the stories and features in the current issue and explain how you can benefit from them. We'd like also to make our column a forum for story ideas: the types of articles and programs you most want to see in *ON THREE*. But be careful what you say, we may call on you to supply the material!

And while we're on *that* subject, we need to put in our pitch for material we can use in *ON THREE*. The descriptions given in the following section should indicate to you the wide variety of material we plan to publish, and we solicit your contributions. Short articles or programs (running less than two printed pages may be forwarded to us "on speculation"; you should inquire first regarding feature material. Material selected for publication will be paid at the rate of \$35 per printed page, exclusive of advertising space or other matter. Author guidelines are available upon receipt of a stamped, self-addressed envelope. They may be mailed to:

ON THREE
P.O. Box 3825
Ventura, CA 93006

We are also interested in contributions of cartoons, news items, rumors, etc. Remember, your efforts are going to benefit all /// users.

Sophisticated Events

Art Schumer, who has written for us in the past, leads off this issue with *Word Processing at its Finest*, an in depth review of Quark's *Word Juggler* word processor, certainly among the fastest written for the Apple ///. Its main drawback appears to be that it can not presently use the extra RAM of a 512K Apple ///, a situation we hope to rectify in the near future. Looking ahead to next month, Art will be back with comprehensive coverage of Word Juggler's supporting products: the *Lexicheck* spelling checker, *typeface*, a formatter for output to typesetting machines, *Discourse*, a software printer spooler and *Mail List Interface* which, as its name proclaims, interfaces to Apple's Mail List Manager.

ImageHelper: the Easy Image Maker describes two Business Basic programs by **Rick Gauger** of Bellingham, WA (near our old Seattle stamping grounds) which take the work out of creating and processing images for Business Basic's invokable module BGRAF, which includes *Drawimage* (in the Apple /// Graphics package.) If graphics and precise animation are your bag, then by all means don't miss this one.

If you can watch your favorite TV program while your Apple /// is running a sort for you, then you're in a heap of trouble and **Brenda Shaw's** *Sort of a Heap* may just solve a problem for you.

Many Apple /// owners have had their machines for several years and have used them only to run "canned" or commercial programs, which is all and well, and certainly one of the purposes for which this fine machine was designed. But you may also be missing a bet if you haven't yet learned programming. Chances are, if you've let the kids get their paws on the machine, that in addition to playing games, they are already writing programs. Our series *Now About That Basic Business...* is written especially for you, to introduce you to programming in easy to understand plain English. Take a gander, and you may be surprised to see how easy it is and how you can increase the power and utility of your Apple ///.

ON THREE is far more than just the magazine you are reading at this moment, it is a producer and distributor of Apple /// hardware and software. Moreover, it is an Apple /// support organization. It is, and should be, one of your major resources for Apple /// information and problem solving. Our readers and customers can help us in that respect through our *Three Questions* letters column. Not only can we answer inquiries regarding *ON THREE* products and services as well as other technical matters, but you, yourself can be of assistance. We encourage you strongly to share problems and solutions you may have run into with the idea of paving the way for other readers to tread. We are unique, in that the computer we revere, although highly viable, is no longer in production. There therefore exists a strong kinship among us. So pitch in. If you have questions, write. If you have answers, write.

We'd also like to direct your attention to listings, elsewhere in this issue, of the *Call Three: Hot Line* and the Apple /// user groups. These are two more examples of the type of support you can reach through *ON THREE*.

Onward Three! 

Block Write

bob consorti

Over the past few years I have had the pleasure of working on the Apple ///, producing products and services needed by Apple /// owners. I have however, out of necessity, become a jack-of-all-trades, conceptualizing, writing and documenting software, producing the magazine and a myriad of other business related tasks.

When Apple cancelled the Apple /// a year and a half ago, like other Apple /// related companies, I had to make a decision. To stick with the Apple /// or go on to another computer. We all know what I chose. The /// is still one of the most powerful computers around, and all of us at *ON THREE* intend to keep producing new products and services for it.

If this is the case, you are probably wondering why hasn't *ON THREE* published for the past year? Good question. I won't get into specifics, but the general day-to-day operations of developing a few new products (described below) has consumed all of my time, thus, no magazine. In response to that, I am now proud to say that we have hired two additional individuals.

The first, **Rob Turner**, has been programming part-time on the Apple /// for several years while holding down a full-time job and going to school. We were able to lure him away and he is now hard at work assuming all day-to-day programming responsibilities. He is able to realize all of the products that I have wanted to see on the /// for a long, long time. Some of the things he is working on will be described below. Suffice to say that with the software we have planned, it will be a long, long time before the /// becomes obsolete.

Our other recent find is **Val Golding**. Some of you may remember Val from the number of years he served as editor of *Call—A.P.P.L.E.* magazine. Although that magazine was primarily an Apple II publication, we have persuaded him to work for us as managing editor of *ON THREE* Magazine. While it will take a bit for him to get 'up and running' on the ///, he brings with him a great deal of professionalism and integrity to *ON THREE* Magazine. As a result, I can now firmly

announce that henceforward, *ON THREE* will publish on a regular monthly schedule.

As Val learns about the ///, I will continue to answer questions in the *Three Questions* column. He's quick to catch on, so give him some time and he will become an ardent /// fan. *ed note: did you say "will"?*

What's New

Over the past year of magazine dormancy, we have pursued a number of hardware and software projects for the Apple ///. It is my pleasure to describe some of them now. The *512K Memory Upgrade*, the *Xebec 10H and 30 Hard Disks*, the *Apple /// Unidisk 3.5*, *Selector ///*, *Draw ON ///* printer upgrades, *ON THREE Games*, and the *ONTIME Desktop Manager*. These products have been designed to keep you up to date with the latest products available for other business computers. So, if you think your /// is headed for obsolescence, look what we have in store for you.

512K Memory Upgrade

As many of you know, the *512K Memory Upgrade* gives your Apple /// a full 512K of memory—the most that the machine can use. Many programs will work directly with it: *VisiCalc*, *Advanced VisiCalc*, */// E-Z Pieces*, *AppleWriter ///*, *Business Basic*, and *Pascal* will have useable space of well over 400K. If you ever have had problems running large programs such as *BPI*, *Draw ON ///*, *Lazarus ///* and others on your 256K machine while using *Catalyst*, the memory upgrade will cure them! Due to memory chip price drops we can now aggressively price the *Apple /// 512K Memory Upgrade* at just \$399. The purchase price is actually \$449, but after a \$50 rebate which you will receive when you return your old 256K board to us, your final cost is only \$399.

We can now also directly upgrade 128K Apple ///'s to 256K or 512K. We prefer to do the upgrades ourselves, but if you have a dealer that you trust, you can have him do the upgrade. The price of the 128K to 256K upgrade is \$200, and installation is \$50, plus

return shipping. The price of the 128K to 512K upgrade is \$449. Likewise, if we install it, the cost is \$50, plus return shipping. If you want us to upgrade a 128K to 256K or all the way to 512K, call (805) 644-3514 to schedule an appointment. We offer same day turn-around since UPS delivery arrives in the morning and our shipment leaves in the afternoon. A full ninety day limited warranty applies to both the 256K and 512K memory upgrades, so if you are running out of memory, call *ON THREE*.

Xebec 10H and 30 Hard Disks

Over the past few years *ON THREE* has looked for a low cost, high reliability hard disk unit to offer our customers. After much searching we found the *Trustor* series from *Xebec*. We have developed an Apple /// interface for the *Xebec 10H and 30* hard disk drives that offers very fast data transfer, making these drives *two to five* times faster than the *Apple ProFile* hard disk. As shipped, these drives come complete with everything you need to get them running on your Apple ///—The drive, interface card, cabling, documentation and driver disk.

The *10H* is an attractively styled hard disk that has a unique option. If you ever outgrow the *10H's* 20808 blocks (20.16 megabytes), you can attach a second one (daisy chain) to the back of the first. Priced at only \$999, the *10H* drive works perfectly with all Apple /// software. Combined with our *Selector /// Program Switching Utility* (described below), you can put all of your programs on the hard disk and switch between them at the touch of a button.

The *30* is an extremely fast hard disk for the ///. It has a capacity of 69,632 blocks (34 megabytes). Although the Apple /// is supposed to be able to work with disk drives up to 32 megabytes, it can actually only address up to 16 megabytes. Because of this, we have partitioned the drive as three devices. The first two are 16 megabyte disks, while the third functions as a two megabyte disk. The *30* also works with all Apple /// software and is compatible with the *Selector /// Program Switching Utility*. The *30* is priced at \$1999.

If you compare the list prices of some other hard disks available for the Apple ///—Apple 10 megabyte *ProFile* = \$1999, *Quark QC10* = \$1999, you will see that for the money, our 10 megabyte drive costs half as much as similar drives on the market. For \$1999 you can get our *Trustor 30*, which holds 3.4 times the amount of information of the other drives—for the same price.

Unidisk 3.5

One of the more exciting announcements that Apple made recently is the *Unidisk 3.5* for the Apple II computers. This is the same kind of 3½" drive used by the Macintosh. While the Macintosh is a single-sided 400K drive, the Apple Unidisk is double-sided and handles 800K of data. You may be wondering why we are so excited about Apple II drives. There are several reasons:

The first is their relatively large capacity. They hold almost six times the information of a standard 140K disk drive. Compared to our A143 560K drive, the Unidisk offers a 43% increase in capacity and therefore makes an even better hard disk backup device than the A143 drives.

Secondly is the *compatibility* problem. Most Apple II computers sold from now on will be offered with Unidisks instead of standard 140K drives. If you have Apple II's in your office and a /// at home, or vice-versa, till now you have probably been able to just put your work on a disk and take it home with you at night. As the 140K drives are pushed out of the marketplace, you'll instead be soon taking home a 3½" Unidisk diskette.

Thirdly is the issue of *convenience*. Those tiny disks are great! They fit in pockets, purses, briefcases much easier than the old 5¼" disks. Since they have a hard plastic shell they can take much more punishment than the regular disks. All in all, they are more reliable, faster, easier to use, and just plain better than the standard 5¼" disk.

In addition, Unidisk will also read from and write to a standard Macintosh diskette. While no file conversion utility exists

presently, we have planned a program for your /// that will convert files to and from Macintosh format. Thus, if you have Mac's at work, you can bring the Macintosh disk home, run the utility program to convert the file you want to use to Apple /// format, and then work on that file on your Apple /// at home. When finished, you change the file back to Macintosh format and use the same disk at work.

As you may have guessed, we have the Unidisk 3.5 working on the Apple ///. They connect to the Apple /// via the //e interface card, and you can daisy chain (attach one on the back of another) up to four Unidisk drives onto the interface card. We offer the *Unidisk 3.5* (800K - 1600 blocks) with Apple //e interface card and the Apple /// software driver and documentation for only \$499. The list price of the drive and interface card for the //e is \$565, so as you can see, we are offering an exceptional buy on this fabulous new Apple /// drive.

We are also offering the Apple /// software driver and documentation separately. It may be that you can get a better buy on the drive and interface card from a mail-order house. In that event, you may obtain just the software driver and documentation from us for \$50.

Selector ///

After much waiting and a lengthy development, the *Selector /// Program Switching Utility* is here! For those of you who are unaware of program switching, let me fill you in. Say you have a few programs. Maybe they are *VisiCalc*, /// *E-Z Pieces*, *Quickfile* and *Keystroke*. To use one program you must boot the computer using that program's diskette. If you want to switch programs, you must re-boot with another diskette. For larger programs this may take a minute or so, not counting the hassle of finding the right disk.

With *Selector ///*, life will become a little easier. If you have a hard disk (or the 800K Unidisk), you can place all of your programs on it. In the morning you will turn on your computer with the *Selector* start-up disk and see a menu listing all of the programs on the hard (or other) disk. You can use any program on the hard disk, and when you leave the program, the *Selector* menu will reappear. Selecting another program is as easy as pressing RETURN. There is an ad elsewhere in the magazine fully describing *Selector ///* and all of the programs that it works with.

The *Selector /// Program Switching Utility* has pop-up menus that make the /// more attractive and easier to approach for first time users. Combined with a hard disk (or the Unidisk), *Selector ///* can improve your productivity. It is priced at only \$99.

Draw ON /// Printer Upgrades

When *Draw ON ///* was first released, it worked only with printers connected to the *PKASO* printer interface. We are now able to offer many different versions of *Draw ON* that do not require the *PKASO* card. We support the *Epson* line of printers, and the *Apple DMP* and *Imagewriter* printers. Also, since the *ProWriter* and the *NEC* printers are similar to the *DMP*, you can order the standard *DMP-Imagewriter* version of the program and it will work with your printer. We also have a version of *Draw ON* that fully supports the advanced printing features of the *PKASO/U* interface.

If you have an early version of the program, we will upgrade it for only \$15, including postage and handling. To upgrade, send in your old *Draw ON* Start-up and Program disk (not the Sample Files disk) along with \$15 and a note indicating which version you would like. You'd be amazed to know how many people send their disks in and ask for an upgrade, but fail to say what version they would like. So when ordering, please let us know the exact printer and interface card you are using.

ON THREE Games:

Just for fun, we have developed a number of game programs for the Apple ///. Two were developed by **Dr. Astrahan**, author of *Draw ON ///*. We offer *Slot Machine*, a full color slot machine game for the ///, and *Card Machine* (Blackjack). They both work in full, vivid color and demonstrate the power and speed of Apple /// graphics. The other games will stimulate your mind. First, comes the first and true Adventure game. Next, is a three-in-one game disk that contains *Star Trek*, *Othello* and *Wumpus* (a multi-level adventure game). Ads for these games appear elsewhere, so if you think the /// is just for business, pick up one of the *ON THREE* Game Disks.

ONTIME Desktop Manager

Perhaps the most exciting thing we have been working on is the *ONTIME Desktop Manager*. Have you ever been working in your word processor and needed to multiply two numbers? Gotten a bit upset because you have a few thousand dollars of computer equipment at your fingertips and can't multiply two numbers? Or, how about this—you're working in your spreadsheet and you need to jot down a note to yourself. While hunting for a pen and paper, you probably curse the computer.

Or, maybe you're working in a program and you need to save a file, but aren't sure if the file name you want to save to already exists. Crossing your fingers, you type in the file name and hope you don't overwrite a file of the same name. Or maybe you're working in a program and need to save a file, but find you don't have a disk with enough room on it. You have a box of blank disks but haven't had time to format one. If you exit the program to use the System Utilities to format a disk, all of your work will be lost.

Have any of these things ever happened to you? How would you like to clear your desk of that old-fashioned calculator, the pens and paper, your appointment calendar and increase your productivity? The *ONTIME Desktop Manager* will do these things and much, much more. From within any program, a keystroke will suspend the program you are working in and display a window into the Desktop Manager. You can stop anything you are doing, go into the Desktop Manager and select any of the following:

- An appointment calendar. Enter times of appointments and things to do at a specific time like "Call Johnson at the L.A. office at 10:30 AM", or "At 5:00 PM leave work and get milk at the supermarket". At 10:30 AM that day a window will appear on your screen and display the first message. At 5:00 PM the second message will appear.
- A full feature calculator (SIN, COS, TAN, EXP, LOG, LN, Memory and more!) You will also be able to do base conversions. Change from decimal to hex to octal to binary and back again.
- An easy to use, always there notepad with full editing features. You will be able to jot down page after page of notes to yourself. You can even edit other files from within the notepad.

The above items are the three main parts of the *ONTIME Desktop Manager*. But that's not all! If you have the *Draw ON* /// program, chances are you have an Apple //e mouse attached to your system. You might wonder if you could use the mouse from within other programs, such as VisiCalc or System Utilities. With the *ONTIME Desktop Manager* you can! If you have a mouse, simply move it, and the cursor on your screen will move around—left, right, up and down. Clicking the mouse also acts as an

ESCAPE or a RETURN key. You can even set up the Desktop Manager so that when you press the button the Desktop Manager window will appear on the screen.

The last feature of the *ONTIME Desktop Manager* is the ability to transfer text from one screen to another. Say you are using the Calculator to do some calculations and want to transfer the result to your word processor or your data base. You can simply cut from the calculator and paste it into your program. Likewise, you can move an entire section of text from your program to the notepad or vice-versa.

We have many other functions planned for the *ONTIME Desktop Manager*. Many of them have been completed while others are still underway. Everything described above will be included in the final version of the product. The basic *ONTIME Desktop Manager* will cost \$99 and should be available mid-December. However, we are not accepting advance orders, so you should call us at (805) 644-3514 to check on availability.

Other products already available include a full-feature communications package which also offers Tektronics graphics terminal emulation, while one which many of you will want is a collection of most-needed disk utilities which, because they work through the *ONTIME Desktop Manager*, will always and instantly be available to you. List, lock, unlock, delete, rename and copy files, format floppy disks, and set the time and date. Most of the functions normally provided by the System Utilities program are available via the *Disk Utilities* package for the *ONTIME Desktop Manager*.

Two additional forthcoming products are a spelling checker and a graphics charting program for /// E-Z Pieces. These Desktop Manager utilities are sometimes called background utilities. This is because they lie in the background, waiting to start working for you. When you're through they disappear at the touch of a keypress.

One of the best future additions will be the *MACRO Utility*. A Macro is a series of keypresses that can be recorded and played back as you wish. For example, you may need to do a number of find and replace operations in your word processor. You may need to change company names, addresses and phone numbers in a boilerplate contract you are working on. You may have already done it a hundred times, but you have to type it in each time. With the Macro utility, you would record the keystrokes just once, and place them in your file as needed.

The *ONTIME Desktop Manager* requires an *ON THREE O'Clock* or equivalent (an Apple clock is just fine) and a large capacity disk drive. An Apple //e mouse is optional but makes it possible to use a mouse from within any other Apple /// Program.

The disk drive can either be a Micro-Sci A143, an Apple Unidisk 3.5, or any hard disk system. If you try using the Desktop Manager from floppies it will, of course, be much slower than a hard disk. You must also have a minimum of 256K (512K is recommended) of memory as the *ONTIME Desktop Manager* takes up approximately 40K.

Copy-Protection Notes

Because we don't want to copy-protect our software, we need your help. We put a lot of effort (time and money) into creating all of our software products. We received a call this afternoon from a gentleman asking if our *Draw ON* Graphics package was copy protected. We inquired why and he replied that he and two other Apple /// owners bought a copy of Visicalc intending to copy it, but found it was copy-protected and consequently could not be copied. He then asked if we allowed our software to be copied.

We are very concerned that people will make copies of our software and documentation and give them away at user group meetings, within companies with a number of ///'s, or between friends. Since we don't copy protect any of our software, all of this is possible. We do, however, offer price breaks on multiple purchases of the software and documentation, in addition to multi-use licenses. The above mentioned cases are typical of what happens when our (or anyone else's) software hits the market.

Everyone should be aware of this fact: If illegal copies of our software start circulating, we will quickly lose both money and interest in creating new Apple /// software. Thus, because we want to continue providing you with quality software and hardware products, we ask you not to copy our software and give it away. You are, of course allowed to make back-up copies for your own use. We work very hard to produce low cost, high quality products and have every right to expect reasonable compensation for our efforts.

Future Plans

Now that we are on a regular monthly publishing schedule, *ON THREE* can again share with you the things we have learned

[Cont'd on page 29]

Word Processing at its Finest

arthur a. schumer

Since its introduction, the Apple /// has had trouble gaining popularity because of early reliability problems and lack of software. That situation was only compounded by Apple's decision not too long ago to drop the machine from production, an unfortunate and unwise decision, particularly so since just prior to that time, things were finally beginning to look up with the dedication by Apple of a new department specifically to support the Apple /// users. At the same time they had just released the long awaited SOS Technical Reference manuals. Outside software vendors realized the power of the /// and published more Apple /// software in the period just before the machine was discontinued than ever before. Even so, there are currently more than 300 software products published specifically for the /// and new products, both hardware and software, slowly continue to appear.

... all of their software enhances Word Juggler ...

Quark Incorporated was one of the first companies to devote their resources to the Apple /// and their software products do an outstanding job of utilizing the variety of features that the Apple /// has to offer. Even more important is the fact that all of their software is centered around and enhances their *Word Juggler* word processor.

This article encompasses a comprehensive review of *Word Juggler* and all of its associated support and enhancement programs. *Word Juggler* itself will be reviewed in this issue, while the supporting software packages listed below will be reviewed in the next issue:

- Word Juggler • A very powerful word processor
- Lexicheck • Spelling checker
- Mail List Interface • Interface to Apple's Mail List Manager
- Typeface • Formatter for output to a typesetter
- Discourse • Software printer spooler

A little history is in order here so that the reader can appreciate the strong feelings that I hold for these products. Since I purchased my first Apple II back in 1977, I have been on the look-out for the best word processor around. This journey has led me through just about every word processing product on the market. Until *Word Juggler*, I hadn't found one that met all of my requirements.

I settled on using a Z-80 card and a CP/M based word processor. While it's true that most of the CP/M based word processors have just about any feature that anyone would ever want to use (and some that you'll never use), there are two reasons why my search went on.

One, why did I buy a 6502 micro-processor based computer if I was going to do all of my word processing in the Z-80 environment? It seems a bit hypocritical.

Two, these word processing programs were written for use under the CP/M operating system and therefore operate in just about any machine that supports CP/M. They rarely, however, take advantage of many of the hardware features of the machine that they are running on. Running under CP/M on the Apple II, the video screen I/O is extremely slow and the document files that you create are foreign to the normal Apple disk environment. Because

they are non-standard, it means that without conversion programs, you would be unable to access any of the CP/M-created documents within the Apple Pascal, DOS 3.3 or ProDOS environments.

What I really wanted was a word processing system that fully supported the Apple, operating in its native environment, yet was powerful and easy to use. With the design limitations of the Apple II computer (ie. no 80-column screen, no lower case display or input and memory limitations), I turned my attention to the Apple ///.

While at an Apple Software Vendors Conference in 1982, I heard from several people that Quark Incorporated had a word processor for the Apple /// that sounded like it had everything I wanted. Except for a few minor points, they were right!

Word Juggler:

The *Word Juggler* package comes in a handsome three-ring binder complete with manual, master and backup program disks (both of which are uncopyable), an auxiliary command reference card and two keyboard templates. One of the templates fits over the numeric keypad while the other sits over the numeric keys on the main keyboard. These templates are a handy reference to most of the commands that *Word Juggler* recognizes. They are a lot handier to use than the standard reference card that you get with most software packages.

The manual contains a table of contents, index and glossary. Each chapter and the appendix are attractively separated by colorful tab sheets. The chapters contain a tutorial, a reference guide and a description of the data file merge capabilities. The appendix describes how to backup your data disks, how to recover from errors, the formats of the data files and printer configuration.

The Word Juggler program is written completely in assembly language and is configured as a SOS interpreter. This means that the program is fast and does not depend on any other language environment in order to operate. It is also the host program for a multitude of utility programs that function from within the word processor. Some of these utilities will be reviewed in part two of this article.

You should be familiar with SOS drivers and have one configured for your printer.

Before you can use Word Juggler you must configure it for use with your printer. You should be familiar with SOS drivers and probably already have one configured for your printer to operate from within another interpreter such as Pascal or BASIC. It is a relatively simple matter to transfer your printer driver from one of your other diskettes to provide Word Juggler with the information that it needs to be able to recognize and operate your printer.

At the same time, if you have a hard disk, you may also transfer its driver to the Word Juggler disk. Through SOS, Word Juggler will support any printer or storage medium that you can configure.

Any configurable printer or driver is supported.

After you have transferred the appropriate driver(s) to the Word Juggler disk, the disk should be booted. The following screen depicts the main menu that you will be presented with that will allow you to enter the word processor for editing, perform file loads or saves, directory catalogs, disk formatting, purging of disk files, definition of the SOS file prefix, changing the printer configuration or ending the program:

You should first select menu option eight (8) to change the current edit

Word Juggler (version 2.6)

Options:

1. **NEW**—Erases all text and goes to text entry mode.
2. **CATALOG**—Lists all files in a directory.
3. **LOAD**—Loads a document from disk and goes to text entry mode.
4. **STORE**—Stores a document on disk.
5. **PURGE**—Removes a file from disk.
6. **FORMAT**—Formats the diskette in the internal drive.
7. **DEFINE PREFIX**—Defines the prefix to be used for disk access.
8. **EDIT CONFIGURATION**—Allows printer and default parameter selection.
9. **QUIT**—Exits the Word Juggler program.

Which Option (Press RETURN for text entry mode)?

configuration. This allows modification of the many program default values such as margins, page length, number of characters per inch and printer criteria.

After that is done, you are ready to start using the program by selecting the 'New' option from the menu. Upon your selection to enter the editor, you are presented with a blank screen with a status line at the bottom. The status line keeps you

informed of the name of the file you are working on, the line number and horizontal column that you are positioned on and the amount of free space left in memory for your document.

You have room for 1536 lines of text with a 256K Apple ///, which is about 143K. Memory is no problem with the Apple /// and Word Juggler takes advantage of the fact. In addition, it recognizes but does not utilize the memory on the newer 512K upgrades, and in fact does not use *all* of the extra memory on a 256K machine, only enough to fill a 143K diskette.†

The complete program is co-resident in memory with your document.

The complete Word Juggler program is co-resident in memory along with your document. Other word processors have to swap different portions of their program in and out of memory to maximize the amount of space for your document. One big benefit of co-residency is immediate response times for text manipulations.

†On *Three* would like to offer a 512K upgrade to Word Juggler and is currently negotiating with Quark to do so. ...ed

Editing Enhancements

Word Juggler makes full use of the special keys on the Apple ///. The four arrow keys as well as the TAB key are used to move through your text. Table I defines how these keys are used:

Table I				
KEY	By itself	Control key	Shift Key	Control/Shift
→	right one space	right one space	next word	end of line
←	left one space	left one space	previous word	start of line
↓	down one line	down one line	next page	end of text
↑	up one line	up one line	previous page	start of text
TAB	go to next tab	clear tab	set tab	tab (w/spaces)

Needless to say, it's a lot easier to use the four arrow keys and their permutations rather than having to learn different keys for every cursor movement. The setting and clearing of tabs is also simple and these tab settings are stored along with your document when you elect to save it to a storage medium.

My first impression while using Word Juggler was the phenomenal speed of moving through your document. It's a real joy to be able to zip through your text, scrolling from page to page, almost faster than your eye can follow!

... full word wrap is utilized.

Word Juggler also utilizes full word wrap to keep your document efficiently compressed. When you delete text on one line, Word Juggler will pull the next word (or words) up to fill in the gap, deleting blank lines automatically.

Word Juggler has a complete set of text manipulation tools to allow you to move, delete, copy or insert text anywhere in the body of your document. A single keystroke invokes these commands. These commands are accessed by typing a key on the numeric keypad by itself or in conjunction with Shift and/or Control keys. A template is placed over the keypad to aid in making your selection.

For instance, you may want to copy a segment of your text to another location in the document. By touching the number five (5) on the numeric keypad, you are prompted (in inverse video at the bottom of the screen) to "position the cursor to the start of the segment that you wish to copy and press the space bar." Next you are asked to do the same for the end of the segment. The segment to be copied will be highlighted in inverse video to show which portion of your document will be copied. After doing so, just place the cursor wherever in the document that you wish to insert a copy of this text and press the space bar to complete the command.

Basically, the same process is repeated for deleting, moving, writing or reading text segments from the storage medium. Other systems re-

quire that you place tokens in your text to outline the segment that you want to manipulate. The way that Word Juggler does it seems much easier to me and the instantaneous response is rewarding.

The search and/or replace commands are fast and powerful. You may search and/or replace any text from the start of your document (or from the current cursor position) to the end. You may also elect to manually verify each replacement or allow Word Juggler to make the changes automatically.

One feature lacking in the search and/or replace command is the ability to find or change the document enhancements (margins, center text, etc.) This is because they are specially coded in your document and not the ASCII representations.

With any of the Word Juggler commands, you have the option of using the escape key to abort an undesired operation before you make any changes. Most word processors 'lock' you in to the command until it's completed. A chart of the editing features follows to allow you to acquaint yourself with their completeness:

Editing Features

Screen scrolling:

Vertical and horizontal.

Jump features:

Text beginning or end, previous or next screen (page) and beginning or end of line.

Block features:

Copy, move or merge.

Insert features:

Character, word, line or block (with word wrap).

Delete features:

Character, word, line, rest of line, block and page.

Search & replace:

Search only from cursor to end of text or from beginning. Search & replace from cursor to end of text or from beginning automatically or manually.

Screen format:

Margins, tabs, justification, page length and width.

Paragraphic:

By manual spaces, by tabs or by hitting the return key.

Printing Enhancements

Every word processing program allows you in some way to specify how you want the output formatted. Some items of concern are margins, headers, footers, character density, justification, line spacing and character enhancements (bold, underline, subscript or superscript).

What sets the good word processors apart from the not-so-good ones is the ease of entering these enhancements into your document and, once invoked, their power.

Word Juggler has a full complement of enhancements that allow you to format your document just about any way that you desire.

Most word processors have you place 'bang' statements in your document to inform the system that some enhancement is to be performed. A typical enhancement in every word processor is the new page command. 'Bang' statements from a few of the leading word processors follow for your comparisons:

Wordstar	.NP
Applewriter ///	.FF
Script ///	-BP

These cryptic 'bang' statements are hard to understand if you haven't memorized them and also make it difficult for someone else reading your text who doesn't know what they mean. Until you get to know them, you most likely will have to refer to a 'command chart', which is distracting. In addition, they typically take up at least three bytes of storage in your text.

... insert any enhancement with two keystrokes.

Word Juggler allows you to insert any enhancement into your document with a two keystroke command. The sequence is first to touch the escape key and then select a key command from the template located over the main keyboard. Word Juggler inserts some coded information into your document and then displays a line of very descriptive text on the screen. For the above example, you would see:

NEW PAGE

The coded information that is placed in your document is not the text that you see on the screen, but the actual keystrokes that you used to generate them. For the new page command, you would have touched the 'escape' key (hexadecimal code \$1B) and the '^' key (hexadecimal code \$5E). Some of these commands prompt you for more information, such as when you are modifying the margins. Following is a chart of the printing enhancements:

Printing Features

Justification:

Ragged right, ragged left, center and fill justify with micro-spacing.

Paragraphic:

Indentation, line(s) between paragraphs.

Headers/footers:

With or without page numbering on any line on the output page.

Line spacing:

Single, double or triple spacing.

Margins:

Right, left, bottom and top.

Pitch:

10, 12 or 15 characters per inch.

Enhancements:

Boldface, underlining, boldface/underline, superscripts and subscripts.

Boiler plate:

Ability to insert other documents or any ASCII type of file (e.g. output from VisiCalc, Quickfile or PFS:File).

Paper type:

Single sheets, continuous forms and multiple copies.

Printer commands:

Ability to pass codes to your printer to use enlarged characters, different vertical pitches, etc.

Print to where?:

Can print to the screen for document preview or to your printer. A nice touch is to be able to select any page or a range of pages from your text for output.

While Word Juggler is not a true 'what-you-see-is-what-you-get' type of word processor, it comes very close, since you can select to display the output to the screen or to the printer with a single keystroke. You may also elect to only output specific pages.

... you can display to screen or printer.

For example, if you are working on a long document and have the first twenty pages complete but want to see how the twenty-first page will look, you may choose to do so. You might also desire to print multiple copies of your document. Word Juggler allows you to specify up to 254 copies.

When you send the output to the screen for preview, before actually printing it out, your document is justified, page breaks are displayed and all your print enhancements are evident. Even the sections of your text that will print in bold face or underline will be displayed in inverse to distinguish it from the surrounding

If the output is wider than 80-columns you scroll horizontally with the arrow keys.

text. On most word processors, if your output is wider than the eighty column screen, you won't be able to see all of it at once. Word Juggler allows you to scroll back and forth horizontally using the right and left arrow keys to allow you access to whatever is beyond the screen boundaries.

Bold face, underline, superscript and subscript are specified by surrounding the text in your document that you want enhanced with the appropriate unique symbols. These symbols are generated by typing a combination of the 'open apple' key and one of the following:

Key	Function	Displayed Symbol
b	bold on	inverse 'b'
B	bold off	inverse slashed 'b'
u	underline on	inverse 'u'
U	underline off	inverse slashed 'u'
↑	superscript on	inverse up arrow
↑	superscript off	inverse up arrow
↓	subscript on	inverse down arrow
↓	subscript off	inverse down arrow

Typing Mode

You may also use Word Juggler as an electronic typewriter. By touching the number two (2) on the numeric keypad, while holding down the control and shift keys, all subsequent output is sent to your printer if it is a serial printer. Parallel printers cannot support this due to the document text being transmitted line by line. This includes using the four arrow keys to move around on the page in your printer (only if your printer supports reverse linefeed, etc.)! This is very nice for general typing or filling in forms that you really don't want to create a special document for.

Another typing function, available on the numeric keypad, is initiated by simply touching the number two (2). The line that the cursor is positioned on will be sent to your printer. This command is used for transmitting just a single line of text from your document to the printer. A typical example would be to take an address from your document and place it on an envelope.

Data File Merge

Word Juggler also has a powerful data file merge capability that will allow you to construct form letters or mail labels.

By placing variables into your document, Word Juggler will ask for the text to be substituted for these variables at the time that the document is to be printed. These variables

are entered into your document by touching the 'open apple' key along with the following:

Key	Function	Displayed Symbol
<	start of variable	inverse '<'
>	end of variable	inverse '>'

You may place up to 127 variables anywhere in your document for Word Juggler to process. Each variable may be up to 78 characters in length, but blanks, or print-out enhancements are not allowed. Whether the variable is in upper or lower case makes no difference to Word Juggler (e.g. <ADDRESS>, <address> or <Address> are the same). When you define variables in your document, Word Juggler will take away two lines of storage from your document to make room for the text that will be substituted for the variable at print time.

Word Juggler has some pre-defined variables that allow you to access the Apple ///'s internal clock. (For those of you that have one!) They are entered just like any other variable, but are automatically filled in for you if Word Juggler can find the internal clock. A list of them follows:

Variable Name	Description
\$Date	Gives the date in the form MM/DD/YY (e.g. 02/15/86)
\$Year	Gives the four digit year (e.g. 1986)
\$Yr	Gives the two digit year (e.g. 86)
\$Month	Gives the month name (e.g. February)
\$Mon	Gives a three letter all caps abbreviation (e.g. FEB)
\$Month	Gives the month number (1 to 12)
\$Day	Gives the day name (e.g. Monday)
\$Day#	Gives the day of the month (1 to 31)
\$Time	Gives the time (e.g. 5:42 PM)
\$Time24	Gives the time in a 24 hour format (e.g. 17:42)

```

LET
<PRINT.THIS.ONE>="y"
IF
<PRINT.THIS.ONE>="y" | <PRINT.THIS.ONE>="Y"
PAGE LENGTH 66
LENGTH 54
TOP MARGIN 7
LEFT MARGIN 10
WIDTH 70
10 PITCH
SINGLE SPACE
RAGGED RIGHT
<$Month> <$Day#>, <$Year>

<Name>
<Address>
<City>, <State> <Zip.code>

Re: Past due account number <Account>
    Past due amount <Amount>

Dear <NAME>:

It has come to my attention that your account is somewhat delinquent.
Since you have ignored all of our notices in the past, I thought I
would take this opportunity to write you a personal letter.

If we do not receive payment for the past due amount, we will have to
send out our bill collector, Bruno, to break your fingers, one at a
time. Now, this is not a very pleasant prospect, but I'm sure you can
appreciate our position in this matter. We will be looking forward to
receiving your certified check in the mail in the next few days.

Sincerely,

T. P. Barnum
Gorilla Trainer
Barnum's Collection Service

END IF

```

-- Figure 1 --

These come in very handy when you are processing a standard form letter. By using these variables, you can be assured that each time you print the letter out, the correct date and/or time will be automatically furnished. Please refer to Figure 1 to see how variables are used in a sample form letter.

IF, ELSE, END IF and LET commands are available.

Word Juggler also has powerful commands to allow you to select only the variables or sections of your document that you wish to print. These are the *IF*, *ELSE*, *END IF* and *LET* commands. Expressions may be evaluated containing literals or other variables with the *IF* command using the following comparison operators:

Operator	Function
<	less than
>	greater than
<=	less than or equal to
>=	greater than or equal to
=	equal to
<>	not equal to
&	AND
	OR
—	NOT

In order to print out form letters or mailing labels, you must first create the letter (or label format) that you wish to use and then create a list of the data that you wish Word Juggler to substitute for the variables in your document. You may use Word Juggler to create this list or you may take input from any Apple /// ASCII file (as long as the format is compatible).

For example, you may use Quickfile or Versaform to create a disk file report that only contains the information that Word Juggler needs to fill in your document (e.g. names and addresses). Refer to Figure 2 for an example of a mailing label format and Figure 3 for an example of the data file.

```
TOP MARGIN 1
LEFT MARGIN 4
PAGE LENGTH 6
LENGTH 5
WIDTH 42
<Name>
<Address>
<City>, <State>
<Zip.code>
```

-- Figure 2 --

```
5
Name
Address
City
State
Zip.code
```

```
Bob Jones
25 West 125th St.
New York
NY
10036
```

```
Nancy Mc Callum
120 River Rd. Apt. 3
New Ulm
MN
55346
```

-- Figure 3 --

Built-in Utilities

In addition to being able to use data from any ASCII file for form letters, Word Juggler can directly access data files from Quickfile or PFS. This means that you don't have to create a special output file from either of these two programs but can get at their information without having to leave Word Juggler. The category names that are used in your Quickfile data file would be the same variable names that you would use in your form letter document. This makes it tremendously easy to process information contained in either of these two data base programs.

Word Juggler doesn't stop there. Through the use of a special 'Utilities' menu, you may also take input from an Apple II DOS 3.3 text file or Pascal text file. Other useful utilities are included as well. A complete list follows:

1. Insert DOS 3.3 text file
2. Insert Apple Pascal text file
3. Set time and date
4. Create subdirectory
5. Install user printer filter
6. Recreate parameters file
7. Define default prefix
8. Define external procedure path

Word Juggler is close . . . to a perfect word processor.

Summation

Word Juggler is as close as I have come to a 'perfect' word processor for the following reasons:

1. The speed at which it operates is fantastic.
2. Learning how to use it is a relatively painless experience.
3. It's very powerful in terms of its editing and formatting commands.
4. The support programs that Quark offers to enhance Word Juggler.

There are some problem areas however that need to be touched on. The manual is missing a program statistics page. This is a page that

should contain all the specifications of the program; the memory storage capacity, maximum number of variables, maximum and minimum values that can be used with the commands, etc. The information that should be on this page is in the manual, but you have to search for it. Also, a few pages devoted to layouts of 'standard' business letters would be appreciated by the novice user.

While most people will use Word Juggler and never need or wish for more features, I have compiled a list of those I feel would make nice additions to the program.

Suggestions

1. There doesn't seem to be any way to have Word Juggler use different pitches other than the supplied 15, 12 and 10. According to the manual, one should be able to use the PRINTER CONTROL command to pass control characters to a printer. In the case of a DIABLO 1640, it would be nice to be able to change the HMI (pitch) by passing the proper sequence. Word Juggler seems to intercept or invalidate any attempt by the user to modify the horizontal motion index. While true that most people will only use the supplied pitches, I find that the ability to use other pitches is critical in the preparation of documents other than common letters.
2. The ability to print in reverse as well as in forward mode would enhance the printer output greatly. Some printers have this feature built-in, but others must be sent a special 'control' character and the following line of text in reverse order. Word Juggler will work with the printers that have this built-in, but won't work with the others. Take note that I am not complaining about the speed at which Word Juggler sends output to the printer, but only the suggestion that greater efficiency of more powerful printers would be a boon.

(Cont'd on page 31)

Now About This Basic Business...

val j. golding

Speaking of Basics...

So you have an Apple ///—a machine most often perceived as a purely business machine—certainly not the newest around, and very, very definitely (albeit under-recognized) one of the finest computers ever designed. Maybe in the “deal” when you first purchased it, some software came with it, perhaps Applewriter /// or /// E-Z Pieces. Somewhat boring programs that work reasonably well and help justify your expenditure, but mainly they just aren’t much f-u-n. Before you relegate the machine to your kids, let us show you how easily you can have fun writing your own programs that play games, balance your checkbook or just let you add a bunch of figures. After all, who wants to boot /// E-Z Pieces and wallow through menus and submenus just to do simple addition.

... who wants to boot /// E-Z Pieces to do simple addition.

Computers are really pretty dumb devices, a collection of chips that cannot respond to anything more earthshaking than changes in voltages which, like an electric light switch, turn memory cells, or registers, on and off. Programmers are what give computers their smarts, using *programming languages* to instruct the computer in various phases of operation.

When we talk about computer languages, we use language in its accepted form. If you analyze a sentence written in German, English or Swedish, you will find common phonemes and spellings. Each sentence conveys to the reader the same information; it is only the form that differs.

So it is with computer languages. There are as many different languages and dialects as there are spoken and written languages as we know them.

They range from low-level languages such as Assembly (compare Latin) to high-level languages like BASIC (compare English). These include Pascal, C, Logo, FORTRAN and many, many others, most of which have found their way to implementation on the Apple.

It is quite possible and indeed reasonable to use an Apple or other microcomputer without ever acquiring programming knowledge. There are vast numbers of commercial programs that can fill your needs, from forecasting annual budgets to teaching your dog to sit up and bark. It is also true that most commercial programs can rarely be tailored to your specifications. They are designed as general-purpose applications and many are copy-protected, a nefarious scheme which normally prevents user modifications as well as access to such niceties as hard disks.

Let’s look at a concrete example. You have a hand-held calculator which serves most of your household requirements. You use it to add or subtract entries in your checkbook; you use it to calculate how many miles per gallon your automobile gets, but today, your spouse is out shopping and has taken the calculator along. The mail has just arrived and with it is a monthly bank statement. At the moment, you’re alone and the time is ripe to balance the checkbook, but you have no calculator (heaven forbid we should use pencil and paper). Aha! Your trusty Apple /// to the rescue. With a simple one-line Business Basic program you can add and subtract numbers to your heart’s desire. Just look at the following program. You can type it in and run it without knowing a single thing about programming.

```
100 INPUT CASH:BTMLINE =  
BTMLINE + CASH: ON CASH  
< > 0 GOTO 100: PRINT  
BTMLINE
```

When you run this short program, you will see a question mark printed on the screen. This is a prompt to tell you to type in a number. If you type a zero, then program *execution* terminates after printing the addition total. If you enter a minus sign before a figure, it will be subtracted instead of added. Easier, we’ll bet, than opening a jar of mayonnaise!

You don’t need to decide right now whether you want to learn enough programming to get by on, to go whole hog and learn to write your own sophisticated programs, or to just ignore the whole mess. For now, just take your computer in hand and follow us while we show you the ropes.

From the Ground Up

Since you’ve gone this far, stick around a bit more for a glimpse of what’s in store for you. Before imparting an iota of programming information, there are a few prerequisites. We’ll describe, in fair detail, what the BASIC programming language in general, and Business Basic in particular, is all about, why you need to back-up data, what a program really is, why computers use such a strange number system, and the function of a *language interpreter*.

One of our goals is to avoid long, drawn-out technical discussions. So that you can progress rapidly, we want to simplify things as much as possible and make them easy for you to understand. In some cases, precise technical accuracy may be slighted in favor of simplicity and understandability. Nevertheless, there are occasions when we need to veer from the straight and narrow, thus as we encounter computer-specific terminology, we may pause in our programming adventure and define a term or three, while in other cases it may suffice to denote them with italic type. Some of the latter may be defined at a future time.

Assuming you have cleared the screen beforehand, this is what you would see, once you have pressed the return key:

```
(PRINT "MY NAME IS VAL"  
MY NAME IS VAL  
)
```

This is nothing more than a program without line numbers. The parser will tokenize the word "PRINT" and immediately interpret and execute it, just as if it had been encountered in a numbered program statement.

The Key to Programming

Keywords or commands are what we use to tell BASIC what we want to happen. Business Basic has 109 of these commands at your beck and call. In the next issue, we will describe what some of them do and present a table, grouped by function. Business Basic commands are simple English-like words, descriptive of their function. Suppose you were to write a "program" to pour a cup of coffee. It might look something like this:

```
GOTO KITCHEN  
GET CUP FROM SINK  
PICK UP POT  
POUR COFFEE INTO CUP
```

And now we have a full cup of coffee, right? Yes, and more so. In fact, the cup is overflowing and we have coffee all over the sink. Why? We forgot one important instruction:

```
IF CUP FULL THEN STOP
```

and so it is with computer logic. We must remember all of the details, all of the ramifications. BASIC is very

BASIC is unforgiving of our errors.

unforgiving of our errors (remember when we said the machine itself knew nothing more than to recognize switch settings, on and off). When we write a program, we must always keep logic, computer logic, in mind. If we do not do so, we will always find problems with our programs. BASIC will always do exactly as it is instructed, no more, no less. You must always be specific to the nth degree.

Go With the Flow

If you've read many computer magazines, you're probably familiar with, or have at least seen, *flow charts*. Those funny drawings with boxes and diamonds and circles and arrows pointing every whichway. Do they really serve a purpose? Yes, if you're a professional, or if you're into a terribly complex program. The real question is do you need them, and the answer to that, for now at least, is a resounding "no." However, lest you think we mean you should forget about the whole idea, think again.

While a full-blown flow chart is not called for, you still need a method of outlining your masterpiece-to-be. One of the simplest and easiest to use is plain old-fashioned scratch paper.

... outline your masterpiece-to-be.

Gad! whoda thought of that? Actually, the idea is to ask yourself a whole bunch of questions. Before you ever type in a line of code, you are going to need to know things like:

- do I need a menu?
- where do my inputs come from: disk, keyboard or both?
- how do I format the output and where does it go?
- what specific solutions (algorithms) do I need?
- will I have frequently-used variables to declare first?

The list above is by no means complete, nor is it intended to be. Use it as an example to get your mind working ("my brain hurts") on the program and thinking about all aspects. Try to cover all bets, all possibilities. If you start now, as you learn to program, the process will quickly become automatic and sooner or later, help you out of a tight spot.

Taking Command

Now, don'tcha go way mad. In fact, don't go 'way at all, 'cause we're gonna introduce you to a couple or so commands that can start you on your way to becoming an expert Apple /// programmer. Each of the following commands will be covered again later, under their appropriate grouping.

For now, we want to show you three of the most often used commands, so that you have something to do while you are waiting with baited breath for your next issue of *On Three*. Each of the following commands may be used in either direct or program mode, but you'll find the usefulness of the latter two pretty much limited to direct mode. (Remember, we told you all about that stuff a few paragraphs back).

PRINT RUN LIST

RUN and LIST are system commands, those commands which help you control what BASIC is doing at any given time. You will find them among your most helpful and useful commands. RUN is the instruction that tells the BASIC interpreter to execute the program you have loaded into memory or typed in from the keyboard. It may be used with a modifier, so that you can specify at which program line number execution will commence.

Many keywords are considered verbs, which denote action. As in the English language, Business Basic verbs may be modified, as we shall see later on. PRINT is one such BASIC verb and is used in a number of different ways. To start you on your way to programming, we'll use it here to print a message to the screen, and in a couple of minutes you will have written your first program! Just follow step by step the simple instructions listed below.

1. Type (without the quotes) "NEW" and press return. *This will clear any existing program from your Apple ///'s memory.*
2. Type "10 PRINT "HELLO"" and press return. *You have just entered a program line into program memory.*
3. Type "LIST" and press return. *The line you just typed (10 PRINT "HELLO") will appear on your screen.;*
4. Type "RUN" and press return. *This time just the word "HELLO" will appear on your screen. Whenever you type "RUN", the program currently in memory will execute.*

In each issue, we will set aside a little space to offer you some special programming “tricks”—information you’ll not find in the programming manuals. In addition, we’ll offer guidance on how to program more efficiently and furnish some “quick and dirty” programs that you can use right away.

Fear not, intrepid traveler, the road to programming proficiency is not fraught with peril. The Eleventh Commandment (here and now just invented) states: “Thou shalt not fear thy computer.” Indeed, as we indicated at the outset, the computer is only a dumb machine. You are the master and it is the slave. While it falls somewhat short of being able to sit on your lap and take dictation, it can do literally almost anything else.

“Thou shalt not fear thy computer.”

Early on, one of our mentors passed on these wise words: “Never do yourself what your computer can do for you.” Moreover, we’ll hang around here should you run into problems. Reader response is an important aspect. We will welcome your calls and letters with programming questions and may select for publication those that can be of assistance or benefit to others.

“Sidebar” is legal terminology for an area in a courtroom where counsel and court can get together to discuss points of law or evidence which may not be proper for a juror’s ears. It has been applied to the publishing industry to mean (usually) a boxed area of type pertinent but not intrinsic to the main topic, so here in the following sidebar, we want to talk about the need to back-up your programs and data files. For those of you who know your way around, you may feel free to proceed to the section following.

Down to the Nitty-Gritty

Computer languages are of two general types: *compiled* and *interpreted*. In a compiled language, the *source code* (what the programmer has typed in) is processed and converted to either *pseudo-code* or *machine code*, resulting in faster program execution. Business Basic is an interpreted

Tricks of the Trade

Take, for example, a disk operating system (DOS). In the Apple III, SOS (Sophisticated Operating System) occupies some 24k of program memory, a fair share of the total available. While many safeguards have been built in, the amount of code required to fully protect against user errors might well double, to say nothing of slowing execution speed. As users, we must always be aware of these limitations and provide for ourselves the maximum amount of safety and security possible when working with disk files.

A few brief rules of thumb, then, to save you from yourself.

- 1. Always have a blank, formatted diskette handy. This will save problems when you inevitably run into a DISK FULL error.*
- 2. Never save a program to an existing file until you have verified (by listing) the integrity of the program in memory, and (by running), that it executes correctly or as you expect. Many hours of work have been lost by saving the most recent version of a program to an existing file name, only to later discover that major bugs had cropped up in that version. When we are developing a program, we use file names such as PROGRAM.1 and PROGRAM.2 for each consecutive save. Diskettes are cheap; your time is not.*
- 3. Save often. The briefest of power failures can wipe out hours of work.*
- 4. Diskettes are fragile. Quality varies from brand to brand and from disk to disk. Minute surface imperfections can crash an entire file beyond ordinary recovery. At the end of each work session, use your System Utilities diskette to make a complete copy of your current work disk.*

language. (There are compiled BASIC’s as well, although none are available for the III.) At *run-time*, the Business Basic interpreter reads the source code, statement by statement, interprets each individual character and calls the appropriate routine within the interpreter to take whatever action is indicated. Final *syntax* checking is

also done at this time. Thus, as may be seen, there is usually no intermediate step in an interpreted language, and the (*tokenized*) source code becomes the *object* or run-time code.

When you type in a Business Basic program line, the BASIC interpreter places each character in the *input buffer* (a buffer is a temporary data storage area) and calls the BASIC *parser*. The parser is a routine in the interpreter that scans each program character as it is typed and then decides, based upon the nature of the character, how it is to be stored in memory.

In order to conserve memory space and to most efficiently interpret program text, each time the parser finds a string or sequence of characters such as P-R-I-N-T, and if the sequence matches a Business Basic command or keyword, it will replace the character group with a single special character, known as a *token*, which thereafter represents the character sequence that forms the command name. When a program is listed, the parser performs the tokenizing process in reverse.

Only keywords are tokenized.

Each time it encounters a token in the program text, it refers to a table, called a *look-up table*, and finds the appropriate ASCII (text) characters to print to the screen. Note that only those character groups which form keywords are tokenized. All others are left in their original (ASCII) form.

Why Put it Off?

Actually, a program can be executed (run) in either of two ways. We use the terminology “deferred mode” to mean a program that has been typed in using line numbers and saved to disk, which can later be loaded and run as many times as you wish. This is also sometimes called “program mode” or “under program control.” The term “immediate mode” indicates a program without line numbers which is typed at the keyboard and executed immediately, i.e., as soon as you press the return key. An example of such a program would be:

PRINT “MY NAME IS VAL”

Congratulations! Even though it was only one line long, you have just written and executed your first computer program. (Notice it is only the the addition of a line number (10 in this case) that distinguishes this *deferred mode* program from an *immediate mode* program such as the earlier example: **PRINT "MY NAME IS VAL"**).

Updating Your Memory

Now let's bring you up to date. We've offered you the opportunity to learn how programming on your Apple /// vs just using it can help you best utilize your computer in a home or small business environment. We like to think your decision will be to take advantage of how writing your own programs, in addition to utilizing "canned" software, can offer the best value for your hard-earned money.

We described in brief, some of the different computer languages that run on the Apple, and hope we whetted your interest sufficiently to explore beyond the boundaries of this series.

We went into a little more detail in describing the BASIC language supplied with the Apple ///, and offered the simplest of programming examples to add a few figures together. Even if you proceed no further, we hope you will put this quickie to immediate and practical use.

... photocopy each article ...

The suggestions we offered about backing up your precious data should be rigidly observed, irrespective of your application.

If you've decided to stick with us, we'd like to suggest that you photocopy each article of this series and put the copies in a looseleaf binder for easy reference. Upon re-reading this material some three or four months in the future, you may be surprised to realize how much of what made no sense whatever will then seem very logical and easy to understand. In addition, if you have not already

done so, you should pick up a copy of the two-volume "Apple Business Basic Reference Manual."

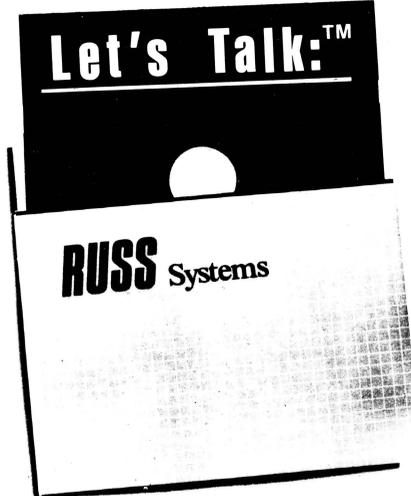
Next time around, we'll really get into programming; we'll discuss the first batch of BASIC commands and show you how to put them to work for you, on the spot. Meantime, we suggest you put in a little more groundwork: skim through the manuals that came with your Apple and see if you can anticipate some of the things we'll talk about in future articles.

Never do yourself what your computer can do for you.

By way of saying "so long," we'd like to refresh your memory (RAM?) and remind you of the thought we expressed midway through this discourse: *Never do yourself what your computer can do for you.* Think about it! 

"You Couldn't Have Made It Any Simpler! Now Everyone Can Afford To Go On-Line"

Shaun Ralston on-line from Personal Computing Magazine.



Apple /// - 256k native mode

Let's Talk supplies SOS - PRODOS to provide the operator with an organized multi-level filing system.

TURN YOUR Apple INTO A FULL-TIME BUSINESS PARTNER

Let's Talk,™ a revolutionary new telecommunications program that operates your Apple unattended as a sophisticated answering machine, and information server.

With **Let's Talk**, your computer has the power to operate your own micro version of Dialog or Compuserve. Services that formally required a mainframe can now be offered on a smaller scale with an Apple computer from your home, office, or school. And you decide what goes online.

Let's Talk is the electronic link that puts your information online. Now your computer can advertise and support products and services, list catalogues, and accept orders while providing electronic mail, and information to callers any time, day or night. All they need to get your message, or send you one, is any computer and a modem.

What's more, callers can upload and download files or create them easily and quickly because **Let's Talk** doesn't require any complex commands to operate. Press a designated key and a file opens. You can also protect selected files or the entire system with a built-in password option.

So get yourself the most sophisticated program available for electronic information exchange.

Let's Talk™ DATABOARDS

Easier to use than an on-line DATA base . . . More professional than a bulletin BOARD.

For 24 hour product support and for more information, HAVE YOUR COMPUTER CALL 408-427-0310

8 bits, no parity, full duplex

Let's Talk for Apple IIe - IIc - III — \$125 plus \$3 shipping and handling

RUSS Systems 408/427-1540
320 Dufour, Santa Cruz, CA 95060

The Easy Image Maker

rick gauger

Introducing ImageHelper and ImageFileLooker, two Business Basic programs that take the work out of creating, storing, and inspecting images for DRAWIMAGE to draw on your console screen.

DRAWIMAGE: The Not-Quite Instant Image

DRAWIMAGE is the name of an assembly language routine inside BGRAF.INV, the Apple ///'s graphics package. The Business Basic statement PERFORM DRAWIMAGE commands your Apple /// to "rubber stamp" an image on your console screen almost instantly, in much the same way that a character appears immediately when you type it. DRAWIMAGE offers the possibility of creating animated figures like Ron Puckett's Walking Man, which appeared in the June-July 1983 *On Three* (Vol. 1, No. 4). Ron's animation only used three frames to achieve its effect. There's no reason why you couldn't use dozens of images, appearing in sequence on the same spot, to obtain truly Disney-like animation.

But, there's no such thing as a free lunch or graphics without drudgery. Before you can understand how ImageHelper eliminates most of that drudgery, you have to understand what you (or ImageHelper) must do in order to make an image for DRAWIMAGE to put on your screen. Before reading any further, take out your copy of the Apple Business Basic Reference Manual, Volume Two, and turn to Pp. 314-315.

An image consists of a two-dimensional array of dots, like the squares on a sheet of graph paper. The array is called a source block. The manual shows you how to design a source block by filling in graph paper squares in the pattern you desire. Now think of your pattern as an array of bits, with the filled-in squares as "ones" and the blank squares as "zeros." You can see that each row of

bits can be divided into groups of four bits. Each group of four bits constitutes a byte. These bytes are numbers, and that's how the Apple /// "sees" the image, as a series of numbers.

Next, the computer stores the array in memory. You provide an integer array for this by using the DIM statement. The array in the manual is arbitrarily called SOURCE%. It is created by the statement DIM SOURCE%(1,4).

Now, one of the rules an integer array follows is that each element (for example SOURCE%(1,1), SOURCE%(1,2), and so on) must contain a pair of bytes, or sixteen bits. Therefore, the width, in bits or graph paper squares, of your source block must be an even multiple of eight like 16, 32, 48, etc. However, the source block can have any height you want, so long as it fits on your console screen.

Now comes the pain in the neck. Turn to P. 316 of the manual to see what it is. Oh no! You have to translate the lines of bits into four-digit hexadecimal numbers before they can go into SOURCE%. For example, 1001 1100 0111 1011 translates to 9C7B. Only then can you use the TEN function to convert the hexadecimal number into a form SOURCE% can store, as in TEN ("9C7B").

ImageHelper Does It For You

Don't despair, relief is on the way. ImageHelper does all the translating and array filling for you. Here's how: Lines 580 to 730 constitute the source block for an image of a skull face. Each line is a DATA statement followed by a row of ones and zeros in quotation marks. To create your own image, just LIST those lines and change them to show whatever image you want.

ImageHelper processes those image data lines (580 to 730) byte by byte thusly:

Line 1150 takes each line of the source block and puts them into the elements of a string array, Bitbloc\$.

Line 1250 makes a reference list (Reflist\$) which becomes an index of binary bytes ("1011") and their equivalent hexadecimal digits ("B").

Line 1420 burrows through each line of Bitbloc\$, taking out each group of four zeros and ones, and uses the Reflist\$ to look up the group's equivalent hexadecimal digit, for example, 1011 equals B. The digit is then added to the end of a string variable called Hexdigit\$. Line 1440 causes the program to branch to subroutine 1490 whenever Hexdigit\$ has accumulated four hexadecimal digits. The four digits constitute two bytes (a "byte pair") that can be inserted into an integer array.

Line 1580 feeds the contents of Hexdigit\$ into the string array Hexnumlist\$. Then Hexdigit\$ goes back to line 1420 empty and ready for the next four hexadecimal digits. Subroutine 1490 also displays each pair of bytes so you can see what the hexadecimal numeric equivalent of each eight bits of your image is. You can modify lines 1600 and 1610 to print these numbers on paper if desired, but you don't really need to know what they are because they can be stored in a file without any further bother.

During the Display Image subroutine, the contents of Hexnumlist\$ are fed into SOURCE%. One of the differences between the way the manual does it and the way the program does it is that, in ImageHelper, SOURCE% is a one-dimensional array, while in the manual it is a two-dimensional array. DRAWIMAGE doesn't care how the numbers are stored, just so it can read them in the correct order.

More Than A Pretty Face

DRAWIMAGE needs five other numbers beside the ones that constitute the image. The manual recommends that they be stored in the variables ROWBYTES, XSKIP, YSKIP, DWIDTH, and DHEIGHT. The complete statement with all its arguments is:

```
PERFORM DRAWIMAGE
(@SOURCE%(1),%ROWBYTES,
%XSKIP,%YSKIP,%DWIDTH,
%DHEIGHT).
```

The image is displayed on the screen as a rectangle that contains whatever dots of pencolor and fillcolor are specified by the source block. As it says on P. 317, ROWBYTES is the length of the source block's rows, in bytes. XSKIP and YSKIP aren't

used for most DRAWIMAGE purposes, DWIDTH and DHEIGHT are the width and height of the rectangle in bits and lines.

As we said before, the size and proportions of the source block represented by lines 580 to 730 can be anything that will fit on your screen. All you have to do is replace those lines in the program with more or fewer lines of the desired length. Just make sure that:

- The number of zeros and ones in each line is an even multiple of eight, and that:
- You change Linebits (modify line 1030) to show the new number of bits in each line, and:
- You change Linenumber (modify line 1040) to show the new number of lines in the source block.

Saving Your Image

Subroutine 2000 saves your integer array, under whatever name you want to give it, in a data file subdirectory called IMAGES/. ImageHelper assumes that you have two disc drives, .D1 and .D2. If you have more or less than two drives, modify all lines in the program in which ".D2" appears. Just erase ".D2" or change it to some other drive number.

Let's say you have designed, named and filed three images. They will have been preserved as a file that might have the following names: IMAGE/SKULL, IMAGE/DEATHS-HEAD, IMAGE/JOLLY.ROGER. Skull, deathshhead, and jolly.roger are the names you gave the images. Now you're all set to invent an Apple /// game about pirates!

The Apple /// DRAWIMAGE procedure stores an image as a 2-D array of bits. IMAGEHELPER calls this array a "bitblock" and allows you to draw it right on the screen. The program then converts your bitblock into a list of hex numbers, each of which contains two bytes of the image, and stores them in an array called SOURCE%.

The bitblock is represented in the IMAGEHELPER program listing by lines 580 to 730. These lines now contain a picture of a skull. To draw your own image, simply replace the lines of "0"s and "1"s with your own image. "1" stands for a dot of pencolor, "0" is a dot of fillcolor. This process can be speeded by sketching your image on the CRT screen with a wipe-off marker.

You can make a bitblock with a different size and/or shape by adding or subtracting lines or by changing the length of the lines, but the number of bits in each line must be an even multiple of eight, eg., 16, 32, 48, etc. You also have to change the variables Linebits and Linenumber to reflect the changes you have made. These variables are found in lines 1030 and 1040.

By altering lines 1820 and 1840 you can change SOURCE% to any name you like, such as Skull% or Deathshhead%.

This program also gives you the option of saving your image in a data file, where it will remain available for use in some other program such as a game program.

LIST OF VARIABLES

Bitbloc\$ — 2-D array for storing image
 Bytepairs — Number of bytes in image/2
 Linebits — Number of bits in each line
 Linenumber — Number of lines in image
 Hexdigit\$ — For temporary hex digit storage
 Hexnumlist\$ — array for storing hex numbers as they are produced by the program
 Refflist\$ — Array for binary-to-hex reference
 Bitbloclin, Halfbyte, Refflistin, Count, Counter, LoopA, LoopB — loop names
 ROWBYTES, XSKIP, YSKIP, DWIDTH, DHEIGHT, SOURCE% — Variable names used in Apple Business Basic Manual, Pp. 316-317. We use them in this program for clarity.

```
10 REM *****
15 REM * IMAGEHELPER - Rick Gauger*
20 REM * This program helps you *
25 REM * quickly create images *
30 REM * using the DRAWIMAGE *
35 REM * procedure *
40 REM *****
50 REM
500 GOSUB 920:REM Display instructions.
510 GOSUB 1020:REM Initialize variables.
520 GOSUB 1110:REM Place bitbloc into its a
    rray.
530 GOSUB 1190:REM Fill bin-to-hex referenc
    e list.
540 GOSUB 1300:REM Convert Bitbloc$ to hexn
    umbers.
550 GOSUB 1630:REM Display image
552 GOSUB 2000:REM Save image in file.
560 END
570 REM**** THIS DATA GOES IN BITBLOC$ **
580 DATA "000000000000011100000000000000"
590 DATA "00000000011111111111111100000000"
600 DATA "000001111111111111111111110000"
610 DATA "000011111111111111111111111000"
620 DATA "000011100001111111100000111000"
630 DATA "000011100000111111000001111000"
640 DATA "000011100000011110000001111000"
650 DATA "000001111111111101111111111000"
660 DATA "000001111111111000111111111000"
670 DATA "000000011111100000111111100000"
680 DATA "000000000111111111111110000000"
690 DATA "00000000010101010101010100000000"
700 DATA "00000000010101010101010100000000"
710 DATA "00000000001101010101010100000000"
720 DATA "00000000000111111111100000000000"
730 DATA "00000000000001111111000000000000"
740 REM**** THIS DATA GOES IN REFLIST$ **
750 DATA "0000", "0"
760 DATA "0001", "1"
770 DATA "0010", "2"
780 DATA "0011", "3"
790 DATA "0100", "4"
800 DATA "0101", "5"
810 DATA "0110", "6"
820 DATA "0111", "7"
830 DATA "1000", "8"
```

You'll notice that when the IMAGE/ file is opened, it is given the number #2. This is because, while you're running this program, file number #1 has already been given to the .GRAFIX driver.

Another noteworthy point is that those five non-image numbers (ROW-BYTES, etc.) are stored as the last five records of the data file.

How To Display An Image File

Now suppose you have created and filed several image files and you want to display them. LOAD and RUN ImageFileLooker. This program is self-descriptive, but it expects to find a data file in Drive 2 under a directory named IMAGES/. It also expects

that the last five records in a file will be ROWBYTES, etc. If you already have some files of images, they probably aren't stored in this format, which means ImageFileLooker won't display them properly. The best thing to do in this case is to remake your images using ImageHelper.

Where Do You Go From Here?

Here are some speculations about what use might be made of various conceivable modifications of ImageHelper and ImageFileLooker.

ImageFileLooker could be modified to continuously pour image data into a running program, hundreds of images rushing across the screen in a very complex animation. It would be

fun, for example, to make a "talking head," with dozens of different expressions, that you could display along with printed dialogue. Or, the head could run on real time and be commanded from the keyboard as a sort of electronic ventriloquist's dummy.

ImageHelper could be modified so that it would work with NEWFONT, for the creation of new typefonts. One kind of font that is needed is one with letters an inch or more high, so that little children can learn to type on the computer.

Good luck to all you Apple /// programmers; may all your images be bright. 

```

840 DATA "1001","9"
850 DATA "1010","A"
860 DATA "1011","B"
870 DATA "1100","C"
880 DATA "1101","D"
890 DATA "1110","E"
900 DATA "1111","F"
910 REM
920 REM***** Display Instructions *****

930 LIST TO 210
980 PRINT (When finished, press a key)
990 WINDOW 1,1 TO 10,21:HOME
1000 GET Keypress:TEXT:HOME:RETURN
1010 REM
1020 REM***** INITIALIZE VARIABLES ****

1030 Linebits=32
1040 Linenumber=16
1050 Bytepairs=(Linebits*Linenumber)/16
1060 DIM Bitbloc$(Linenumber)
1070 DIM Replist$(2,16)
1080 DIM Hexnumlist$(Bytepairs)
1090 RETURN
1100 REM
1110 REM***** PLACE BITBLOC INTO ARRAY *
1120 REM This subroutine reads the image
1125 REM line by line and places
1130 REM them into Bitbloc$.
1140 FOR Loop=1 TO Linenumber
1150 READ Bitbloc$(Loop)
1160 NEXT Loop
1170 RETURN
1180 REM
1190 REM** FILL BIN-TO-HEX REFERENCE LIST *
1200 REM This subroutine takes the two
1205 REM items on each line from 750 to 900
1210 REM and places them into the two
1220 REM columns of the Replist$ array
1230 FOR LoopA=1 TO 16
1240 FOR LoopB=1 TO 2
1250 READ Replist$(LoopB,LoopA)
1260 NEXT LoopB
1270 NEXT LoopA
1280 RETURN
1290 REM
1300 REM* CONVERT BITBLOC$ TO HEXNUMBER$ *
1305 REM four bits in order of their
1310 REM This subroutine takes groups of

1320 REM appearance in Bitbloc$, compares
1330 REM them with Replist$ thereby finding
1340 REM each group's equivalent hex digit.
1345 REM The hex digit is stored in
1350 REM Hexdigit$. When Hexdigit$ has
1360 REM accumulated four hex digits,
1365 REM subroutine 1490 stores its
1370 REM contents in Hexnumarray$.
1372 HOME:PRINT"Your image is represented by the ";
1374 PRINT"following list of hex numbers:"
1380 FOR Bitbloclin=1 TO Linenumber
1390 FOR Halfbyte=1 TO Linebits STEP 4
1400 Count=Count+1
1410 FOR Replistlin=1 TO 16
1420 IF MID$(Bitbloc$(Bitbloclin),Halfbyte,4)=Replist$(1,Replistlin) THEN Hexdigit$=Hexdigit$+Replist$(2,Replistlin)
NEXT Replistlin
IF Count=4 THEN GOSUB 1490
NEXT Halfbyte
NEXT Bitbloclin
RETURN
1430 REM
1440 REM* COLLECT & DISPLAY HEXNUMBERS *
1450 REM This subroutine collects each four
1460 REM hex digits and stores them, as a
1470 REM series of single four-digit hex
1480 REM numbers, in Hexnumlist$. Each hex
1490 REM number contains a pair of bytes.
1500 REM They are shown in Hexnumlist$ in
1510 REM same order they appear in image.
1520 Counter=Counter+1
1530 Hexnumlist$(Counter)=Hexdigit$
1540 Count=0:Hexdigit$=""
1550 PRINT"Hexnumlist$(;Counter;)"=;
1560 PRINT Hexnumlist$(Counter)
1570 RETURN
1580 REM***** DISPLAY IMAGE *****
1590 REM This subroutine loads the series
1600 REM of hex numbers in Hexnumlist$ into
1610 REM another array called SOURCE%.
1620 REM SOURCE% becomes the name of the
1630 REM place in memory where DRAWIMAGE
1640 REM can find the bit pattern it uses
1650 REM to draw your image. The binary
1660 REM data in the image had to be
1670 REM loaded into SOURCE% in pairs of
1680 REM bytes because that is how integer
1684 REM variables like SOURCE% hold data.
1686

```

```

1710 PRINT"the upper left corner of the ima
ge block"
1720 PRINT"to appear on the screen:";PRINT
1730 INPUT"Enter X (horizontal) coordinate:
";X
1740 INPUT"Enter Y (vertical) coordinate: "
";Y
1750 PRINT:PRINT"When you have finished loo
king at"
1760 PRINT"your image display, press any ke
y to"
1770 PRINT"terminate the program."
1772 PRINT"(Press key to continue)";GET Key
press
1780 ROWBYTES=(Linebits/8)
1790 XSKIP=0;YSKIP=0
1800 DWIDTH=Linebits
1810 DHEIGHT=Linenumber
1820 DIM SOURCE%(Bytepairs)
1830 FOR Loop=1 TO Bytepairs
1840 SOURCE%(Loop)=TEN(Hexnumlist$(Loop))
1850 NEXT Loop
1858 ON ERR INVOKE " D1/BGRAE INV"
1860 PERFORM GRAFIXON:OFF ERR
1870 PERFORM MOVETO(%X,%Y)
1880 PERFORM DRAWIMAGE(@SOURCE%(1),%ROWBYTE
S,%XSKIP,%YSKIP,%DWIDTH,%DHEIGHT)
1890 GET Keypress:TEXT
1900 RETURN
2000 REM*** SAVE IMAGE IN FILE *****

2010 PRINT:PRINT:PRINT
2020 GOSUB 2050:REM Create & open IMAGES/ f
ile.
2030 GOSUB 2210:REM Move image from SOURCE%
into file.
2040 END
2050 REM** CREATE & OPEN IMAGES/ FILE **
2060 PRINT"Do you want to save this image i
n a file?"
2070 PRINT"Enter Y (yes) or N (no).";GET Ke
ypress$
2080 IF Keypress$="Y" OR Keypress$="y" THEN
GOTO 2110
2090 PRINT"Program terminated.":END
2100 PRINT
2110 PRINT
2120 PRINT"Your image will be stored, under
the name"
2130 PRINT"you give it, in a file whose pat
hname is"
2140 PRINT".D2/IMAGES/(name of image).";
PRINT:PRINT"What do you want to name y
our image?"
2150 INPUT Imagename$
2160 ON ERR GOTO 2180
2170 CREATE".D2/IMAGES", CATALOG
2180 OFF ERR:OPEN#2,".D2/IMAGES/"+Imagenam
e$
2190 RETURN
2200 REM
2210 REM FEED CONTENTS OF SOURCE% INTO FILE
2212 REM Line 2250 reflects the fact that
2213 REM the last five variables in the
2214 REM SOURCE% array aren't part of the
2216 REM image. Instead, they are four
2218 REM numbers that instruct DRAWIMAGE
2219 REM about the size & shape of image.
2220 FOR Loop=1 TO Bytepairs
2230 WRITE#2;SOURCE%(Loop)
2240 NEXT Loop
2250 WRITE#2;ROWBYTES,XSKIP,YSKIP,DWIDTH,DH
EIGHT
2260 PRINT:PRINT"Your image is now preserve
d in a data"
2270 PRINT"file called .D2/IMAGES/";Imagena
me$
2280 CLOSE#2:RETURN
2290 REM This is the bottom line.

```

```

10 REM*****
20 REM* ImageFileLooker - by Rick Gauger *
30 REM* A program that uses the DRAWIMAGE *
40 REM* procedure to display an image *
50 REM* been stored in a data file. *
60 REM*****
90 REM This program is designed to be used
in conjunction with another program call
ed IMAGEHELPER.
94 REM IMAGEHELPER stores an image in a dat
a file with the pathame .D2/IMAGE S/(nam
e of image). This program reads that fil
e & displays the image.
104 REM
130 REM LIST OF VARIABLES
150 REM Bytepairs: Number of pairs of bytes
in image.
170 REM Count: Loop name, goto loop counter
190 REM Imageholder%: An array for temporar
y storage of the file's image data.
230 REM Imagename$: Holds user's input of t
he name of the image.
270 REM SOURCE%, ROWBYTES, XSKIP, YSKIP, DW
IDTH, DHEIGHT: Variable names used in A
pple /// Business Basic manual, Pp. 315
-317, used in this program for clarity.
350 GOSUB 450:REM Initialize variables.
370 GOSUB 530:REM Read image file.
390 GOSUB 950:REM Move data to SOURCE%.
410 GOSUB 1270:REM Display Image.
430 CLOSE#2:END
450 REM***** INITIALIZE VARIABLES ****
470 DIM Imageholder%(1000)
490 Count=1
510 RETURN
530 REM***** READ IMAGE FILE *****
550 REM This subroutine discovers how man
y byte pairs are contained in the file
by loading the contents of the file in
to Imageholder%.
610 REM Each record in the file contains on
e byte pair, except the last five, whic
h contain ROWBYTES, XSKIP, YSKIP, DWIDT
H, and DHEIGHT.
670 REM When all the records in the file ha
ve been read, the statement ON EOF send
s the program to line 910, where Bytepa
irs is filled.
730 PRINT:PRINT:PRINT
750 PRINT"Enter the name of the image you"
770 PRINT"wish to see: ";
790 INPUT Imagename$
810 OPEN#2,".D2/IMAGES/"+Imagename$
830 ON EOF#2 GOTO 910
850 READ#2;Imageholder%(Count)
870 Count=Count+1
890 GOTO 850
910 OFF EOF#2:Bytepairs=Count-5
930 RETURN
950 REM***** MOVE DATA TO SOURCE% ****
970 REM This subroutine moves the image d
ata from Imageholder% to SOURCE%. The
contents of the last five variables of
Imageholder% are placed in ROWBYTES, XS
KIP, YSKIP, etc. (Cont'd on page 29)

```

Three Questions

PKaso/U Works with IDS

Dear Bob,

We received your excellent four color brochure today, breaking a too-long dry spell of any communications from *ON THREE*. It is my hope that this mailing will quell the many doubts about *ON THREE*'s continuing existence. It's certainly no secret that Three's have always felt lonelier than the Maytag repairman, and this fall back in the magazine's publication schedule has not helped. Possibly because my name is at the top of your hotline list, I have received 6 or 7 calls in the last month from very tentative and concerned *THREE* owners whose principal worry seems to be the continuing viability of the machine that Apple says they will no longer support. The other main concern is whether *ON THREE* magazine has been discontinued as has the Open Apple Gazette newsletter.

My recent visit to your offices was a great boost of confidence for me and I am grateful for the four hours 'overtime' you allowed me in your intensely busy schedule. We reviewed some very important philosophical questions as well as some critical technical points about the ///. Your profile of a *THREE* owner (a small business owner of 3.6 ///'s) was most revealing, and would be worth an article. Your products are outstanding in timeliness and quality. I could not be without *ON THREE O'Clock*, (an absolute *must* for any Applications user), and *ONTIME*. Being one who assertively upgrades his system, I make great use of the *.MOUSE* (driver), *Draw ON ///*, and the *Trustor 30*, the *34 MegaByte* hard disk. *Lazarus* under *Catalyst* bombs out, probably because of the 256K memory in my machine. I am anxiously awaiting the 512K update ordered and *Selector ///*.

For those interested, the *Pkaso* revision 3 dated May 1982 of the ID12 color printer interface for the IDS Color Prism is the only one there will be. The *Pkaso/U* will not work with it. Also *Quark's Discourse* spooler will not work with any other card than the Apple UPIC.

We're all hoping to see some system bottlenecks relieved under your careful guidance, with the new products you intend to produce. If you don't mind a suggestion, the publisher needs a managing editor for *ON THREE* Magazine, paying particular attention to an established output schedule, and someone who will keep right on top of the subscription roster. You simply cannot do everything well without burning yourself out. As I asked, what can I do to further help you to help all 100 thousand of us?

Coville Woodburn
Grantham, NH

Thanks so much for your valued support. As you can see, we have made a number of changes at ON THREE. We have added two new people, to both help get us on a monthly publishing schedule and to develop the state of the art software that is so needed.

Regarding the new PKASO/U card not working with the IDS Color Prism: Since the old PKASO card is no longer made, there could be a problem for those of you needing color printouts. However, I have been assured by Interactive Structures (maker of the PKASO/U card) that you can use the new card with that printer. You simply have to change out the ROM chip on the card. Interactive Structures will do this for you.

DRAW ON Printer Update

Dear Sirs:

After working with *Draw ON ///* for two months, I find the program terrific, especially the font and shape utilities. I have a question. In your article on page 4 of the last issue and on the order form, you state that there is a version of *Draw ON ///* that can be used with an Apple DMP (or equivalent) printer and a UPIC interface card. I have a NEC 8025A printer with a UPIC card and purchased one of your first *Draw ON ///* programs. The program is currently set up for a PKASO interface card. How can I get the version necessary for my equipment? Also, is there any difference between the PKASO and the PKASO/U version?

I look forward to your reply, Thanks for keeping the Apple /// alive.

L.J. Guthrie
New Orleans, LA

We do now have a version of Draw ON for the Apple DMP-Imagewriter-ProWriter and compatible printers. If you send us your original Draw ON Start-up and Program disk along with a check for \$15, we will send you the version that will work with your hardware. As long as your NEC supports graphics there should be no problems.

Regarding the differences between the PKASO and PKASO/U version of the program, there is a subtle, yet major change. The program now supports all of the PKASO/U features, such as the ability to print just a portion of your drawing screen. It also allows you to blow up or shrink down the size of that portion of your drawing when you go to print. You are no longer limited to set printout sizes.

Mouse Needs Card on ///

Dear Bob,

The last issue of *ON THREE* was great as usual, but too short. I was not surprised about Apple's decision to drop the ///. The obvious lack of interest has been apparent for a long time. I've been lucky to have had great support from my local dealer (The Computer Place, Baton Rouge) as they are Apple /// fans. They use a /// for their customer data base and in-store accounting, so they are very familiar with all the problems. They have sold a lot of ///'s and support them well, in fact they still recommend and sell the /// (mainly to

businesses). Any /// owners in my area who need help could contact them. They emphasize service and support and welcome calls from Apple owners even if they didn't purchase their equipment from them. Of course, *ON THREE* has been a tremendous help.

I also got my Gameport /// and it works well, but I had some difficulty finding the connector that goes from the board to the joystick. My dealer was able to get one, but he says that Apple is no longer supplying them (Apple][Game Port Adapter Kit # A2M0082). Perhaps you could get some and sell them with the Gameport ///. I like the board as I am able to run programs that wouldn't even start up with the regular Apple][emulation. I am particularly interested in Rocky's Boots, a great program to introduce children to logical thinking.

Now for a few questions.

- 1) To use the //e mouse with Draw ON, do you need an interface card (I'm out of slots)?
- 2) With Selector ///, can you load diskettes that are 'Catalyzed'? Also, can you load the regular Apple][emulation and the Gameport Apple][simultaneously?
- 3) Are you still planning a bulletin board? It would be much easier for your clients and you if we could call the bulletin board and get the latest info and also leave questions to be answered as well as place orders.
- 4) Will *ON THREE* magazine be published on a more regular basis in the future?
- 5) Is Quark still interested in the ///? The only thing I've seen from them lately is their hard disk which is marketed mainly for the][.

I would like to see some articles on how to work on your Apple ///. Especially since Apple is backing out, self-service may come to be a necessity. A list of books available for the /// would also be very helpful. Possibly you could offer these through *ON THREE*. I would also like to see more advertisements. This could provide a source of income for you as well as keep your readers aware of what products are available. I think advertisements should be only for Apple /// products. There are plenty of ads for other stuff in other magazines.

Congratulations on a great publication and a great service. Looking forward to the next (and more regular) issues. Good luck!

Louis Jeansonne
Baton Rouge, LA

Thanks much for your comments, I'm delighted to hear from you and others that there are a few good dealers left. If anyone out there has had a particularly good experience with a dealer, please let us all know.

The Apple][Game Port Adapter Kit only needs to be purchased if you want to attach an Apple //e joystick to the Game Port. There are still a number of regular old Apple // joysticks available that can be connected directly to the Game Port.

About the //e mouse, it does come with an interface card that you need to put in one of your Apple ///'s slots. With the introduction of the ONTIME Desktop Manager, the mouse becomes usable from all programs, thus it might be worthwhile to free up one of your used slots and replace it with a mouse card.

Our bulletin board should be up and running by the time you get this issue, call the main office for the correct modem number. In regards to the magazine, we have now resumed regular monthly publication, commencing with this issue. Val Golding, our new managing editor, is doing wonders and I am confident that we can publish on a monthly basis from

this point on.

Quark has continued to support their Apple /// products, it's just the majority of their market is in the Apple][and Mac right now. I wouldn't count them out yet, as I think they may still be working on products that will work on both the Apple][and ///.

I've taken note of your request for self-service articles and I will be working with Val to see just what we can put together in terms of how to fix your own Apple ///. If you, or anyone else for that matter, has information of this type, I encourage you to send it in for possible publication.

Also in connection with service, we have just recently received word of a new Apple /// repair service in Lubbock, TX. Contact Elias Bengoa at (806) 765-9039 for further information.

AppleWriter Hint

Dear *ON THREE*,

I was extremely upset when I heard Apple cancelled one of the best computers they have ever offered. I just wish they would reconsider, admit their marketing mistakes and start a whole new marketing strategy. I firmly believe the Apple /// would drive many of the competitors (including the one that I believe sells by its three letter name only) out of the market. The new marketing strategy should include packages like COBOL, FORTRAN, BASIC, C, Pascal, Apple-Writer/Speller, /// E-Z Pieces and all of the reference and technical manuals.

Aside from all of that, I would like to make a few comments and suggestions that I feel some of your readers would like to hear.

Now that the Apple /// has been discontinued, how will I be able to get updated versions of Apple programs like Pascal and Apple Writer? If it wasn't for your magazine, I would not have known of the new versions. The dealers in this area have the same answer as many of the dealers across the nation do on the Apple ///: "HUH! I didn't know there was an update for these programs." Also, what about the interleave and graphics update for the regular /// owners (these updates are included in the Apple /// Plus)?

Apple has yet to release information for controlling the Apple /// monitor. In the Apple][, you can get in and out of the monitor without rebooting. But, in the Apple ///, I cannot get out of the monitor without a cold-start. What is probably needed is a complete list of the monitor locations, what they are used for, and what can be changed where. Can you help?

Will Apple start releasing the source listings for the various assembly programs in Business Basic, Pascal, System Utilities and Apple][Emulation without charging us poor Apple /// users a small fortune? Most of us have a lot of time and money invested in what we consider the ultimate affordable Apple. With a few modifications to some of these programs, the real power of this machine would shine through.

I have a suggestion on entering control characters in the Apple Writer text without the sequence CONTROL-V, control character, CONTROL-V. Although I could not find it in the manual, you can use the OPEN-APPLE key. The sequence is to press the OPEN-APPLE key while you enter a control character like ESCAPE or a control sequence. This should save a lot of time.

Here is a list of Apple /// would be favorites:

1. FORTRAN for the ///
2. C for the ///
3. A Disassembler for the ///
4. A fast disk copy utility
5. A MONTHLY ON THREE magazine
6. ON THREE telecommunications

Art Kassel
Layton, UT

P.S. Are there any Utah Apple /// user groups?

Unfortunately, I don't believe that Apple will ever reintroduce the Apple ///. I do think they will continue expanding the Apple //e until it becomes something close to what the /// is. Regarding the software updates, we are currently working on software update mechanisms for as many software products as we can. I will keep everyone informed on this and hopefully in an issue or two we can have a software update system in place.

There is a way to get into the monitor from the Apple /// and go back into your program without rebooting. One of the undocumented calls to \$1900 page enables the RESET key so you can enter the monitor. We will shortly publish some technical information like this, so if you hold on a little longer you'll get just what you want!

Realistically, Apple will never release anything concerning the Apple ///. One will just have to try and work around the major inconveniences of some of these programs. We do have available the full ANSI 77 FORTRAN language for the Apple ///, and have planned a release on our super fast (45 seconds per disk) copy routine that will copy up to four disks at one time. We also will shortly make available our NEXUS communications package that includes Tektronics graphics terminal emulation.

Now that we are on a monthly basis, the only things left that you want are the C language and a disassembler for the ///. Both of these items are available from outside sources and in the future we will publish a listing of just who makes what. How about it readers, are there any Apple /// user groups in Utah?

Keeping Up to Date

Dear Mr. Consorti:

I've been waiting for three years for something like ON THREE! Keep up the good work, and I hope your prices can come down as your volume builds up. Maybe you or some of your readers can help me with these problems.

When I boot my Business Basic or AppleWriter /// disks, the date-time info. that appears on my screen has the right date and time, thanks to my ON THREE O'Clock! But the day of the week is one day behind! How can I fix this irritating problem? Also, is there a reserved variable in Basic that holds the day of the week?

Does anyone have a program that will let me use my Hayes Micromodem //e? I already have the device driver, but I'm too lazy to write a Basic program for it right now.

I'm a professional cartoonist, and I don't think anyone can draw worth a poop with a mouse or a joystick. What the Apple /// needs is a drawing pad, especially one that will work with Draw ON. I wonder if anyone has ever designed a card and a driver that will make the cheap and effective Koalapad work with the ///.

You can put me down as a consultant. I use AppleWriter ///, VisiCalc ///, and do a lot of Basic programming. People can call me between 10 AM and 10 PM.

Rick Gauger
Bellingham, WA
(206) 767-8371

As you have probably seen, our prices do come down with time. Our 512K Memory board price has dropped considerably since its release a year ago. This is due to the tremendous price decrease in the 256K RAM chips we use in the board. Regarding the wrong day of the week: The clock chip does have a day of the week register. The standard System Utilities program does not allow you to change it. However, there is a BASIC program that will let you. It is called TIMESET on the System Demonstration disk that came with your ///. If you can't find it, send me a blank disk and a few dollars for postage and I'll put a copy on it for you. By the way, Business Basic does not have a reserved variable for the day of the week, but you can calculate it from the date information that BASIC does maintain. Again, it's in that demonstration program.

If you have an .RS232 driver that supports the standard .RS232 driver calls (as defined in the Standard Device Driver's Manual), any communications program should work just fine.

We have a version of the Draw ON program that supports the Apple II Graphics Tablet. We have the complete .TABLET driver and it works very nicely. I don't know if you can interface the Koalapad to the ///, but I'll look into it. Thanks for the offer of becoming a consultant, we need the help of persons like you.

Visi Gone?

Dear ON THREE:

I am having trouble getting support for some software I recently obtained. Could you, your readers, or someone in your organization help me?

I have a copy of Advanced VisiCalc for the Apple ///. I asked Visi Corp for a backup copy, which they offer for \$20. They said they sold their interest in the software to Software Arts in Wellesley, MA. I contacted them in May of this year, but still no response. Do you know who offers support for the Visi Corp series of software now?

This is excellent software. Much of it is widely used by Apple /// owners. Many users of this software are in the business environment where backup copies are crucial. I think you would be doing your readers a great service if you revealed where to obtain support for Visi Corp software.

Jim Sprague
Ashland, OR

I have heard many stories like yours over the past few months. To the best of my knowledge, nobody is offering support for the Visi Corp line. ON THREE is working out a software update/backup service and will keep everyone informed as to when this is available.

Draw On Manual Correction

Dear ON THREE:

I previously wrote ON THREE concerning the installation of Draw ON /// under Catalyst using the QC10 hard disk. As I have not heard anything, I am writing again.

I have been very careful in checking and rechecking to make sure everything was per your instructions. I keep getting the message 'DRW3.DIR/DRW3DATA not found'. The Draw ON /// program works well when booted from its own diskette. I don't know what to do next. I wrote to Quark and they were not familiar with the program but would research it if I would send them a copy of the disk. I would prefer your advice.

Another question that I have wanted to ask for a long time and maybe you people will have an answer. It is, using Apple Writer /// with an Epson MX-100 printer, how do you get super script to work as Apple Writer /// uses the null to determine end of file. I have tried everything I could think of but never could get it to work. Do you have any answer?

When do you expect the next issue of ON THREE to be out? I like the letters and articles very much.

Thanks for the help. I'll be waiting for your reply.

John C. Laughlin
Honolulu, HI

I'm sorry about the delay in responding but I never received your original letter. There is a very simple fix to the Draw ON ///—Catalyst problem you're experiencing. There should have been an ERRATA sheet in the Draw ON box. There is an error in the manual relating to how to install Draw ON under Catalyst.

On the bottom of page 111 of the Draw ON /// Graphics Tool User's Guide, the last line reads:

**Program path: .PROFILE/CATALYST/DRAW.ON.3/
SYSTEM.STARTUP**

To work correctly under Catalyst, it should read:

**Program path: .PROFILE/CATALYST/
DRAW.ON.3,SYSTEM.STARTUP**

Note the change, the last slash became a comma. The error is in the strange way that Catalyst sets the prefix. I won't go into detail, but they simply don't set the SOS prefix as they should!

Regarding the Apple Writer null character problem, Sun Data has a product called Printer Driver /// that gives you the ability to send a null to the printer through Apple Writer. Depending on what interface card you are using it should work.

The Talk of the Town

Dear Bob,

You're a hard man to get to talk with! I've phoned a few times but always you are busy. You may remember me if I enclose a copy of a brochure on a business program we developed to run on the ///, only to be shot out of the saddle when Apple dropped the machine.

We are in the process of translating the program to the IBM PC which really shouldn't be too much of a problem since it is written in UCSD Pascal. However, we're now considering an alternative approach, which our computer dealer insists is "walking backwards". Our typical customer is a hospital, medical clinic or utility and frankly they couldn't care less which machine a program runs on as long as it does the job. Further, most of them would rather rent than buy. Result: We are considering a plan to offer a lease of hardware, software, and support on the Apple ///.

We have located a supply of new ///'s and checking with a couple of outfits specializing in ///'s, and feel we can get all the ///'s we want. We can charge a reasonable rent and with the costs of ///'s now, it would be a most profitable way to go. We don't plan to maintain a repair shop and will rely on an inventory of extra machines to swap in when needed. Oh yes, we realize Corvus is out of the picture (they no longer produce it for the ///) and are presently looking at the Space Coast ICE drives.

So, what's the problem, you may ask. My son-in-law who wrote this program tells me with 512K or more memory he can considerably speed up the program's execution. (Generally it's fast but there are a couple of functions that are slowed because of limited memory.) This is where you would come in. Guess I must ask a blunt question. Are the sales of your 512K upgrades satisfactory so that you will continue with them? Also, assuming this goes well for us and we end up acquiring some 128K machines. Can you upgrade them? If so, at what cost?

Any information you can give will be much appreciated. We may go ahead and use the 256K version, as it works very well, and save the upgrading for an option for which we could charge later on. On the other hand it makes more sense to put our best foot forward with a better and faster program at first.

We'd like to try a couple of updates to see if the increased speed would be worth additional costs and changing the program. What can you offer us in this respect? One last question, when will Selector /// be available? We have an application where it will work well and are anxiously awaiting it.

Burt Whitlock
Cody, WY

Unfortunately I'm very hard to get hold of because I'm so busy. I'd love to talk to people more but there just isn't enough time in the day. I'm glad to see that the /// is working so well in your vertical market. Regarding the 512K Memory Upgrade, I just don't know if it will help you increase the speed of the program.

Since your program is written in Pascal, it is limited to a 64K data space. On a 256K machine there is plenty of additional memory (depending on the program size and what segments are swapped in and out). However, it may be that your program is taking up a good amount of memory and you are currently having to swap program segments. If this is the case, the memory upgrade will help you.

I just want you to realize that to use the extra memory available on a 256K machine (or a 512K machine for that matter), you will have to handle direct memory access. This is a limit imposed by the fact that the Pascal system for the /// uses 2 byte pointers, thus a 64K limit on data size. If you choose to handle the extra memory of the machine, you will have direct access to approximately 128K of memory on a 256K machine, and about 384K of memory on a 512K machine.

We are selling the 512K Memory Upgrades at a good pace and will continue to do so in the future (next few years, at least!). If you can find some 128K machines, we can upgrade them to 256K or 512K. We charge \$50 plus return shipping for the upgrade service, in addition to the memory board cost of \$200 for the 256K board and \$449 for the 512K board.

Selector /// is now available. If you have a need for program switching, or a series of vertical market programs such as yours, Selector /// is the way to go. 

Titan /// Plus ///

don russell

If you haven't read Al Evans' article */// to the Max* in *ON THREE*, Vol. II, No. 1, I'll wait here while you do. For those who already have, you probably remember Al's statement at the end, something about: "...an Apple II running *inside* an Apple ///." Well, thanks to Titan Technologies, he's going to get his wish. Well, almost.

Titan has released their "*/// plus //*" board, a major step in the direction of having just *one* computer (a ///, of course). The object of their new product is to give you a 64K Apple II (that's right folks, a *language card*) in Emulation mode. There are three versions available, a 16K, a 64K and a 128K. Each uses the same 48K of the Apple /// RAM as does the regular Emulation mode. Note however the usability of the extra memory on the 128K version is limited to that of a RAM disk.

With the 16K version you get exactly what you would expect, all the same things you get on your Apple II with a language card, including Integer Basic if you boot with the System Master, and of course the Mini-Assembler. If you are interested in the Mini-Assembler and believe what you may have heard about "no Mini-Assembler on the II plus," then you have a pleasant surprise in store. Drop a note to me or to *ON THREE* and we will be happy to let you know how to access it. As you would hope, if you are a Pascal user at all, you can use Apple1 as your boot disk after you run the special Emulation disk supplied by Titan. Since you have a language card now, Pascal comes up running like a charm—in 40-columns. (Don't despair, it gets better.)

The 64K and 128K versions come with added RAM and slightly different Emulation disks. The 128K version allows the use of a RAM disk in Apple ///mode or in Emulation mode under ProDOS. If you have been paying

attention, you will notice this is the first mention of ProDOS, Apple's new disk operating system. If you don't have it for your II plus, it can come in quite handy. One of the tricks it can do is to create a single disk that will boot in Apple /// mode or Emulation mode, and it does it all automatically. You can share files and not even have to swap disks.

Since Titan sells the RAM disk software separately for \$49.95 and 64K chips have come down to under \$2 each, you can decide if you want to save a few dollars or buy a ready-to-run set-up. The card comes with sockets for additional RAM chips. I got the 16K version, paid \$30 for 16-64K RAM chips and \$50 for Titan software and ended up with a 128K /// plus II, the ProDOS RAM drive, the SOS RAM drive and saved about \$70 in the process.

By now you may think I never find anything wrong with a new product for the ///. Wrong again. After I booted the Titan version of the Emulation disk, I was having a ball trying all kinds of software requiring a 64K Apple II. Then I decided to send a listing to my printer. Perhaps I'm a little strange, but I have a serial printer connected directly to the RS232 port on the back of my Apple ///. I didn't see any reason to take up another slot, which is just as well, since they are all full now. Being a born optimist, I entered "PR#7" and pressed the return key. That was the last I saw of my cursor, which, along with Peter, was off to never-never land.

Suffice to say that the Emulation disk (and memory) had been changed to accommodate software needed by the Titan card. If you guessed they used slots five and seven, you're absolutely right. Oh well, no big deal. I called Titan, who told me they were aware of the problem and would send me the correct disk for serial printers.

Unfortunately, the disk I received after a three week wait didn't work either. To make a long story short, I waited over three months for Titan to fix the problem and their customer support people (whom I know now on a first name basis) never did get the corrected software to me.

With many thanks to Daryl Anderson of *DA Datasystems* in New York, I finally got the correct disk—from him. I booted it and the printer worked like a jewel. If you have a Titan card and have run into this problem, send me your Emulation disk and I will copy the correct software onto it. My address is:

Orion Software Systems
831 Cliff Drive, Suite C1
Santa Barbara, Ca 93109-2508

Now for the fun part. Since I had used the Apple II patch (see *ON THREE* Vol. II, No. 1) to add lower case in Emulation mode, I wanted to do the same thing with the Titan disk. While attempting this easy (or so I thought) task, the Patch program wouldn't verify that the Titan disk was an Emulation disk. Part of the Patch program accomplishes this by checking for the commands on the disk that actually set up Emulation mode. Since I could not get it to work, I booted with the Apple Emulation disk and using Don Worth's fine utility *Bag of Tricks*, I started to examine the Titan disk on a sector basis. In the end, I found you need to make a couple of changes to the *On Three* Patch program so that it will modify the Titan disk correctly. The "FUNCTION Emulation_disk_verified" now reads:

```
-----more stuff-----  
FOR INDEX:=0 TO 7 do  
  IF {CHK_BUF[INDEX]<>BLK  
    [383+INDEX]} THEN
```

Change the 383 shown above to 369. The code to set up Emulation

mode on the Titan disk is displaced by 14 bytes and this change will allow the Patch program to identify the disk as an Emulation disk so that it can proceed with the modification.

One more change and you can recompile the program. If you don't have the Pascal system you can get the revised program from *On Three* or send your disk to me at the address given a few paragraphs back. The second and final change is in the Patch_Character_Set program, which starts:

```
BEGIN(Of Patch_Character_Set)
  Get_Cset;
  -----more stuff-----
  MOVELEFT(C_SET, T_BUF[2694],
  SIZEOF(C_SET));
```

Change the 2694 shown above to 2794. Compile and run the program and you should now have lower case whenever you use Applesoft Basic on your 64K Apple II. By the way, you've only heard this a few hundred times, but once more never hurts: Be sure to use *backup* copies for the changes to the Titan disk.

For those of you who still use Integer Basic, there remains one more glitch. For reasons I do not understand, the configuration mode

can not be set to Integer. If you do, DOS will not boot.† However, you can get around this by setting the configuration to Applesoft and booting with the System Master. Integer Basic will then be loaded into the "language card" and the INT command will switch you to Integer with DOS up and running. Since with this method, Integer is loaded from the language card, you will not have lower case, but upon switching back to Applesoft with the FP command, everything works fine.

One more piece of good news. If you have an 80-column card in an Apple II, or want to buy one, it will work equally well if you place it in slot three of your Apple III. If you use Pascal, the boot disk will find the 80-column card, turn it on and come up running in 80-columns. I have a *Videx Videoterm* and it is about 1/16" too long to fit inside my III. I got an *Extend-a-slot* (two connectors and a cable) for the Apple II, plugged it into slot three of my Apple III and plugged the Videx into the outboard end. I haven't taken time yet to analyze the circuit to see if the softswitch would work, so I installed a slide switch (Radio Shack has all kinds) to go

back and forth between 40- and 80-columns. Upper and lower case is toggled with [Ctrl-A], and is explained in excellent detail in the Videx manual.

A couple of additional details before I go. There is a game paddle port on the III plus // board so all your Apple II games work fine. In addition, if you use Applesoft programs a lot, Titan has a program called "Ember" that gives you access to about 150K in your 128K III plus // board. Neat, huh?

All in all, I'm very pleased with the III plus // and am looking forward to the announcement of the III plus //e board, which should be before year's end. Then I'll be able to run Applesoft on my III and really thumb my nose at big blue. Now if only the shift key would work in 80-column mode, and maybe double hi-res graphics. Oh well, perhaps someday...

†We suspect perhaps the Emulation routines expect to find Applesoft the "native state" in Emulation mode as it would be in an Apple II plus. When booting on a II plus, you will first come up in Applesoft and then may switch to Integer. This holds true even when the Hello program is written in Integer. ...ed 

COVER ALL BASES



Available from:

CompuCraft
P.O. Box 3155
Englewood, CO 80122
303-850-7472

With

POST MASTER
For All of Your Mail List Needs

Large Capacity — 32,000 Entries
Word Processor Interface
Dials Phone Numbers
Sophisticated Duplicate Checking
Bulk Mail Features
Flexible Data Selection
and
FAST!!!

Price: \$200

Requires 256K and Profile or 2nd Disk Drive

10 Day Money Back Guarantee!

Technique:

Sort of a Heap

brenda shaw

Sorting is a very important part of all types of programs. Whether you are working in accounting, education, or farming, sooner or later you will want to sort a list of names, numbers, amounts or places. A sort can come in many shapes and sizes, and sometimes it is hard to decide which one is best for your program application. A lot depends on the key, or element to be sorted on, and the number of records to be sorted. Generally, a bubble sort can be used if the number of records is small (15 or less). If the list is long but the key is short, a radix sort is a possibility. The radix sorts numbers one digit at a time and works quite well in appropriate programs. With long keys and a long list, the Heap sort, merge sort or quick sort could work for you.

For a long time I used the Bubble Sort for everything. I assumed that since it was the first sort routine introduced in most computer classes, it must be the best. I discovered the error of my ways when I tried to run a year end accounting report. It took over 30 minutes to get through the sort alone! Since staring at my monitor for a half hour isn't my idea of fun, I decided to try the Heap sort as my next step. It is a little more involved than the Bubble Sort, but it is simpler than many others and it doesn't

require any extra arrays in memory to store pointers or other information.

The Program and Figures

Before I explain the inner workings of a Heap sort, I would like to outline the program and explain the figures. That way, those of you who are content to know that it does work, and don't thrive on the "how and why", can try it out.

The program is split in four main parts, two of which comprise the Heap sort, and two of which could be considered extra. Part I (lines 110-155) generates a list of 25, 2-digit random numbers, stores them in the array NUMBER(i), and prints them out as they appear in the beginning array. Part IV (lines 350-405) is also not a direct part of the Heap sort. It prints out the same beginning list as Part I and then proceeds to sort it using a Bubble Sort. This was included for comparison purposes only.

Part II (lines 165-235) is the first part of the two-part Heap sort. This section isn't actually sorting, it is taking the original 25 numbers and rearranging them to create a Heap. figures 1.a through 1.e show the step by step creation of a Heap using 5 numbers. The original five numbers were already sorted, but they are treated like any unsorted list. Num-

bers are inserted from the top down and from left to right. So the first number is at the top of the Heap and the second number is inserted in the next level down, starting at the left. The second level must be full before filling the third level, etc. Notice that once the numbers are arranged in a Heap, the largest number in the list is at the top (or root) of the Heap. Lines 215-235 print out the numbers in their present 'Heaped' order. Part III (lines 245-335) is the actual sorting procedure. Figures 2.a thru 2.c illustrate the sorting steps. It sorts in a somewhat backwards manner. The first number that is put in its final position is the largest number—found at the top of the Heap. Continuing, it sorts to find the next largest number. Try the program if you want to see some actual number crunching and then read on to discover how it works.

What is a Heap?

The term, Heap, refers to a particular kind of binary tree that satisfies the following rule: $\text{NUMBER}(j) \leq \text{NUMBER}(i)$ for $2 \leq i \leq n$ and $i = \text{INT}(j/2)$. In the program $n=25$, while in the figures $n=5$. Study figure 1.e to see how it conforms to this rule. The program creates the Heap one number at a time. It starts with the first number, NUMBER(1), which is considered a Heap by itself, and then inserts the second number, NUMBER(2). These two numbers are then rearranged such that the final structure is a Heap. Then, another number is inserted and these three numbers are structured into a Heap. Insertions are performed repeatedly until all numbers in the original list form a Heap (figure 1.e).

The actual sorting process depends on the fact that the root, or top number in a Heap will always be the largest number in the list. In our program we are sorting an array of $n=25$ NUMBERS, and after Part II is

SAMPLE OUTPUT

```
HEAP SORT
Begin: 20 10 29 49 7 44 35 20 3 12 13 49 20 15 22 19 5 47 8 1 18 24 31 40 50
Heap:  50 47 49 29 31 49 35 19 20 13 24 44 20 15 22 10 5 3 8 1 7 12 18 20 40
Number of comparisons to create HEAP: 45

Final: 1 3 5 7 8 10 12 13 15 18 19 20 20 20 22 24 29 31 35 40 44 47 49 49 50
Number of comparisons in HEAP sort: 110

BUBBLE SORT
Begin: 20 10 29 49 7 44 35 20 3 12 13 49 20 15 22 19 5 47 8 1 18 24 31 40 50
Final: 1 3 5 7 8 10 12 13 15 18 19 20 20 20 22 24 29 31 35 40 44 47 49 49 50
Number of comparisons in BUBBLE sort: 300
```

completed, the largest number in our list is now in the first position, NUMBER(1) (figure 2.a). The sorting procedure then swaps the first and last number in the list; in our case, NUMBER(1) and NUMBER(n) (figure 2.b). Then NUMBER(n) is disregarded in the remainder of the program because it is in its final position—it is the largest number and it is in the 25th position. The next step is to use NUMBER(1) thru NUMBER(n-1) and rearrange them to remake a Heap (figure 2.c). The result of this is that the second largest number will end up in

NUMBER(1). Next, NUMBER(1) and NUMBER(n-1) are swapped and now NUMBER(n-1) is forgotten because it is in its correct place. Using NUMBER(1) thru NUMBER(n-2) another Heap is made and the process continues. By repeating this exchange and reconstruction process, the initial list is sorted into ascending order.

A Few Comparisons

A common way to determine or compare the efficiency of a sort is to look at the number of comparisons between elements needed to produce

a sorted list. In general, the Heap sort requires $n \cdot \log(\text{base } 2)n$. In our case, n equals 25 and the sort should require at least 100 but less than 125 comparisons. On the other hand, the Bubble sort usually requires $[n(n-1)]/2$ comparisons. For our list, that would mean 300 comparisons. Since each comparison takes time, the Heap should, in most cases, be faster. It is not always faster because if the list is originally in order the Bubble only requires one pass where as the Heap will go through every step in the procedure.

```

1  REM *****
2  REM * Heap Sort Program for the /// *
3  REM * by Brenda Shaw (c) 1985 On Three *
4  REM * Program is in four main parts: *
5  REM * Part I: Generate a list of *
6  REM * 25 random integers *
7  REM * Part II: Rearrange the *
8  REM * integers into a 'heap' *
9  REM * Part III: Sort the 'heaped' *
10 REM * numbers into ascending order *
11 REM * Part IV: A bubble sort for *
12 REM * comparison purposes *
13 REM *****
14 REM -----Part I-----*
15 comp1=0:comp2=0:DIM number(25),samenum(
16 25)
17 FOR i=1 TO 25
18   number(i)=INT(RND(1)*51):samenum(i)=n
19   umber(i)
20 NEXT i
21 last=i-1
22 REM * Print out the beginning list of
23   random, unsorted numbers *
24 HOME:VPOS=2:HPOS=36:PRINT"HEAP SORT":VP
25 OS=4:PRINT"Begin: ";
26 FOR i=1 TO last
27   PRINT number(i); " ";
28 NEXT i
29 PRINT:PRINT
30 REM -----Part II-----*
31 FOR q=2 TO last
32   temp1=q:temp2=number(q):temp3=number(
33   q)
34   IF temp1<=1 GOTO 200
35   j=INT(temp1/2)
36   comp1=comp1+1:REM Not
37   necessary, only for demonstration
38   IF temp3<=number(j) GOTO 200
39   number(temp1)=number(j):temp1=j
40   GOTO 175
41   number(temp1)=temp2
42 NEXT q
43 REM * Print out the list in current
44   order - they are now in a 'heap' *
45 PRINT"Heap: ";
46 FOR i=1 TO last
47   PRINT number(i); " ";
48 NEXT i
49 PRINT:PRINT:PRINT"Number of comparisons
50   to create HEAP: ";comp1:comp1=0:PRINT:
51 PRINT
52 REM-----Part III-----*
53 FOR q=last TO 2 STEP-1
54   SWAP number(1),number(q)
55   i=1:temp4=number(1):temp5=number(1)
56   j=2
57   comp1=comp1+1:REM Not
58   necessary, only for demonstration
59   IF j>(q-1) GOTO 300
60   j2=i+1
61   comp1=comp1+1:REM Not necessary,
62   only for demonstration
63   IF j2=q OR number(j2)<=number(j) GOT
64   0 285
65   j=j+1
66   comp1=comp1+1:REM Not
67   necessary, only for demonstration
68   IF number(j)<=temp5 GOTO 300
69   number(i)=number(j):i=j:j=2*i
70   GOTO 265
71   number(i)=temp4
72 NEXT q
73 REM * Print out the final list - sorted
74   in ascending order *
75 PRINT"Final: ";
76 FOR i=1 TO last
77   PRINT number(i); " ";
78 NEXT i
79 PRINT:PRINT:PRINT"Number of comparisons
80   in HEAP sort: ";comp1:FOR i=1 TO 3:PRI
81   NT:NEXT i
82 REM-----Part IV-----
83 REM * Print out the beginning list of
84   random, unsorted numbers *
85 HPOS=34:PRINT"BUBBLE SORT":PRINT:PRINT"
86   Begin: ";
87 FOR i=1 TO last:PRINT samenum(i); " ";:N
88 EXT i:PRINT:PRINT
89 REM * Sort the numbers into ascending
90   order *
91 FOR i=1 TO last-1
92   flag$="no"
93   FOR j=i+1 TO last
94     comp2=comp2+1:REM Not necessary,
95     only for demsonstration
96     IF samenum(j)<=samenum(i) THEN SWAP
97     samenum(j),samenum(i):flag$="yes"
98     NEXT j
99   IF flag$="no" GOTO 395
100  NEXT i
101 REM * Print out the now sorted numbers
102   and the number of comparisons *
103 PRINT"Final: ";
104 FOR i=1 TO last:PRINT samenum(i); " ";:N
105 EXT i:PRINT:PRINT
106 PRINT"Number of comparisons in BUBBLE s
107 ort: ";comp2
108 END

```

over the years. With Val as managing editor, I'm sure we will see a good mixture of feature articles, reviews and tutorials—all designed to help you get the most out of your machine.

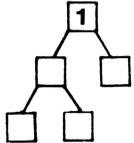
We have big plans for the ///, and I promise that 1986 will be a very interesting year!

DE-CLASSIFIEDS

Classified rates: \$1 per word, \$25 minimum. Copy must reach us 60 days prior to cover date, e.g., January 1st for March issue, which would be mailed February 1st.

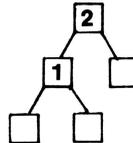
256K Apple /// System for sale complete with: second Disk ///, ProFile 5 MegaByte Hard Disk, Serial Card ///, VisiCalc ///, Business Basic, Pascal 1.1 with Tech Manual, SOS Manuals, SOS Device Drivers Writers Guide, Record Processing Services, AppleCare Service Plan to 6/1/86. Asking \$2500. Tom O'Conner, 6501 Arnold Road, Raleigh, NC 27607 or phone (919) 851-6471.

**ON THREE is
Your Apple ///
Support Group**



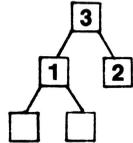
1,2,3,4,5

fig. 1.a



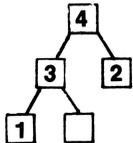
2,1,3,4,5

fig. 1.b



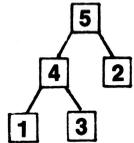
3,1,2,4,5

fig. 1.c



4,3,2,1,5

fig. 1.d

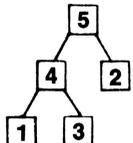


5,4,2,1,3

fig. 1.e

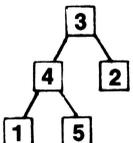
**Figure 1
Creating a Heap**

fig. 1.a — 1 is considered a heap by itself.
 fig. 1.b — 2 is inserted and 1 & 2 are arranged in a heap.
 fig. 1.c — 3 is inserted and 1,2,3 are in a heap.
 fig. 1.d — 4 is added and 1,2,3,4 are heaped.
 fig. 1.e — 5 is inserted and the entire list is arranged in a heap.



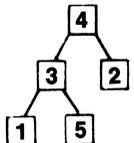
5,4,2,1,3

fig. 2.a



3,4,2,1,5

fig. 2.b



4,3,2,1,5

fig. 2.c

**Figure 2
Sorting a Heap**

fig. 2.a — The original heap—the largest number is at the top, or root, of the heap (5).
 fig. 2.b — NUMBER(1)=5 and NUMBER(n)=3 are swapped—the largest number (5) is now in its final place.
 fig. 2.c — 5 is now disregarded and a new heap is made with NUMBER(1) thru NUMBER(n-1)=4). Steps 2.a, 2.b, 2.c will be repeated until all 5 numbers are in ascending order.

ImageHelper (from page 20)

```

1050 DIM SOURCE%(Bytepairs)
1070 FOR Loop=1 TO Bytepairs
1090 SOURCE%(Loop)=Imageholder%(Loop)
1110 NEXT Loop
1130 ROWBYTES=Imageholder%(Bytepairs)
1150 XSKIP=Imageholder%(Bytepairs+1)
1170 YSKIP=Imageholder%(Bytepairs+2)
1190 DWIDTH=Imageholder%(Bytepairs+3)
1210 DHEIGHT=Imageholder%(Bytepairs+4)
1230 IMAGE
1250 RETURN
1270 REM***** DISPLAY IMAGE *****
1290 PRINT:PRINT
1310 PRINT"Enter the X or horizontal coordi
nates of
1330 PRINT"where you want the upper left co
rner of
1350 PRINT"the image block to appear on the
screen:"
1370 INPUT X
1390 PRINT"Now enter the Y or vertical coord
inates:"
1410 INPUT Y
1430 ON ERR INVOKE".D1/BGRAF.INV"
1450 PERFORM MOVETO(X,Y):OFF ERR
1470 PERFORM DRAWIMAGE(SOURCE%(1),%ROWBYTE
S,%XSKIP,%YSKIP,%DWIDTH,%DHEIGHT)
1490 PRINT:PRINT
1495 PRINT"You are about to view the image
which"
1496 PRINT"is stored in a data file called"
1497 PRINT".D2/IMAGES/";Imagename$
1498 PRINT
1510 PRINT"Press any key to view image. Wh
en you
1530 PRINT"have seen enough, press any key
to
1550 PRINT"end the program."
1570 GET Keypress$:PERFORM GRAFIXON
1590 GET Keypress$:TEXT:RETURN
1610 REM This is the last line.
    
```

Call Three: Hot Line/Apple /// User Groups

If you would like to get together with other Apple /// owners and exchange ideas, a user group is for you. Below is a listing of all Apple /// user groups known to us. If you have recently formed a group or know of one we have not listed here, please contact *On Three* and let us know so that they may be included. There is no charge for this service.

California

Sacramento Apple /// User Group
1433 Eldon Circle
Carmichael, CA 95608
(916) 482-6660

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

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

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

Apple /// Users of Northern California
220 Redwood Highway #184
Mill Valley, CA 94941
(415) 383-0203

Silicon Valley /// User group
707 Continental Circle
Mountain View, CA 94040
(415) 969-6093

*Psychologists on ///
West Covina/Glendale
Apple /// User Group*
413 W. Heatherlglen Lane
San Dimas, CA 91773
(818) 963-2980

Florida

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

Georgia

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

Illinois

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

Kansas

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

Maine

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

Maryland

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

Massachusetts

Applesauce
24 Dickinson St.
Amherst, MA 01002

Ohio

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

Texas

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

Virginia

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

Greater Tidewater

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

Washington

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

The *Call Three: Hot Line* is a service whereby Apple /// users with problems can call an area number to get assistance. The individuals answering the phones are fellow Apple /// users who have volunteered to help others over some of the rough spots. They are not compensated for this service, therefore we owe them a resounding "three cheers."

We would like to expand this service even further, so if you are familiar enough with your machine to be able to aid others and answer questions, please write us, stating your areas of expertise and availability in terms of days and hours. Certainly you can bask in the knowledge that you have been able to help a fellow Apple /// user.

For those of you who have questions, feel free to call our consultants listed below. **Please** observe however, the calling hours shown and before placing a call, double check the time zone so that you don't inadvertently wake someone up! There are no other restrictions on using the service other than as stated above. Again, **please** remember these people are volunteers, and if we receive information indicating that calling hours are not being observed, we will have no choice but to remove the consultant from the listing or, worse, discontinue the service.

The following is an alphabetical listing of subjects and abbreviations used in the "subjects" column of the consultants listing.

Subject	code	subject	code
Accounting	AC	Graphics	GR
Agriculture	AG	Micro-Sci	MI
Assembly	AL	Modems	MD
Lang.			
Business	BB	Pascal	PA
Basic			
Catalyst	CT	ProFile	PR
Cobol	CO	Quark	QU
CP/M	CP	SOS	SO
Data Base	DB	Spread- sheets	SS
		Telecom.	TC
Education	ED	Word Proc.	WP
Financial	FI	Emulation	AE
Fortran	FO	/// E-Z	EP
General	GE	Pieces	

Name	State	Telephone	Days	Hours	Zone	Subjects
Coville Woodburn	NH	(603) 863-5590	M,Tu,Th,F	7-8pm	Eastern	CT, QU
Ken Johnson	MA	(413) 253-3700	Su-Sa	6-9pm	Eastern	BB, PA, MD, WP, MI
Richard F. Malley	CT	(203) 232-9505	M,Tu,W,F	6-9pm	Eastern	GE, SO, WP, SS, QU, CT, PR
Harry T. Hanson, Ph.D.	NJ	(201) 467-0712	M-F	6-9pm	Eastern	CO, SS, PR, MD, CT
Edward N. Gooding, Sr.	VA	(804) 747-8751	Su-Sa	6-9pm	Eastern	CO, SS, PR, MD, CT
Al Johnson	FL	(904) 739-1042	M-F	9am-6pm	Eastern	GE
Paul Sanchez	FL	(305) 266-5965	Su-Sa	10am-4pm	Eastern	SS, PR, CT
John & Lisa Beckett	MO	(417) 678-2500	M-F	6-9pm	Central	GE
Jim Ferencak	IL	(312) 599-7505	M-F	10am-5pm	Central	GE, EP, DB
Neil Quellhorst	IL	(217) 434-8727	Su-Sa	7-9pm	Central	AL, BB, GR, PA, SO, TC
David B. Hays	KS	(316) 722-1242	M-F	7-11pm	Central	GE
Terri Wiles	CO	(303) 850-7472	Su-Sa	10am-6pm	Mountain	PA
Pat Holwagner	CA	(415) 433-2323	M-F	10am-6pm	Pacific	GE, SS, WP, CT, DB, SU, AE, EP
Vincent F. Latona	CA	(818) 703-0330	M-F	9am-5pm	Pacific	GE, WP, BB, SS, AE
Carl & Anita Reynolds	CA	(714) 734-9324	M,Tu,Fr	4pm-9pm	Pacific	GE
Wayne Hale	CA	(619) 450-3856	M-F	7-11am	Pacific	BB, GR, CT
Dennis R. Cohen	CA	(818) 956-8559	Su-Th	10am-10pm	Pacific	GE
Kelly C. McGrew	WA	(206) 943-8533	Su-Sa	6-10pm	Pacific	DB, GR, SS, PR, MD, CT

Word Juggler
(from page 12)

3. While the search and/or replace features are powerful and extremely rapid, I find it tedious to have to supply the search and replace strings each time this command is invoked. The ability to use the the same parameters would be a benefit. This could be accomplished by supplying a control key to the prompts for SEARCH FOR and REPLACE WITH to indicate the use of the last supplied strings. In addition, the ability to search for and/or replace the various printer enhancements is needed.
4. An 'over-print' command to allow one line to over-print another is needed.
5. A backspace command to allow the over-printing of a single character is needed.

6. A command to utilize the phantom rubout and phantom space characters that are on some print wheels (ie. trademark symbol, copyright, etc.) would also be appreciated.
7. Glossary features, footnoting and hyphenation help would be a nice addition as well.
8. It would be nice to have the ability to 'Print' to a disk file so that you could print it out later.

A version for the Apple //e, running under the new Apple ProDOS operating system, has also been released. It operates exactly like the version for the Apple /// and document files are easily used by either the /// or //e.

It is my opinion that Word Juggler is one of the finest word processors available on any microcomputer and the fact that it was implemented on the Apple /// leaves no doubt in my mind that the /// will be around for a long time, notwithstanding that it is no longer in production. 

CURSOR ///s for *Draw ON ///*

- LIMITED SUPPLY...ONLY 50 AVAILABLE!!! APPLE CURSOR /// JOYSTICKS FOR USE WITH THE NEW DRAW ON /// GRAPHICS PROGRAM.
- EACH CURSOR /// HAS BEEN TESTED FOR USE WITH DRAW ON ///. 
- INSTALLATION INSTRUCTIONS & SETUP PROCEDURES ARE INCLUDED.
- SEND CHECK OR M.O. FOR \$42.95 (CALIF. RESIDENTS ADD 6% SALES TAX) + \$2.50 S&H WITH NAME, ADDRESS & ZIP TO: CREIGHTON HOLTE
3557 KEATING ST., SAN DIEGO, CA 92110

Apple Cursor /// is a registered trademark of Apple Computer, Inc.



Cap'n Magneto

- The only REAL Apple /// game
- Interactive Animated Fantasy Adventure
- 256K Apple /// with second drive or Profile
- Compatible with Quark's Catalyst

Only \$40 Postpaid from
PowerTools, 1206 Karen Avenue,
Austin, TX 78757

Cap'n Magneto

Tricky Quickle:

Flasher

sharon webb

Do you ever feel nostalgic for the Apple II? Sure you love your Apple ///, but your old buddy had some features that Apple /// doesn't have—like the FLASH command.

Well, your Apple /// can be flashy too, but with a difference: With this Business Basic subroutine you can simulate the Apple II FLASH command. You can also vary the number of flashes from molasses slow to lightning fast by altering one line, and if you're into mind control, you can even print subliminal messages on the screen.

Lines 200 to 270 contain the subroutine. Line 210 controls the speed. Change it to read : FOR SLOW = 1 TO 3, then RUN. If the result looks familiar, it is. That's the speed of the Applesoft FLASH command. FOR SLOW = 1 TO 1 gives the subliminal effect.

Store the statements to be flashed in FLASH\$ as in the example in line 400, then call the routine. Line 230 designates the screen position of the flashing line. It may be used either where it is or outside the subroutine. You might like to experiment with changing the vertical position in a loop for interesting effects.

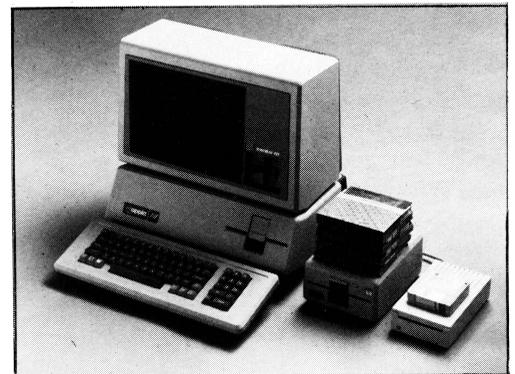
```
10 REM *****
20 REM *   FLASHER   * by Sharon Webb *
30 REM *****
100 GOTO 400
200 FOR FLASH=1 TO 20
210   FOR SLOW=1 TO 10
220     IF FLASH/2=INT(FLASH/2) THEN INVERSE
       :ELSE NORMAL
230     VPOS=10:HPOS=28
240     PRINT FLASH$
250     NEXT SLOW
260   NEXT FLASH
270   NORMAL:RETURN
400   HOME:FLASH$="Apple /// can flash too!"
410   GOSUB 200
```

Now! 800K on a Micro-floppy!

- UniDisk™ ///.5
and interface card\$499
plus \$10 shipping and handling
Including Apple /// driver and documentation

Diskettes fully transportable to UniDisk
equipped Apple //e, //c; drive and
interface can plug into //e.

ON THREE
P.O. Box 3825
Ventura, CA 93006
(805) 644-3514
VISA, MASTERCARD,
AMERICAN EXPRESS*
*3% surcharge on AE



**Would you back-up
your Profile
with 36 standard floppies
or just 7 micro-floppies?**
(consider the cost savings, too)

Unidisk is a registered trademark of Apple Computer, Inc.

ON THREE Presents . . .

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

Features such as rubber banding of lines, user adjustable grids, built-in help screens and easy to follow menus make **Draw ON III** the ONLY graphics package for the **Apple III** that is both powerful and easy to use. Combined with an excellent (**Apple** styling) instruction and tutorial manual, you can be doing useful work in less than an hour. The only limit as to what you can do with **Draw ON** is your imagination.

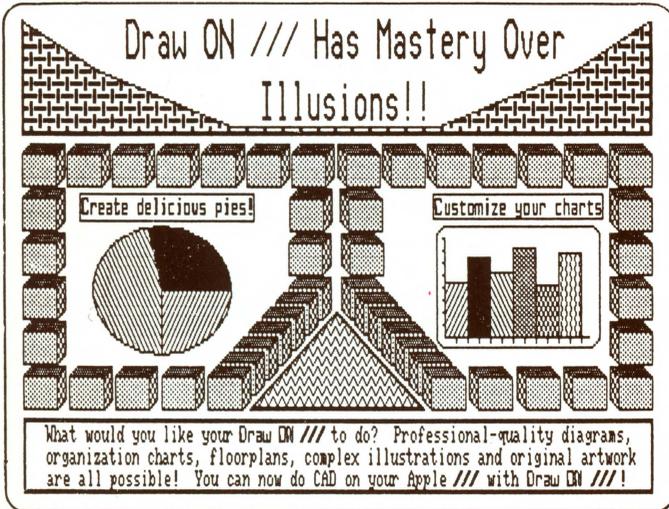
Draw ON gives an individual the power of a graphic arts studio. Use it in creating charts, preparation of slides and tables for presentation, and letterhead design. With **Draw ON** you can make changes to the dull graphs

Draw ON III Graphics Tool



that your other programs create by adding borders, textures and different typefaces. Even **Computer Aided Design** (CAD) applications such as circuit layouts, drafting and flowcharting are now possible on your **Apple III** with **Draw ON III**.

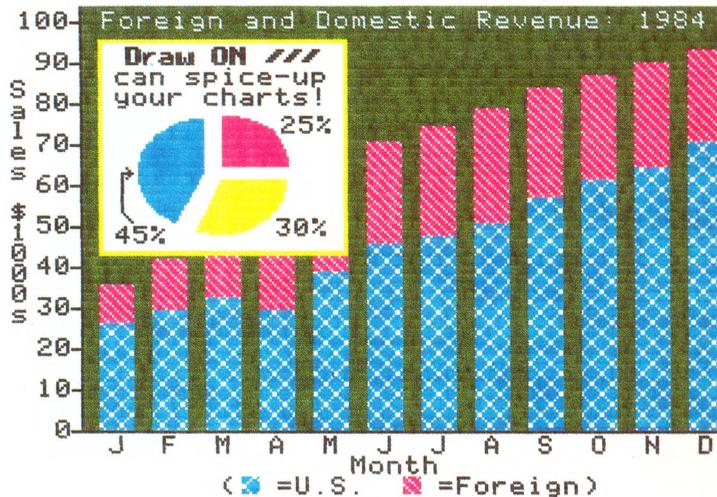
Draw ON combines powerful cut and paste facilities with the ability to mix text (in a variety of sizes and styles) with your drawings. If you don't like any of the text fonts or objects that come with **Draw ON** you can design your own! You can label your drawings with these fonts or even use them in your other programs. You can pick up objects and expand, shrink, rotate, invert



and texture them. You can also zoom in on a particular portion of the screen to do detailed work.

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

After creating your chart, table or other piece of art you will probably want to print it out. If you have an **Apple DMP** (or **C. Itoh Prowriter**, **Imagewriter** or **Epson (MX, RX, or FX)** printer, **Draw ON** can print out your drawings directly. For those of you who don't have these more popular printers, **Draw ON** also works with all of the printers the **PKASO** and **PKASO/U** interface card support. This includes **Centronics**, **Epson**, **NEC**, **Okidata**, the **IDS Prism** and **IDS Color Prism**. To print out color drawings you will need the **IDS Color Prism** printer and the **PKASO** interface card.



Optional Equipment

Cursor III Joystick, **Apple Mouse IIe** or **Apple II Graphics Tablet** (**Graphics Tablet** version costs \$50 extra), **RGB Color Monitor**, **Dot Matrix Printer**.

A **PKASO** or **PKASO/U** interface card is needed if you don't have an **Apple DMP** (or **C. Itoh Prowriter**), **Imagewriter** or **Epson (MX, RX or FX)** printer. You must specify the printer and interface card you are using before ordering.

Draw ON requires an **Apple III** or **Apple III Plus** with a minimum of 256K and is available for only \$179 + \$5 for shipping and handling. **Draw ON III** is not copy-protected and may be installed under **Selector III** and **Catalyst**.

The Apple III Fruit Machine

For everyone who doesn't have the time or money to hop a plane to **Las Vegas**, we have the **Apple III Fruit Machine**. Superb full color graphics gives the **Fruit Machine** the 'feel' of a real **Slot Machine**. Arcade style sound effects add to the excitement.

Winning big is just a pull of the handle away. Get your copy today and tomorrow, well the **French Riviera** beckons . . .

The **Apple III Fruit Machine** requires a minimum 128K **Apple III** or **III Plus** and is priced at only: \$19.95 + \$2 for shipping and handling.



The Apple III Card Machine (Blackjack)

After you win big on the **Fruit Machine** you can try your hand at the **Card Machine**! The **Card Machine** is a realistic simulation of the game of **blackjack**. Full color graphics makes this an exciting and challenging program.

You can even use the **mouse** to control the action of the game. Now, if you could only break even...

Each of the multi-colored face cards was created with the **Draw ON III Graphics Tool**.

The **Apple III Card Machine** requires a minimum 256K **Apple III** or **III Plus** and is priced at only: \$24.95 + \$2 for shipping and handling.



SPECIAL OFFER: Order both the **Fruit Machine** and the **Card Machine** today for only \$39.95 + \$3 for shipping and handling.

Selector /// Program Switching Utility

After a long wait and lengthy development period, the *Selector /// Program Switching Utility* is here! For those who are unaware of what a program switcher is, or how it functions, it is a utility that serves as your personal secretary by always having at hand and available those tools you most often work with. Let's assume you have a number of frequently used programs. Perhaps VisiCalc, /// E-Z Pieces, Quickfile or Keystroke. To use any one of them, you must boot the computer with the application program diskette. If you want to switch programs, you must re-boot with a different diskette. For larger programs this may take a minute or so, not counting the hassle of finding the right disk.

Selector /// can make life a little easier for you. If you have a hard disk (or the 800K UniDisk), you can place all of your programs on it and in the morning when you turn your computer on, using the *Selector* start-up disk, you will see a menu listing all of the programs on the hard (or other) disk. You can use any program on the hard disk, and when you exit the program, the *Selector* menu will reappear. Selecting another program is as easy as pressing RETURN. With *Selector ///*, all of the programs and more on the following checklist may be run from your hard disk or UniDisk:

- ✓ Access ///
- ✓ Access 3270
- ✓ Advanced VisiCalc
- ✓ Apple II Emulation
- ✓ AppleFile ///
- ✓ Apple Speller ///
- ✓ Apple /// Pascal
- ✓ AppleWriter ///
- ✓ Backup ///
- ✓ BPI
- ✓ Business Basic
- ✓ Business Graphics
- ✓ Cobol
- ✓ Draw ON ///
- ✓ Easyterm
- ✓ Graph'n Calc
- ✓ Haba Merge
- ✓ Keystroke Data Base *
- ✓ Keystroke Report Generator *
- ✓ Lazarus ///
- ✓ Multiplan
- ✓ Nexus
- ✓ PFS: File *
- ✓ PFS: Graph *
- ✓ PFS: Report *
- ✓ Quick File ///
- ✓ Script ///
- ✓ Senior Analyst ///
- ✓ VisiCalc ///
- ✓ /// E-Z Pieces

BULK RATE
U. S. POSTAGE
PAID
Permit No. 90
Ventura, CA



Many other programs will work under *Selector ///*. The *Selector* allows the copy-protected programs shown above (followed by asterisks) to be placed on your hard-disk, but requires that the 'key' or boot disk for those programs be placed into the built-in drive before starting that particular program.

With pop-up menus, The *Selector /// Program Switching Utility* is a new approach to productivity and cost-effectiveness, making computers more attractive and easier to use for first time users. Complete with a handsome, 122-page hard-covered user's guide, the *Selector /// Program Switching Utility* is priced at just \$99 plus \$7 shipping and handling.