

TM

# ON THREE

The Magazine For Apple III Owners and Users



Volume 3 - Number 4

April 1986  
\$4.00

• **Installing  
the Xebec  
9730**

• **BASIC  
Internals:  
A Look Inside**

• **On Pascal ///**

- Piracy
- Graphics
- and more



**Desktop Manager is here!**

# Apple III 512K Memory Upgrade



= Increased Productivity With a More Powerful 512K *Apple III*!

**ON THREE's 512K Memory Upgrade is the Single Most Exciting Enhancement to the Apple III Ever!**

**Specially priced at just \$399\***  
for a limited time only

Look forward in 1986 to more file capacity for your applications programs like VisiCalc (regular and advanced versions), */// E-Z Pieces*, *Selector III*, *Business Basic*, and others. Imagine having 450K to work with on a spreadsheet model or data base with a 512K Apple *///*. Think of the forecasts you could create. Or how would you be able to type PRINT FRE from *Business Basic* and see 467542 print out on your screen. Wow! The most powerful BASIC around.

The *ON THREE 512K Memory Upgrade* is simple to install by following the directions in the installation manual. Even better, it does not use any of your precious expansion slots and works with all SOS programs. If you ever run out of memory once you have your 512K upgrade in place, you may need a minicomputer!

Another problem the *ON THREE 512K Memory Upgrade* can solve is when you are running a hard disk with *Selector III*

or *Catalyst*. Certain programs take up a lot of memory and sometimes there is not enough to go around. And if you think the hard disk is fast, wait till you try the *RAMDisk* that comes free with the 512K upgrade. It'll amaze you with its speed. If you were used to making notes, etc. while your drive was working, you can forget it.

You see, with the limitations of a 256K system, programs like *Selector III* and *Catalyst*, in conjunction with special purpose utilities like *ONTIME* or the *Calendar Pak* will run on only minimal *Selector* or *Catalyst* systems. This means no spooling and a lot of dynamic driver loading. Who needs problems like this? Now you can run, for example, *Draw ON* with *Catalyst* and see your pictures being printed on the printer while you have already started word processing with *AppleWriter III* or *Word Juggler*.

Read the checklist in the box below to see all the freebies that come with the *ON THREE 512K Memory Upgrade*.

## Look!

At no extra charge, *ON THREE's 512K Memory Upgrade* includes:

- ✓ Complete 24-page instruction manual.
- ✓ Ultra-fast *RAMDisk Drive* with demonstration programs.
- ✓ *The Upgrade to 512K Utility disk* . . . updates all your disks to work with the expanded memory and the Updated version (1.2) of the *System Utilities* program that permits larger SOS DRIVER files.
- ✓ A copy of the *Confidence Memory Program* . . . tests all memory and ensures your 512K Memory board is working correctly.
- ✓ *ON THREE's full 90-day warranty*.
- and of course, an **Apple III 512K memory board with state-of-the-art 256K memory chips.**

ON THREE (805) 644-3514  
P.O. Box 3825, Ventura, CA 93006

Calif. residents add 6% sales tax (products only) We accept Visa, Mastercard, American Express!  
13% surcharge on American Express orders

\* The full purchase price is \$449 plus \$10 shipping and handling. (And plus 6% Calif. sales tax for residents.) After installing the *ON THREE 512K Memory Upgrade*, you can return your old 256K board to us for a \$50 rebate.

If you have an older 128K machine, the cost is a flat \$449 (plus shipping) and no rebate. Installation must be performed by *ON THREE* or a dealer.

*ON THREE* also will install any upgrade for you at just \$50. We offer same day turnaround on 256 to 512K upgrades. Call for more information.

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

# ON THREE

The Magazine For Apple III Owners and Users

T.M.

**Publisher:**

Bob Consorti

**Editor:**

Val J. Golding

**Comptroller:**

Joseph Consorti

**Logistics:**

Janet Schanz

**Interior Art:**

Virginia Carol

**Technical Support:**

Rob Turner

**Order Dept.:**

(805) 644-3514

**Typography:**

West Coast Typography

Van Nuys, CA

**Printing:**

Ojai Printing & Publishing

Ojai, 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: \$25 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. Your full address will be published only when specifically requested.

**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 ///, Apple /// plus, Applesoft, Apple Business Basic, Disk II, Disk ///, UniDisk IIsa, LisaDraw, Macintosh, MacPaint, MacWrite and Profile are registered trademarks of Apple Computer, Inc. MicroSci and Gameport /// are registered trademarks of Stardum Control, Inc. Selector /// is a registered trademark of Sabre Software, Inc. ON THREE, O'Clock, Lazarus ///, Draw On and ONTIME are registered trademarks of ON THREE, the Apple /// magazine.

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

Entire contents copyright ©1986 by ON THREE. All rights reserved.

Volume 3, No. 4

Table of Contents

April, 1986

**FEATURES**

<b>A Look Inside:</b>		<b>The Command Line:</b>	
<b>Business Basic Internals</b>	<b>5</b>	<b>ON Pascal ///</b>	<b>15</b>
<b>Part I</b>		<b>Dennis Cohen</b>	
part I of the most comprehensive		if you have put off investigating pascal,	
exploration of what makes business		here is the perfect opportunity to start	
basic tick, including never before		from scratch	
revealed pointers			
<b>Graphics:</b>		<b>Technical Tips:</b>	
<b>Easter Egg</b>	<b>13</b>	<b>Installing the Xebec 9730</b>	<b>19</b>
<b>Gary Crum/Ian Barland</b>		<b>John R. Sollman</b>	
this simple basic program creates		an apparently formidable task made	
scads of random easter egg designs—		simple with these easy step-by-step	
just in time for the season		instructions	
<b>FBI Apprehends Subject:</b>		<b>Coping With Device Dependent Error 34:</b>	
<b>Piracy Doesn't Pay</b>	<b>14</b>	<b>Taming Timing</b>	<b>29</b>
this release points out, to the tune of		<b>Dan Martin</b>	
250 grand, pitfalls that lurk for		if you've received sos error \$34, your	
profiteering pirates		drive timing may be off. here's a quick	
		step-by-step fix	

**DEPARTMENTS**

<b>The Editor Dishes it Out:</b>		<b>Reading the Mail:</b>	
<b>Apple.Sauce</b>	<b>3</b>	<b>Three Questions</b>	<b>29</b>
<b>Val J. Golding</b>		most readers turn to this page first,	
plentiful platitudes pursuant to		looking for solutions to problems	
perusing program prose			
<b>Notes from the Publisher:</b>		<b>For April:</b>	
<b>Block Write</b>	<b>4</b>	<b>New Products</b>	<b>31</b>
<b>Bob Consorti</b>		once a year about this time, a	
your friend and mine, our esteemed		startling new chip comes out. this	
publisher, opens avenues of thought		year is no exception, and we offer	
for future on three products		insightful insights	
		into its innards	
<b>ON THREE Price List</b>	<b>12</b>	<b>Call Three: Hot Line</b>	<b>32</b>
updated to reflect our newest product		if it's not in print, try one our	
prices		experts	
<b>De Classifieds</b>	<b>21</b>	<b>Apple /// User Groups</b>	<b>32</b>
a buy/sell market designed to meet		newly updated, there are many	
your needs, with special subscriber		groups around the country offering	
rates		their help and expertise	

**ON: The Cover**

This month our cover highlights the first of a two-part paper on Business Basic Internals, perhaps one of the most significant Apple /// articles to appear in print. Shown in detail is ON THREE's **512K Memory Upgrade**, along with the many other internal components.

Captured in its pristine nakedness, the Apple /// was photographed by David Ojerholm, H & O Studios, 614 E. Main St., Ventura, CA 93001

# Smart users select the

# Selector ///

## The smart Program Switching utility

from ON THREE

- Switch to Selector /// now
  - Ideal for UniDisk or hard disk systems
  - Compatible with more than two dozen major Apple /// applications
    - AppleWriter ///
    - Business Basic
    - Haba Merge
    - VisiCalc
    - /// E-Z Pieces
- ... and more

**\$99** plus \$7 s/h

*Selector ///* is a state-of-the art program switcher.

A program switcher is a utility that functions as your computerized personal secretary. Its purpose is to make instantly available to you, without rebooting, a wide range of applications programs stored on your hard disk or high capacity floppy (such as UniDisk). Programs such as AppleWriter ///, Quick File /// and more than two dozen others.

Each time you require a different application, just tell your personal secretary, *Selector ///*, with a couple of keystrokes, and it will be there in seconds. No need to find your way through sub-directories or paw through a stack of floppies.

When you start your system up in the morning, just boot *Selector ///* and that's it for the day. Smart users are switching to and with *Selector ///* now.

### Selector /// works with all of these programs:

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

\*Indicates boot disk required in internal drive.

Selector /// \$99 plus \$7 shipping & handling

ON THREE (805) 644-3514  
P.O. Box 3825  
Ventura, CA 93006

Send me \_\_\_\_\_ Selector ///s

Name \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_

State \_\_\_\_\_ Zip \_\_\_\_\_ Phone \_\_\_\_\_

enclosed \$ \_\_\_\_\_

M/C  Visa  AE\*

number \_\_\_\_\_ exp. date \_\_\_\_\_

Signature \_\_\_\_\_

\*3% surcharge on American Express  
California residents add 6% sales tax.

The Editor Dishes it Out:

# Apple.Sauce

val j. golding

## Hard Talk

One of the advantages in working with a computer like the /// (and particularly with 512K of memory) is the ease with which it interfaces to various peripherals. All one needs is a driver and presto, your new device is usable. Unlike the Apple ][ where far more consideration must be given to questions of compatibility, software, hardware, interface card, protocols, and the like, the /// remains alone and aloof in its sophistication.

We are no stranger to hard disks, they have been around for many moons. The ProFile was one of Apple's earliest, and it opened new horizons for /// users, even with the limited memory available at the time it was introduced. We have used ProFiles a bit, but when we recently installed a *Sider 10*, along with *Selector ///*, we were literally amazed at the tremendous speed improvement on file saves and loads.

What's more, because our work often involves writing about software, how simple and painless it is, indeed, to jump out of AppleWriter and go to BASIC to test or list a program, then jump back and resume word processing.

These are some of the 'miracles' we are learning about with the Apple ///. And so, in a sense, this is what *ON THREE* is all about, to exchange information, acquaint you with new products—and surprisingly there seem to be more coming out now than at any other recent time—both those of *ON THREE* and of other producers. But topmost on our priority list is providing the material *you* want to read. There is, as we have mentioned a number of times before, a need for more articles, programs and hints. Our page rate is \$25 per page (not as reported previously) and not only will you see your name in print, but you can bask in the knowledge that you have helped others by *sharing*. You should not be overly concerned with your writing skills, smoothing out text is one of the functions of an editor. The important thing is *sharing* your knowledge with others.

In many ways *ON THREE* operates as a gigantic user group. In this regard, we would like to suggest you read **Bob Consorti's** *Block—Write* on page 4. There you'll note Bob is asking for your specific suggestions as to new products you would like to see developed by *ON THREE*. Where else could you find a manufacturer and a magazine as responsive to customer/reader input as is *ON THREE*. And for those of you who have been with us for a while, your subscription may be coming up for renewal soon. Certainly, you don't want to miss out on any of our forthcoming products or articles. There is information on this subject and descriptions of many of our new products elsewhere in this issue.

## A Real Look Inside

Sometimes it is difficult to pack all the material you want into a single issue. Such is the case this time around. Recently we glommed onto a stack of deep technical material on *Business Basic Internals*, the first part of which appears on page 5. This article offers information on writing Invokable Modules, etc., and provides never before published information on zero-page pointers and Dispatch routines. If you can't wait the necessary 30 days for the concluding part, here's what we will do: Remit a modest service charge of \$5 to cover photocopying and mailing, and we'll ship out a copy of the complete article. Deal?

*ON THREE* has recently released a line of 10-, 20-, and 34-MegaByte hard disk drives, and while they are relatively simple to install, **John R. Sollman** has written a step-by-step tutorial on *Installing the Xebec 9730* to make things even easier. If you are considering any of the three drives (installation techniques are similar), then John's words are

required reading.

While we're on the subject of disk drives, if you've been getting *Device Dependent Error #34* lately, perhaps you should look into **Dan Martin's** *Taming Timing*, another step-by-stepper that shows how to recalibrate drive speed.

Not everything is techie stuff, however, as we introduce a beginner's series, *ON Pascal ///*, hosted by **Dennis Cohen**, no stranger to our pages. In this opening chapter, Dennis starts at the beginning (or considering it's Pascal, at the top) and explains the Pascal Command Line. Future editions will lead you gently through the the layers of a structured language.

As you read this, Easter will soon be with us and we have accordingly published *Easter Egg*, a dazzling graphic display written in Business Basic by **Gary Crum** and **Ian Barland**, along with a bit of Apple-Foolery to be found on one of the back pages.

With us as always is one of our most popular departments: *Three Questions*. This pass through you'll find questions and answers on compatibility between *Desktop Manager* and Powerkeys, unexplained system failures and Dvorak keycaps as well as a host of others, so dig in and enjoy.

## Coming Attractions

Coming up next month, **Mel Astrahan**, author of *ON THREE's* famous *Draw ON ///* graphics tool, continues his series *Graphically Speaking*, with an extended look at graphics memory organization. Of course we'll conclude the *Business Basic Internals* pair. We will also be covering telecommunications and bulletin boards in the immediate future. That reminds us: our own bulletin board, another free *ON THREE* service, should be up and running by the time you read this. The number is (805) 644-1055.

...

*As we write this in late January, a technological tragedy has unfolded. We would like to devote the balance of this space to the memory of seven brave pioneers.*

## In Memorium

**Gregory Jarvis**  
**Christa McAuliffe**  
**Ronald McNair**  
**Ellison Onizuka**  
**Judy Resnik**  
**Dick Scobee**  
**Michael Smith**

Challenger 51L  
January 28, 1986

*Requiescat in pace*

# Block\_Write

bob consorti

Last month I asked everyone if they would like *ON THREE* to build a slot expansion box for the *///*. Because of printing schedules, I won't get a response for at least another month. I wish there was a quicker way of getting your input back, but for now this forum will have to do.

*ON THREE* has among its resources some fine programming and hardware talent. I firmly believe that if our people can't build or program it—it can't be done. We have a number of projects going, but we would like your ideas. What is it you would like to see for your Apple *///*? How about a faster Apple *///*? More memory? Greater compatibility with the *///e*, *///x*? A new data base program? Home Accounting?

I'd like to have your opinions on what you would like *ON THREE* to develop. At the end of this column you'll find a small form that I'd like you to cut out or copy and mail back to me. With your help, we can analyze the results, and can define just what hardware and software projects *ON THREE* will undertake in 1986.

### ON THREE Bulletin Board

By the time you read this, we will have finally set up our bulletin board. For those of you with a modem and communications software, you can call (805) 644-1055 to obtain the latest Apple *///* information. Our bulletin board will have useful hints, techniques, programs and mini-reviews. You can even order *ON THREE* products directly. If you have questions (or the answers), you can post them for other *///*users to read and respond to.

There is no fee for this service—all you have to do is have an Apple *///*—one of the finest computers ever built.

### Subscription Renewal Time

You didn't think this time would ever come, did you? Well, *ON THREE* is coming up on its 12th issue. This means it's time for some of you to renew your *ON THREE* subscription. I have faith that with *ON THREE*'s recent return as a monthly magazine, specifically for Apple *///* owners and users, almost all of our current subscribers will want to renew. Combined with all of the other products and services that we offer for the *///*, we are the *one* source of information that you [need] to have.

Over the past couple of years *ON THREE* has developed the most innovative products—all designed to help keep your Apple *///* working for years to come. Items such as our 512K Memory Upgrade and the Desktop Manager were developed per your suggestions. In the future we will also be working for you by

creating the products you ask for. Without an *ON THREE* subscription, you have no 'pipeline' to our stream of new and exciting Apple *///* products.

I won't spend any more column space on the subject—I think you know that a subscription to *ON THREE* is one of the best buys in the industry. It ensures that you will continue receiving all the latest information, product reviews and that your Apple *///* needs will be fulfilled.

### Disk Copying

Today, it is a rare Apple dealer who even knows what an Apple *///* looks like. What happens if your AppleWriter or VisiCalc disks go bad? Until now, your only choice was to buy an Apple *[[* program to copy those disks. We have developed a driver that allows the standard Apple *///* System Utilities program to copy AppleWriter and VisiCalc (regular or Advanced Version).

Never again will you have to worry about damaging your original disks—you can make copies of them! Our *UNCOPYPROTECT* driver allows you to make archive copies of your own AppleWriter or VisiCalc disks. The driver comes on a disk including full documentation, everything you need to back-up your software—all for only \$19.95 plus \$2 for shipping.

The *UNCOPYPROTECT* driver also allows running AppleWriter or either VisiCalc from *Selector ///* without any special 'key' disk.

NOTE: The above utility is intended only for use in making legitimate backups by the owner of the software package. ***ON THREE has never condoned, and will not condone copying software for any other purpose.***

### Desktop Manager Modules

The first add-on module for the *Desktop Manager* should be ready by the time you receive this magazine. The *Disk Utilities* module will cure any headaches you may have when working with your *///*. Have you ever been in the middle of a typing in data for an hour and ready to save it, only to find that you don't have a formatted blank disk? How about this: A program asks you to type in a file name. Do you remember what it was? It would be a pain if you had to quit the program and run the System Utilities, just to find the name of a file.

Up until the *Desktop Manager* came along, it was a problem. In fact, you couldn't do it. Now, with the addition of the *Desktop Manager* and the *Disk Utilities* module, you can do all this and more. A simple keypress suspends your program and allows you to format a blank disk. Another

keystroke lets you list the files on a disk. Are you familiar with the System Utilities program? The *Disk Utilities* module has most of the features of the System Utilities—all available in an instant—and while you're still working with your current program.

I personally use the List Files option a dozen or more times a day—it alone is worth the price of the module. Another great *ON THREE* value for only \$44.95. Call *ON THREE* for the exact shipping date.



### Apple *///* Software and Hardware Wish List

Please Return To:

*ON THREE, Inc.*  
Attn: Bob Consorti  
4478 Market Street, Suite 701-702  
Ventura, CA 93003

Dear Bob,

I'd like to see the following items for my Apple *///*.

Hardware Items:      What I Would Pay:

- Apple *///* Speed Up Kit \_\_\_\_\_  
(Makes all programs run 3-4 times faster)
- Slot Expansion \_\_\_\_\_
- More than 512K of memory \_\_\_\_\_
- Larger capacity hard disks \_\_\_\_\_
- Full Apple *///e*, *///x* compatibility \_\_\_\_\_
- Better/Faster Graphics \_\_\_\_\_
- AppleTalk Networking \_\_\_\_\_
- Other hardware (please specify) \_\_\_\_\_

Software Items:

- Data Base \_\_\_\_\_
- Home Accounting \_\_\_\_\_
- Hard Disk Boot \_\_\_\_\_
- Fast Hard Disk Backup \_\_\_\_\_
- Slower Hard Disk Backup (Just Kidding) \_\_\_\_\_
- Fast Floppy Disk Copy \_\_\_\_\_
- Utility For Larger SOS.DRIVER files \_\_\_\_\_
- Other software (please specify) \_\_\_\_\_

Problems You Would Like Fixed:

# Business Basic Internals

part i

## Introduction

The Business Basic statements INVOKE and PERFORM provide an easy to use interface to assembly language procedures and numeric functions. This allows extending the capabilities of BASIC for any given application without requiring that it carry around all possible functions. When these additional functions are not required this has the effect of conserving memory.

If you have not already read the sections of the *Apple Business Basic Reference Manual* that pertain to this subject (Chapter 6 of Volume 1, Page 161 . . .), you should do so at this time. This paper assumes that you have read and are familiar with that material as well as the Apple /// Pascal Assembler (ref: *Apple /// Pascal Program Preparation Tools Manual*, Chapter 2).

## The INVOKE Statement

The INVOKE statement accomplishes four functions:

1. Deletion of all previously invoked modules and recovery of the memory used by those modules.
2. Loading of one or more assembler modules from a storage volume (diskette or hard disk).
3. Relocation for execution of the code image(s), excluding any relocation tables.
4. Building the procedure name table to allow symbolic access via the PERFORM statement or EXFN functions.

The INVOKE statement creates a procedure name table to be used by the PERFORM statement. This table contains the PERFORMable names for the assembly language procedures and functions. It also contains the expected parameter word-count for validating the amount, but not the form, of the parameters passed.

*Those of you who are familiar with the "innards" of the Apple II, as well as the ///, may recall an article that appeared in the Apple Orchard in 1980, and subsequently revised and printed in All About Applesoft, by John Crossley of Apple Computer, entitled: "Applesoft Internals." It is likely that the publication of that one article was responsible more than any other single event for the onslaught of software development in those early years. To the delight of hackers and developers alike (and in those days there was little difference), complete, although brief, descriptions of all of the major internal routines allowed programmers full access to the Applesoft ROM routines along with infor-*

*mation on using CHARGET and TEXTPTR, etc.*

*Business Basic, however, was a slightly different story. Apple Computer, in their infinite wisdom, declared the /// a business machine and decided that what the user didn't know would never hurt. It should be noted, however, that Apple did take pity on the Apple /// developers and made available several pages of tech notes that were roughly the Apple /// equivalent of the Applesoft Internals material. The situation has changed since the demise of Apple /// production and along with it, technical support. Thus we are pleased to present in public print for the first time those tech notes pertaining to Business Basic Internals.*

The lack of specific type validation implies that this facility can be misused with potentially fatal results for the entire application. An example of this might be passing an integer in place of an address parameter. Since both these parameters are of the same size on the 6502 stack (i.e., two bytes), the PERFORM statement will go ahead and interpret the data as an address, since that is what it is expecting. In this event, the procedure will attempt to execute code at the false address indicated by the data bytes.

The PERFORM statement saves the name following the word PERFORM and begins processing the parameter list (if one exists) by pushing the parameter data on the stack. When this processing is complete, the procedure name table is then searched for the supplied name. When it is found, the stacked word count is checked against the count from the procedure name definition and if they match, the invocable routine is entered via a JSR instruction.

## The EXFN FUNCTION Calls

The EXFN. and EXFN%. functions are almost identical to PERFORM except that a function name must be found in the procedure name table and four dummy bytes are pushed onto the stack just before the JSR is executed. The four dummy bytes are not included in either the expected word count or verification process.

## Invokable Module Description

An invokable module is a relocatable assembly language code file (no Pascal routines are allowed!) produced by the Apple /// Pascal Assembler.

The relocatable code file may not have been linked or in any way modified after generation by the Pascal assembler: The linker in the Apple /// Pascal system is *not* a linker that will combine a collection of assembler code files into a bigger linkage-edited code file. It is designed only to link assembler routines to Pascal programs.

A single invokable module may contain any reasonable number of entry points. These are specified by using the .PROC directive for procedures and/or .FUNC's within that module. The name given with these directives is the symbolic name that will be used by the PERFORM and EXFN statements to access the assembly language routine.

Several directives in the Pascal Assembler can *not* be used when creating invokable modules for use with Business Basic. .ABSOLUTE, .ALIGN, .INTERP, .CONST, .PUBLIC, .PRIVATE and .ORG will generate an INVOKE ERROR if used.

The .DEF and .REF assembler directives can be used within a *single* invokable module to allow routine-sharing among multiple .PROC's and/or FUNC's within that module. Cross-module .REF's and .DEF's are not supported and their use will cause an INVOKE ERROR.

Invoked modules will *always* be found in bankable memory within a single bank. This limits invoked modules to 32K of combined memory including the relocation data and interpreter tables.

The assumption is made that the reader understands the special enhanced indirect addressing of the Apple ///. The pointer addresses created by Business Basic are always passed to invoked modules with the extended byte adjusted so that the high byte of the enhanced indirect address is in the range \$02 to \$81. This is done to avoid the anomalies that occur when (\$00xx),Y or (\$FFxx),Y addressing is attempted.

Note that system interpreters are always assigned their zero page and stack at addresses \$1Axx and \$1Bxx, and the registers that control this assignment should never be modified by an invokable module.

The expected word count parameter of the .PROC and .FUNC directives gives the number of [two-byte] *words* (not bytes) that will be passed on the 6502 stack by BASIC to the invokable module. This count is calculated using the word count table (Table I), for the various types of possible parameters that BASIC can pass.

The total word count of all expected parameters for any given entry point must be added together and the sum specified in the .PROC or .FUNC directive. BASIC does not allow variable length parameter lists when viewed as a word count. You can pass two single-precision reals in place of one long integer, or four integers in place of two single-precision reals, but this will probably give your assembly code a severe case of indigestion. Thus the parameter passing is positional only, *not* symbolic and fixed length.

### Parameter Description and Conventions

The specific order and type of parameters chosen for an invokable module is up to you. You can also decide either to pass a parameter by value or address. The parameters are processed from left to right and pushed on the stack in that order, thus the last parameter in the argument list will be the first to be pulled off the stack.

Parameters are passed by address by preceding the variable name in the PERFORM statement with the commercial "at" sign [@]. This causes two actions by BASIC:

1. The low two bytes of the enhanced indirect address are placed on the stack.

2. The enhanced indirect address is also set up on the interpreter zero page and in the interpreter extended page (\$16xx).

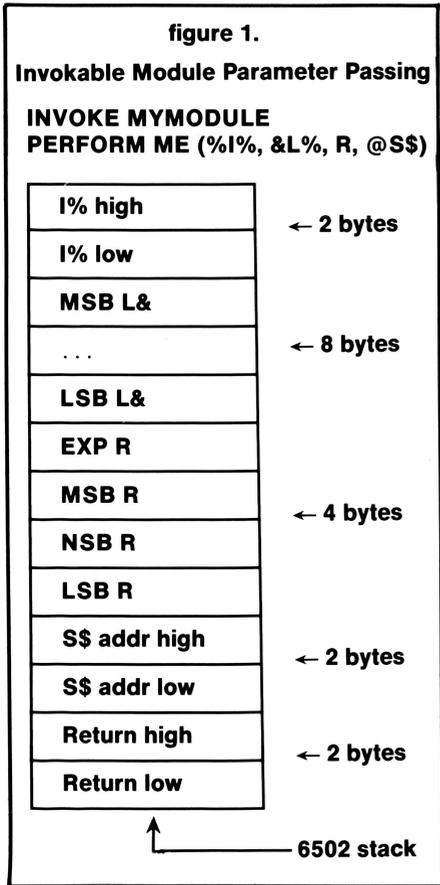
The PERFORM statement provides for up to eight address pointers in zero page for use by any specific procedures or function. This 'pointer stack' begins at address \$E8, \$E9 and \$16E9 and continues through \$F6, \$F7 and \$16F7. This allows you to set up and use extended addressing.

When both the address and value modes of parameter passing are used in a single procedure or function call, the pointer stack is filled in order as address (@) parameters are encountered. Thus the first occurrence of an @ parameter address will be found at \$E8, \$E9, and \$16E9 regardless of its position in the parameter list.

Note that all parameters that are passed by address *must* be accessed via the enhanced addressing mode. This is the only means of access that will work, because the bank holding the assembly code of the currently executing invoked module must be selected as the current bank, but BASIC's data storage tables may not be in the currently selected bank. It is impossible for invoked code to bank switch while residing in bankable memory itself!

Table I: Parameter Descriptions			
Parameter Description	Type Char.	Word Count	Byte Count
Short Integer	%	1	2
Long Integer	&	4	8
*Single-precision Real	none	2	4
**Address	@	1	2
*Numeric expressions follow the specifications for the appropriate numeric type.			
**Extended addresses are three bytes, but only two are passed on the stack.			
Table II Parameter Stack Usage			
Integer%	Long Integer	SP Real	@Address
LSB	LSB	LSB	ADDRLO ← Top of stack
MSB	*	*	ADDRHI
	*	MSB	
	*	EXP	
	*		
	MSB		
	EXP		

Parameters are pushed on the stack high (most significant) byte first, therefore the invoked code must pull parameters off the stack low (least significant) byte first. This is shown graphically in Table II for the five types of parameters. Note that expressions will have the appropriate number of bytes pushed on the stack for the resultant value. Note also that the JSR return address is on the stack above the parameter data in the usual low byte order.



**String Parameters**

String parameters can only be passed by address and explicit string functions are not allowed. String processing can be done by passing string addresses and operating on strings in memory with the PERFORM statement.

BASIC uses a three-byte *descriptor* for each string. The address passed by BASIC for a string parameter is actually the address of this three-byte descriptor, and not a pointer to the actual string data.

A string descriptor consists of a length byte followed by a relative offset. The length is an unsigned

**Figure 2. Pointer @ Passing**  
**PERFORM, EXFN., EXFN%**

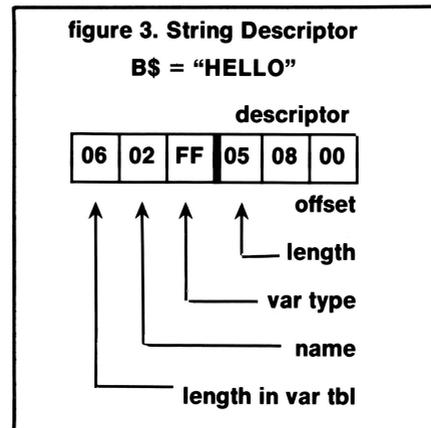
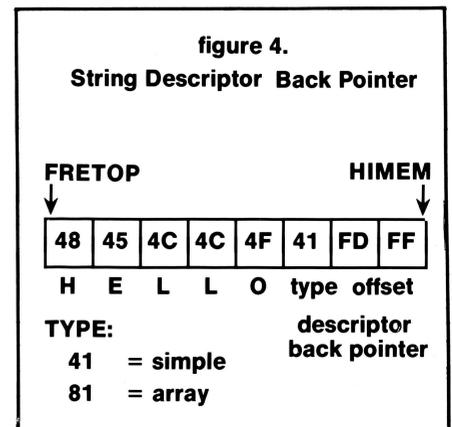
The Apple /// uses three-byte extended addressing pointers. They are passed both on the 6502 stack (two bytes) and in the interpreter's zero page and X-page. Up to eight pointers may be passed for simple variables and array elements.

(Zero page register = \$1A, so Extend page is \$1C)

\$E8	\$EA	\$EC	\$F6	Address low
\$E9	\$EB	\$ED	\$F7	Address high
\$16E9	\$16EB	\$16ED	\$16F7	Extend ("X") byte
1st ptr found	2nd ptr found	3rd ptr found	8th ptr found	

The parameter list is processed from left to right. These pointers are only passed for "@" variables in the parameter list, *not* for the 'pass by value' expressions or variables.

value, and if zero, indicates a null string (a string with no data). The next two bytes are a relative offset to the beginning of a string buffer allocated for the contents of a given string. The offset is an unsigned number that is subtracted from the highest address available for the storage of string data. The result of the subtraction is used to locate the beginning of the buffer which contains the actual ASCII data. When the length is zero, the offset is invalid, therefore length must be checked before using the offset.



A string data buffer contains more than just the ASCII characters that have been assigned to the string. It also contains a second three-byte descriptor that points back to the original string descriptor. This back pointer consists of a type byte and an unsigned two-byte offset from the beginning of the simple variable table or the array variable table. The type byte normally specifies to which of the two variable tables the offset applies.

Since the descriptor and the back pointer have 16-bit relative offsets, all the string descriptors must occur within the first 64K of the appropriate variable table and the string data pool itself may not exceed 64K.

Trying to create new variables, particularly strings, or to change the length of a string involves the entire BASIC memory system. This is necessarily so since either of these actions could cause 'garbage collection', a process which compresses the string data storage area and reclaims free memory for new strings.

Business Basic provides an interface that allows access to some of the interpreter's internal subroutines that process strings and perform other functions. This interface, known as DISPATCH, is somewhat analogous to the ProDOS MLL, or Machine Language Interface, and will be described later. It is the *only* recommended interface to the internals of the Business Basic interpreter.

The difference between a 'pointer to a descriptor', a 'descriptor', and a 'pointer to the string data area' must be clearly understood. The subroutines to be described later, such as PTRGET, normally return a pointer to a descriptor.

A descriptor cannot be used directly to reference the string's data without first being converted into a pointer to the string data. (This conversion is performed by the routine NOTNOW.) When BASIC returns a pointer to a descriptor or a temporary descriptor, the descriptor always contains a relative offset.

If you want to assign some ASCII data to a string variable, the ASCII data must be copied into BASIC's string data storage area and a descriptor built for that data. Then the new descriptor can be copied onto the existing original string descriptor in the appropriate variable table, and the back pointer in the string buffer can be set up to point to the string's descriptor in the variable table.

Before the descriptor for an existing string variable is replaced with a new one, the old string data buffer must be deallocated. Deallocation involves changing the back pointer to indicate the size of the 'free memory' space that is being deallocated.

BASIC has two routines, STRCP and INPCOM that will perform these two tasks for an invocable module. STRCP copies the ASCII data into the string data storage area, creates the associated null back pointer, and builds a temporary descriptor for the data. If the DOPAR routine is used to evaluate a string expression, the string result is returned as if STRCP was used to create it. Temporary descriptors are stored in zero page by the various routines. Temporary descriptors are exactly like regular string variable descriptors and if you wish to reference the ASCII string data they must also be converted by NOTNOW.

INPCOM performs the assignment of your temporary descriptor to a variable's descriptor, deallocates the old one and fills in the null back pointer. Using strings in BASIC is complicated, and it is recommended that you clarify your understanding of how it works by explaining it to another programmer.

### Functions and Passing their Results

Externals functions in Business Basic operate through the EXFN. and EXFN% statements. EXFN. is used to return single-precision real floating point results and EXFN% is used to return integer results.

These functions return their results on the 6502 stack. To be compatible with Apple /// Pascal, which pushes four dummy (BIOS) bytes on the stack, BASIC likewise pushes four dummy bytes on the stack after each parameter. This requires that your external function routine pull the four bytes off before pulling any parameter bytes. The dummy bytes contain meaningless garbage and should be ignored.

The results must be pushed on the stack in the same manner as they are passed, i.e., with the LSB stacked last, just before the return address. The integer result is a signed two's complement integer with the sign bit taken as bit seven of the high byte.

The single-precision real result is a normalized 23-bit mantissa with an assumed significant one-bit which is replaced by the mantissa sign bit and an excess 128 signed exponent. When the mantissa sign bit is set, or one, the mantissa is negative and when the exponent is less than 129, the number is less than one, except that a zero exponent defines a zero result although the bytes of the mantissa may not be assumed to be zero (i.e., they may be non-zero even though the number is zero). For clarity, some examples are shown in Table III.

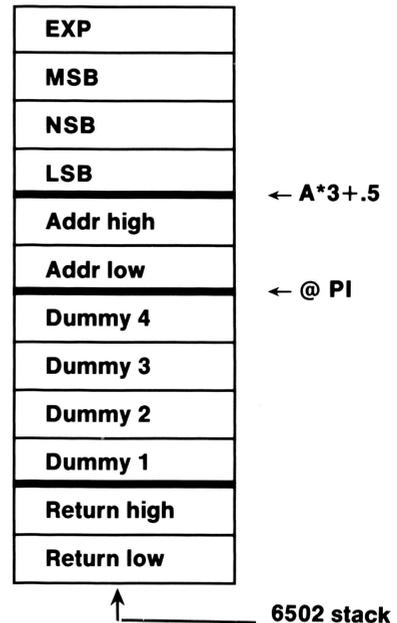
figure 5. EXFN. Parameter and result passing

INVOKE MYPACKAGE

PI = 3.14159

X = EXFN.XX( A\*3+.5, @PI)

Stack on entry to code:



Stack just before RTS to BASIC:

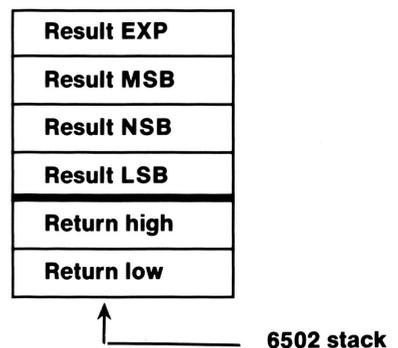


Table III SP Real Example

Numeric Value	Packed Expn	SP-Real MSB	...	(Hex) LSB
.0078125	7A	00	00	00
.25	7F	00	00	00
.50	80	00	00	00
.999999	80	7F	FF	F7
0	00	xx	xx	xx
1	81	00	00	00
4	83	00	00	00
17	08	00i	00	00
128	00	00	00	00

(xx=0 for literal assignment)

**Figure 6 Parameter Pull Order**

Real=	EXP MSB NSB LSB	four bytes
INT%=	High Low	two bytes
Long Int&=	MSB ... LSB	eight bytes
Ptr@=	Addr. low Addr. high	two bytes
X-Byte?		

### Interfacing With the BASIC Interpreter

Invokable modules used to extend the Business Basic language commonly will need to access some internal subroutines within the interpreter. A special interface is provided for this purpose.

This interface has been specially designed to maintain the independence of code within the interpreter and the invokable modules while allowing access to the interpreter. This should allow the interpreter to be modified and upgraded in the future without requiring modification of existing invokable modules. While Apple no longer supports the /// and presumably will not be releasing future enhancements of Business Basic, it is not without the realm of possibility that *ON THREE* or some other independent developer may give consideration to this prospect. Thus the use of any other information or linkage is not recommended, since it could become invalid at a later date.

The call interface consists of a routine number and a fixed address entry point in the interpreter's zero page. The routine number is stored at \$E7 and a JSR to DISPATCH is then executed. DISPATCH is located at \$E4 in zero page.

Invokable modules will normally reside in a bank other than the one containing the desired routine, so direct access of the interpreter is impossible in many cases. DISPATCH performs the necessary bank switching, calls the requested routine and then reverses the bank switch and returns to the invoked module.

Most, if not all, of the routines that can be used within the interpreter can take error exits which will go through the BASIC interpreter and not return to the invoked module. The BASIC program should therefore be prepared to properly handle such errors when the occasion arises.

SOS calls can be made directly to SOS from invoked procedures or functions. Doing so can be very useful and at times very dangerous: BASIC may have files open, memory allocated, events armed, etc. and doing anything to undermine the assumed state of these things will not be reflected in the status information being maintained by BASIC. This

does not preclude making SOS calls, but some reasonable care should be given in selecting what the invokable module does directly with the operating system.

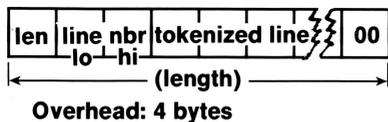
Table IV is a summary list of some of the routines available through the DISPATCH interface. These routines are described in further detail in part two of this article. Obvious by their omission, there are undocumented values that have a place in this table and that do exist. Apple warns us that they are not usable without intimate knowledge of the interpreter internals and cautions that anyone using them does so at their own risk.

**Table IV**

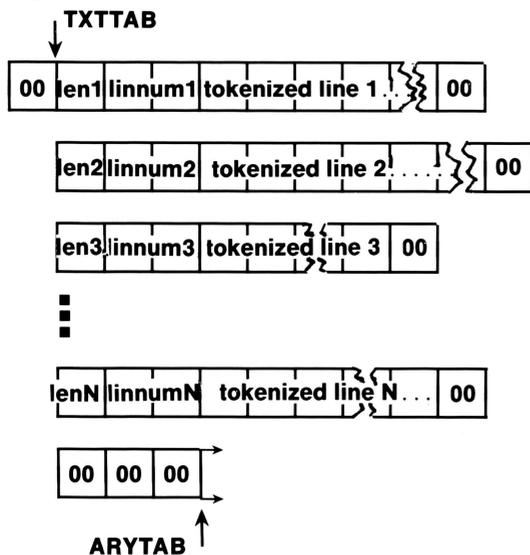
Name	Nbr.	Description
DOPAR	0	Evaluate an expression, return type and result
PTRGET	1	Find, in the variable tables, the variable at TXTPTR
ERRDIR	5	Returns if a program is running, else ILLEGAL DIRECT error
LINGET	6	Converts the line number in LINNUM into a binary integer
GOTOB	7	Returns TXTPTR at the line in LINNUM if it exists, else error
GETADR	8	Rounds real in FAC to a 16-bit address in POKER, POKER+1
FNDLNCO	9	Search forward from Y-X-A for line number >= LINNUM
FNDLIN	10	As above with search starting at beginning of program
NOTNOW	13	Given a pointer to a string descriptor, returns string data pointer and length
ERROR	14	Raises BASIC error condition given in X. May not return
SERROR	15	Raises if possible BASIC error matching the SOS error in A
SCRUNCH	16	Requests BASIC to free up A pages of unused memory to SOS
EXPAND	17	Inverse of SCRUNCH. Expand memory for BASIC by A pages
FRECNOW	20	Free up string descriptor and string space at pointer in FAC
OPENIT	22	Open the filename at TXTPTR if it is of the file type in X
GIVAYF	25	Convert A-Y integer into sp-real in FAC
POSINY	26	Rounds sp-real and converts to positive integer in FAC
FIN	27	Converts ASCII data at TXTPTR into floating numeric in FAC
DATAN	32	Moves TXTPTR to end of the current tokenized statement
STRCP	33	Copies string data at STRNG1 into string data area and builds temporary descriptor pointing to it
INPCOM	34	Assigns a string (pointer in FAC) to string variable in VARPNT
LETP2	35	Assigns a value in FAC to a variable of any type in VARPNT
INT	39	Returns sp-real in FAC of the integer portion of real in FAC
CVT2STR	44	Converts FAC into a string and returns descriptor in FAC

figure 7. BASIC formats

Line Format:

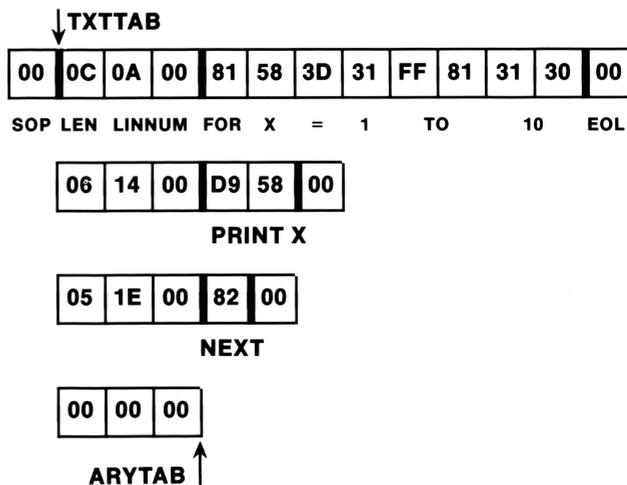


Program Format:



Overhead = 4 bytes  
Lines are stored in ascending order

figure 8. Sample BASIC program



Equivalent source program:  
10 FOR X = 1 TO 10  
20 PRINT X  
30 NEXT

At this late date, it is unlikely that those undocumented entry points would be changed. Readers having additional information are urged to submit it to *ON THREE* for publication.

Business Basic is a tokenizing interpreter that upon program entry compresses keywords to one-byte tokens. The resulting code is often referred to as *program text* or *tokenized program text*. No other processing, syntax checking or conversions are performed until statement execution of the program text occurs. BASIC scans the tokenized form of a statement byte by byte using its own program pointer, known as TXTPTR. All statements commence with a token and the token is used to call the proper routines to execute each statement. Each statement routine performs the necessary syntax checking whenever a statement is executed. TXTPTR's position determines what line and statement of the program is currently executing.

Some of the routines that can be called by the DISPATCH interface may modify TXTPTR. Since TXTPTR is the sole means of continuing BASIC program execution after returning from an invoked module, it must be preserved if any of the routines that change it are going to be used.

An example of this is the OPENIT routine for opening a file. It requires that TXTPTR be modified to point to the pathname to be opened. If the

existing TXTPTR (all three bytes, remember) is not saved and restored, when you RTS back to BASIC, it will attempt to continue program execution using data found at the end of the pathname, since that is where OPENIT left TXTPTR.

Business Basic implements TXTPTR as a subroutine that resides in zero page. This subroutine has two entry points, CHRGET and CHRGOT. This routine is very useful for syntaxing tokenized ASCII data. CHRGET (JSR \$B5) gets the next sequential non-space character or token from the data pointed to by TXTPTR (\$BC). Upon return from CHRGET, the accumulator contains the character or token and the 6502 status flags are set as follows:

## ON THREE Presents . . .

### LAZARUS /// File Recovery System

- A wrong keystroke can lose hours of work
- The correct keystroke with Lazarus /// can undelete those lost files. Need we say more?

**\$49.95** plus \$2 s/h

- Not copy-protected—Installs easily on Selector ///, Catalyst

```

*****
2 *                                     *
3 *           figure 9                 *
4 *                                     *
5 *           CHRGET Routine          *
6 *                                     *
*****
8
9          ORG      $B5
10 *
11 CHRGET   INC     TEXTPTR      ;Low byte
12         BNE     CHRGOT
13         INC     TEXTPTR+1    ;High byte
14 TEXTPTR  =       *+1
15 CHRGOT   LDA;    $0200      :Load byte from program
                               text
16         CMP     #$3A        ;Colon
17         BCS     END          ;If so, RTS
18         CMP     #$20        ;Space?
19         BEQ     CHRGOT      ;If so, restart
20         SEC
21         SBC     #$30
22         SEC
23         SBC     #$D0        ;Clear carry if digit,
                               otherwise
24         set
00CC: 60      24   END     RTS

--End assembly, 24 bytes, Errors: 0

```

Z-flag: set if the character is a colon (\$3A) or  
a line terminator (\$00)  
cleared for all other conditions

Carry:  
clear if the character is an ASCII digit  
set if the ASCII character is non-numeric

CHRGOT (\$BB) returns the current character at TTXTPTR if it is not a space (\$20), otherwise the next non-space character is returned. CHRGOT performs the same functions as CHRGET except that TTXTPTR is not incremented before fetching the character. Normally, CHRGOT is used only after a call to CHRGET and if this is always true, CHRGOT will not change TTXTPTR, since CHRGET will never leave TTXTPTR pointing to a space.

Figure 9 is an assembler listing of the Business Basic CHRGET routine, whose purpose, design, and coding is byte-for-byte identical with that of Applesoft Basic.

Many of the characters available are really three-byte enhanced indirect addressing pointers and should always be treated as such, regardless of actual content. BASIC goes to considerable effort to maintain the high byte of these pointers in the range of \$02 through \$81 to prevent

anomalies of enhanced indirect addressing from messing up and crashing the system. Before you attempt writing invokable modules you must have a *complete* and thorough understanding of this Apple /// addressing mode.

In the above descriptions, BASIC defines a floating point accumulator known as the FAC. This ten-byte area, starting at \$69 in zero page, is used for many purposes. These purposes are varied, but the normal ones relevant to this paper and the Business Basic internal routines are input arguments and output results. These come in four types that use different parts of the FAC as shown in figure 10. The locations with the FP comment are used for sp-real, those with the LI comment are used for long integer data, SI indicates short integer, and STR-PTR refers to pointers to string descriptors.

The FACLO location is used by the floating point routines but is not stored in the storage tables after computations. This extended precision at compute time minimizes most of the anomalies usually associated with floating point numerics. In the previous description of floating point results that are returned by an invoked EXFN routine, the 'packed' format of BASIC real numbers was described. The unpacked format of reals in the FAC is the same, except for the mantissa sign bit which must be removed from bit seven of FACMSB and placed in bit seven of FACSGN and the 'implied' most significant mantissa bit must be set on in FACMSB.

Pointers to string descriptors left in FACMO, FACLO, and FACMOXB may or may not be enhanced indirect addressing pointers. Routine 33 creates a 'temporary string descriptor' for your string data (possibly read from a storage volume). These temporary descriptors must be released with FRECNOW before returning to BASIC if they are used in the invoked module. The temporary descriptor stack is located in zero page and so a pointer in the FAC left by STRCP will not actually be an enhanced indirect pointer, but to simplify writing invokable code, you should always handle these pointers as if they were three-byte pointers.

*This concludes Part I of our two-part Basic Internals paper. For those of you who have a real need to pursue the substance of the internal DISPATCH routines, we have made available, for the nominal service charge of \$.50 postpaid, photocopies of the entire article, part two of which will appear in the next issue of ON THREE. Additionally, we expect to be able to update, most likely in the June issue, additions and corrections to the list of zero page pointers and other data.*

**figure 10.**  
**General Purpose Floating Point Accumulator Fields**

1601:	ETXPAGE	.EQU	1601	;Add 1 to offset to X-Byte from pointer
0069:	FAC	.EQU	69	;FP LI
0069:	FACEXP	.EQU	FAC+0	;FP LI
006A:	FACMSB	.EQU	FAC+1	;FP LI
006B:	FACMSB	.EQU	FAC+2	;FP LI
006C:	FACMO	.EQU	FAC+3	;FP LI SI STR-PTR
006D:	FACLO	.EQU	FAC+4	;FP LI SI STR-PTR + FACMOXB
006E:	FACSGN	.EQU	FAC+5	;FP LI
006F:	FACX1	.EQU	FAC+6	; LI
0070:	FACX2	.EQU	FAC+7	; LI
166D:	FACMOXB	.EQU	FACMO+ETXPAGE	;Address of string X-Byte

## Products & Services Available for the Apple ///

### Current ON THREE Price List

Product	Price	S/H
12-issue subscription to ON THREE	\$40.00	
Back issues of ON THREE	\$5.00 ea	
Disk Of the Month (DOM)	\$14.95 ea	\$2.00
Order 2 or more for only	\$12.50 ea	\$4.00
Desktop Manager-ON THREE O'Clock Combination	\$173.95	\$8.00
Desktop Manager	\$129.00	\$6.00
Desktop Manager Disk Utilities	\$44.95	\$3.00
Desktop Manager ASCII Conversion Table	\$9.95	\$3.00
ON THREE O'Clock	\$49.95	\$3.00
Lazarus /// Version 2.0	\$49.95	\$2.00
Draw ON ///	\$179.00	\$5.00
Apple /// Fortran	\$99.00	\$5.00
Selector /// Program Switching Utility	\$99.00	\$7.00
Uncopyprotect Driver	\$19.95	\$2.00
CROSSWORD-SCRAMBLER	\$19.95	\$2.00
The Apple /// Fruit Machine (Slot Machine Game)	\$19.95	\$2.00
The Apple /// Card Machine (Blackjack Game)	\$24.95	\$2.00
Fruit-Card Game Combo	\$39.95	\$3.00
Game Card ///	\$59.95	\$2.00
Micro-Sci Disk Drives		
A3-140K	\$299.00	\$6.50
A143-560K	\$399.00	\$6.50
3M DS, DD Disks (10) for A143	\$39.95	\$3.00
Apple /// Unidisk 3.5 (800K disk) with interface card, Apple /// software & documentation	\$499.00	\$10.00
Unidisk doc & disk only	\$50.00	\$3.00
Sider 10 (10 MegaBytes)	\$999.00	\$35.00
Sider 20 (20 MegaBytes)	\$1299.00	\$35.00
(Siders can be daisy-chained)		
Xebec 9730 (34 MegaBytes)	\$1999.00	\$35.00
512K Memory Upgrade (256K to 512K) *You remit \$449 and we rebate \$50 when the old 256K board is returned.)	\$399.00*	\$10.00
256K Memory Upgrade (128K to 256K)	\$200.00	\$10.00
256K or 512K upgrade installation by ON THREE (appt. only)	\$50.00	
256K Reconditioned Apple ///	\$999.00	\$50.00
512K Reconditioned Apple ///	\$1399.00	\$50.00
Apple //e Mouse and interface card (can be used with Draw ON /// or Desktop Manager)	\$150.00	\$5.00
Dust Cover for Apple /// with Monitor ///	\$11.95	\$2.00
Dust Cover for Apple /// with ProFile & Monitor ///	\$12.95	\$2.00
I ♥ MY APPLE /// T-Shirts Sm, Med, Lrg, X-Lrg (Color choices: Yellow, beige, white, blue)	\$11.95	\$3.00
I ♥ MY APPLE /// Sweat Shirts Sm, Med, Lrg, X-Lrg (Color choices: Silver, white, blue, yellow)	\$18.95	\$3.00
I ♥ MY APPLE /// Caps	\$5.95	\$2.00

†subject to increase

California residents add 6% sales tax.

To order, call (805) 644-3514 or write:

ON THREE, Inc.

Attn: Order Dept.

Post Office Box 3825

Ventura, CA 93006

We accept Visa, Mastercard, and American Express. There is a 3% surcharge on orders charged to American Express.

### Table V

#### Useful Business Basic Zero Page Variable Equates

Address	Name	Description
\$07	VALTYP	Input/output variable type flag
\$08	INTFLG	Input/output variable type flag
\$0A	SUBFLG	Must be \$00 for PTRGET and DOPAR calls
\$0D	ANYNUM	Output flag from FIN
\$17,18	???	
\$19,1A	VARPNT	Points to first byte beyond last used vbl's name*
\$1B,1C	???	Points within BASIC program*†
\$24,25	LINNUM	16-bit register for line numbers, etc.
\$24,25	POKER	Output data from GETADR routine
\$35,36	INDEX	General purpose pointer for many routines*
\$39,3A	TXTTAB	Pointer to beginning of program text*
\$3B,3C	VARTAB	Pointer to beginning of simple variable table*
\$3D,3E	ARYTAB	Pointer to beginning of array variable table*
\$3F,40	STREND	Pointer to end of variable storage*
\$41,42	FRETOP	Pointer to end (bottom) of string storage area*
\$43,44	INVTAB	Pointer to beginning of invoked modules*
\$45,46	RAMEND	Pointer to highest available memory?†
\$47,48	FRESPC?	General purpose pointer*†
\$49,4A	HIMEM	Highest location, less one, available to BASIC*
\$4B,4C	CURLIN	Current line number being executed
\$4D,4E	OLDLIN	Last line number executed†
\$4F,50	OLDTEXT	Pointer to next stmt to be executed*†
\$51,52	DATLIN	Line number of current DATA statement†
\$53,54	DATPTR	Pointer to next byte of data to be read*†
\$55,56	INPTR?	Pointer to current input source*†
\$57,58	FORPNT	Pointer to string variables, general purpose*
\$5C,5D	*†	
\$5E,5F	*†	
\$62,63	*†	
\$67,68	LOWTR	Output from FNDLNCO routine*
\$69-70	FAC	General floating point accumulator
\$6C,6D	FACMO	General string pointer in FAC*
\$7B,7C	STRNG1	Input pointer for STRCP*
\$B5	CHRGET	Entry point: get next character or token
\$BB	CHRGOT	Entry point: get current character or token
\$BC,BD	TXTPTR	Current character or token pointer*
\$E4	DISPATCH	Internal entry point for BASIC routines
\$E7	SUBRNUMB	Internal subroutine request number for DISPATCH
\$E8-F7	VARPTRs	Call by address enhanced indirect pointers (8)*

An asterisk [\*] following descriptions in the above table indicates a three-byte pointer. The location of the third byte, or X-byte, can be calculated by adding \$1600 to the high byte of the zero page pointer. X-byte labels are formed by adding an "XB" suffix to the two-byte label names. A dagger [†] following a description indicates this pointer did not appear in the original paper and has been added by the editor or others. In some cases because the purpose or function of the supplemental items was arrived at by a number of methods, their accuracy can not be 100% guaranteed.

Apple cautions again that while these zero page locations are not likely to change (even less so, now), all references to these locations should be done with equates so that invokable modules could easily be reassembled if for any reason changes are required at some future time.

Graphics:

# Easter Egg

gary crum/ian barland

```
0 REM *****
1 REM * Easter Egg *
2 REM * *
3 REM * This program draws an n-sided *
4 REM * polygon, stores the vertices *
5 REM * in an array, then connects them. *
6 REM * To use, type the program in, *
7 REM * make sure "/Basic/Bgraf.inv is *
8 REM * on-line, and type "run". *
9 REM *****
10 REM Apple /// version
11 REM by Gary Crum and Ian Barland
12 REM Copyright (c) 1986 ON THREE
110 ON ERR INVOKE"/p/languages/basic/bgraf.
inv"
130 PERFORM grafixmode(%2,%0):OFF ERR
140 PERFORM initgrafix
150 PERFORM fillport
160 PERFORM grafixon
170 OPEN#1, ".grafix"
180 DIM x.vertex(30),y.vertex(30)
190 pi=3.14159
200 xr=279:yr=95
210 n=INT(RND(1)*20+10):i=0
220 PERFORM moveto(%2*xr,%yr)
250 FOR side=0 TO 2.01*pi STEP(2*pi/n)
260 PERFORM lineto(%xr+xr*cos(side),%yr+y
r*sin(side))
270 i=i+1
280 x.vertex(i)=xr+xr*cos(side)
290 y.vertex(i)=yr+yr*sin(side)
300 NEXT side
310 FOR loop1=1 TO n-1
320 FOR loop2=loop1+1 TO n
330 PERFORM moveto(%x.vertex(loop1),%y.
vertex(loop1))
340 PERFORM lineto(%x.vertex(loop2),%y.
vertex(loop2))
350 NEXT loop2
360 NEXT loop1
370 PERFORM moveto(%20,%10)
380 PRINT#1;"Hit any key";
390 GET a$
400 IF a$=CHR$(27) THEN HOME:TEXT:END
410 PERFORM fillport
420 GOTO 210
```

**ON THREE Presents...**

## Reconditioned Apple ///'s with Monitor ///

Available Again!

**256K \$999**

**512K \$1399**

Get 'em while the getting's good!

- for yourself
- for your office
- for a friend
- for a business associate

plus: \$50 shipping and handling

**OF COURSE!**

Backed by *ON THREE's*  
limited 60-day warranty

Only *ON THREE* sells 512K Apple ///'s—the world's finest microcomputer. *ON THREE* provides complete support at no extra cost, plus a full line of *ON THREE* peripherals and sophisticated software.

FBI Apprehends Suspect:

## Piracy Doesn't Pay

Gary L. Penrith, Special Agent in Charge of the San Diego office of the Federal Bureau of Investigation (FBI), announced the arrest by FBI agents of *Raymond Paul Johnson*, age 31, at Waddell's Trailer Park, 1250 Barrett Lake Road, Dulzura, California.

Johnson was arrested for copyright infringement in violation of Title 17, United States Code, Section 506A and

his arrest was based upon a federal warrant issued by United States Magistrate *Edward A. Infante* on January 13, 1986.

The federal warrant charged Johnson with the unauthorized duplication and selling of copyrighted computer software games and business programs.

Additionally, Johnson's residence was searched pursuant to a federal search warrant issued by United States Magistrate *Infante* on January 13, 1986, at San Diego, California.

This search was conducted to obtain additional evidence including computer disks, computer hardware and business records in connection with this copyright infringement.

Johnson was afforded a hearing before a United States Magistrate in San Diego on January 14th. He was released in lieu of a \$5000 personal surety bond.

If convicted, Johnson could be sentenced to a fine of not more than \$250,000.00 or imprisonment for not more than two years or both. 

# Disk of the Month

**What is the ultimate time-saver? Why ON THREE's Disk of the Month diskettes, of course. Why use your precious time typing in ON THREE program listings when they are available on diskette for just \$14.95 (plus \$2 shipping and handling) each?**

**Better yet, mix and match. Any two or more for \$12.50 each (plus \$4 total shipping and handling). Best bet: the works.**

**Now is the time to start your collection of these program-filled diskettes from all issues of ON THREE Magazine. Bulk and group purchase rates are also available, call (805) 644-3514 to inquire about these super savers.**

## DOM #1—Extra Disk Space Plus!

This diskette contains all programs from volume I, nos. 1 and 2 of ON THREE Magazine. Included: Disk Pak1 with a program to give your four additional blocks of space on your data disks, and Disk Pak2, something you can't do without if you are a Pascal user, a convenient and easy way to list the files on a Pascal directory. Plus graphics and sound demos and more.

## DOM #2—Changing Printer Characters

Here is an amazing program you won't want to miss. With it you can print to the Apple Dot Matrix and compatibles such as Imagewriter or ProWriter the same characters that are shown on your video display. Many special fonts, including fancy gothic characters, can enhance your printed output. And, it comes with complete documentation. Also on DOM #2 are the other programs from issue number 3, more graphic demos plus: a program to list files from an Apple II diskette without needing to enter emulation mode.

## DOM #3—Redefining a Keyboard

This disk is jam-packed full with programs that appeared in Volume I, No. 4 of ON THREE, and includes an easy-to-use program that allows you to redefine any or all keys on the Apple /// keyboard. Of particular interest is the ability to reassign the "V" to be the delete key so it can be used on AppleWriter /// and other programs. Also included are all the WPL programs, a disk formatting utility, a graphics sketching tool and still more that we don't have room to list here.

## DOM #4—Emulation Patch

Volume II, No. 1 had so many great programs it took two disks, DOM's 4 and 5, to hold them all. DOM 4 has all of the Pascal programs and the *Apple II Emulation Patch*, a way to use any Apple /// Font in emulation. Also included is the Pascal startup program for *Access ///* that lets you autodial. Another fine utility is a Pascal program and UNIT to permit calculations from within the Pascal environment. Demos haven't been forgotten either with *Radiate Graphics Demo* and *Beatles Music Demo*. To top things off, we have included a number of *Draw ON* pictures you can view with the program on DOM #5.

## DOM #5—Access Draw ON

Here we find the BASIC startup program to autodial from *Access ///*, and *Ben's SUPER Slot Machine*, along with all of the *VisiCalc* and *WPL* programs, and the *Circling Graphics Demo* which will show some of the fantastic images that *Draw ON* can create, plus still more *Draw ON* pictures, along with the *Draw ON ///Picture Demo* which you use to view *Draw ON* pictures.

## DOM #6—BASIC Lister Plus!

Straight from the pages of Vol. II, No. 2 is a program which will give you perfectly formatted listings of Business Basic programs, and a Pascal program to guide and assist you in selecting noises for animation and game programs. Both the *Pascal Noisemaker* and the BASIC lister come with full documentation. We've also tossed in still more *Draw ON* pictures and some new fonts, as well. You can use the *Draw ON* viewer from DOM 5 to see them.

## DOM #7—Heap Good Stuff

From Vol. ///, Nos. 1 and 2 we present a BASIC heap sort routine and demo, *IMAGEHELPER*, a neat graphics utility to simplify graphic image design, and a menu-driven program to pre-select printer codes and parameters.

## DOM #8—Directory Sorting

Here is what you have been waiting for, a complete BASIC and Assembly program to take those old chopped up directories and sort them out in just the order you want. Included also is *Clean.Heads*, a Pascal program which exercises your disk drive at cleaning time and writes a program to remind you when you last cleaned heads, and a simple utility to read a text file and find out what the contents are without having to write a program on the spot.

# ON Pascal ///

dennis cohen

This is intended as the first in a continuing series of articles on Apple /// Pascal and *you*. Many of you are Business Basic programmers who don't want to learn Pascal; however, I am also going to be discussing the Pascal system on the /// and that includes the Filer, Editor, and Assembler, all of which are necessary if you are going to write an invokable module or even install one that you find listed in this (or another) magazine. This first installment is going to be a history lesson and introduction to the upper layers of the Pascal system.

## History

In the beginning, Dr. Ken Bowles of UCSD had a vision. This vision related to implementing a self-contained operating system on a microprocessor (an LSI-11) where students could be taught to program in Pascal. As the years have shown, his vision was a good one as UCSD Pascal has become the "Standard" Pascal for microcomputers.

There are now implementations of UCSD Pascal on a wide variety of machines. While others might disagree, I (and many others) believe that the best implementation is the Apple /// Pascal system. In all other incarnations, the Pascal system is separate from any other OS on the machine, with different disk formats, different directory structures, and different protocols for accessing the BIOS (Basic Input/Output System). With the Apple /// and SOS, Apple made the first major breakthrough—the Pascal system sits "on top of" SOS, using the native operating system's formats and conventions. While doing this, Apple went one step further and maintained the capability to read and write UCSD formatted disks. In this series I will attempt to teach you about the Pascal System and programming in it.

## The Command Line

The first thing you will notice after you boot Pascal on your system (I'm going to assume that you have already used SCP to configure your boot disk) is that there is a funny line across the top of your screen that reads something like:

**Command: F(ile E(dit C(ompile ... ? [A3/1.1]**

This is called a command line. At all levels of the Pascal system you will find lines of this sort, which inform you of the various current options. When UCSD Pascal was first developed, many users did not have video terminals to use with their computer and were restricted to Teletype devices to communicate. The command line was chosen as a simple implementation for menus which would not eat up a lot of paper.

The first thing you should note is that any option is invoked by typing only the character preceding the left parenthesis symbol. This character is also the only one capitalized in the word, but those of you who remember the Apple II will also remember that some terminals display only upper-case characters; therefore, there are two ways to determine the letter which invokes the desired action.

Another thing that you will notice is that some command lines end with a question mark and others do not. This is neither an inconsistency nor a mistake; the question mark is also a valid command. If a question mark appears at the end of a command line, it means that there are more options than could fit on one line and if you enter a "?" additional choices will be displayed.

The only items on a command line which do not specify options are informational. The first thing on a command line tells you at what level of the system you are operating (Command, Filer, Edit, etc). The final item on the line will be something of the form

**[A3/1.1]**

This tells you which version of UCSD Pascal you are running, and which revision it is (the above is Version = Apple3, revision 1.1).

In this column we are going to look at the options available from the Command level (see Table I for a complete list as of revision 1.1).

## F(ile

F will invoke the Filer, which is a very powerful utility in the Pascal System (and the Apple /// in general). You are already familiar with many of its capabilities from the System Utilities program. Except for formatting disks, it includes all of the capabilities of both the "File Handling Commands" and "Device Handling Commands" in System Utilities. You are not limited to using these utilities on SOS diskettes, but can use them on UCSD (Apple II Pascal) disks as well (except for making a subdirectory).

There are also a few utilities here which are specific to UCSD diskettes. The most notable of these are K(runch and eX(amine. If you attempt to use either of these commands with a SOS diskette, the Filer will disallow the operation and tell you that such an action is only valid with an Apple II disk.

The Filer also provides a few additional facilities. Most of these are related to the "Workfile," a file (or pair of files) which is the default file used by other system programs such as the Editor and Compiler. There are also two directory listing commands, L(dir and E(xt-dir, which list respectively the current directory or the current directory and all of its subdirectories. On the Apple II (or standard UCSD) the difference between these commands was the amount of information printed about each file.

Another handy command is Z(ero). It does a "Remove All" on the files in the specified directory. There are also various "wildcard" capabilities built into the Filer which allow you to operate on groups of files rather than individual files (similar to the "windowing" selection capability of System Utilities without the graphics).

### E(dit)

The UCSD Editor (usually referred to as SOE for Screen-Oriented Editor) has also been extended by Apple for use on the ///. SOE is an extremely versatile and easy to use full-screen editor which operates in two modes, program-entry and word-processing. SOE defaults to program-entry mode upon invocation unless the last time you edited the file in question you set the mode to word-processing. In other words, SOE remembers from invocation to invocation what mode to use on a particular file. SOE achieves this through the use of a two-block long "header page" on each TextFile. This header page contains "environment" information

about the file such as the mode, margin settings, and markers. Each textfile may have up to 10 named markers set so that you may move around conveniently in the file.

SOE will edit both TextFiles and AsciiFiles (in Business Basic they are PasTxt and Text, respectively). There are a number of differences between the two file types, and your application should determine which type of file you use. Certain programs both expect and require that the file be of a specific type. For example, the Compiler and the Assembler require that the file be a Pascal Text File, whereas Business Basic programs work with ASCII Files. To you, the user, there is no real difference between the two file types. Internally, however, the differences are clear. An ASCII file conforms to the "standard" Pascal definition of a file of type TEXT. It is a sequence of characters separated into lines by EndOfLine characters. A Pascal Text file is "optimized" for SOE and program-entry. In addition to the two-block header, the physical file is

always broken into two-block "pages." That is why a directory listing always shows PasTxt files as being an even number of blocks in length, and at least four blocks long.

Some of the restrictions on a Text file are that a physical line may not cross a page boundary (the empty space at the end of a page is filled with ASCII Nulls [Character 0]) and that leading blanks on a line are compressed into what is referred to as a DLE (Data Link Escape) code which consists of the DLE character (ASCII 16) and a byte which is the number of spaces plus 32. While this might seem wasteful at first, you should know that Pascal programmers are encouraged to indent their code to display its logical structure and in large programs the space savings due to DLE compression are substantial. Because of this, I always do my programming with Text files (which the Compiler and Assembler require) but generally do my documentation in ASCII files (unless I'm using Word Juggler). The only inconvenience here is that I must always set the editing environment to word-processing when I have an ASCII file since the header block is not there for SOE to see.

I am planning to devote my next column to SOE and the Filer, so I will leave further discussion until that time.

### C(ompile)

This command does just what its name implies, it invokes the System's Compiler (the code file SYSTEM.COMPILER) which, in an Apple-supplied system, is the Pascal compiler (Volition Systems sells a Modula-2 system for the /// which is a superset of the Pascal system), and attempts to translate the Pascal language statements it is directed to read into a compact form called p-code (which is actually the machine code of a hypothetical piece of hardware). The file SYSTEM.PASCAL in the root directory is actually a piece of software that "emulates" the p-code machine. The compiler looks first to see if there is a workfile to compile. If there is not, it asks you for the name of the input file, then for the name of the output code file. If you are running

E(dit)	Invokes the codefile named SYSTEM.EDITOR
R(un)	Compiles (if necessary), Links (if necessary) and executes the current workfile
F(ile)	Invokes the codefile named SYSTEM.FILER
C(omp)	Invokes the codefile named SYSTEM.COMPILER after initializing various System Globals (this could be Pascal, FORTRAN, or Modula-2) Compiles workfile if one present
L(ink)	Invokes the codefile named SYSTEM.LINKER - combines p-code files with other. p-code files or with assembled, native-code routines.
X(ecute)	Invokes the codefile you specify in response to a prompt. Version 1.2 gives a default of the last (non-system) codefile executed.
D(ebug)	Invokes the Debugger (if you have one), which allows you to execute a program as a subtask and trace its execution in p-code. Not documented in the manuals.
A(ssem)	Invokes the codefile SYSTEM.ASSMBLER. Assembles workfile if one is present.
-----	
U(ser Restart)	Reinvokes the last codefile executed, including System files
I(nitialize)	Reinitialize the Pascal System - almost a "warm boot." Prefix information is maintained
O(ptions)	Change System Level options - currently, graphics space allocation, residency of Apple ][ Pascal disk routines, & filename display format
H(alt)	A full "warm boot" - SOS.KERNEL info is maintained; however, SOS.INTERP is reexecuted beginning with its initialization routines.
M(ake Exec)	Records keystrokes to an Ascii file up to a user specified terminator
Q(uit)	Exit the Pascal System - displays message "Insert System Disk and Reboot"

Table 1:  
Pascal System command line options

(although all are available from the command line, the dashed line separates them into the two groups which may be visible, a "?" will display the other section of the command line)

revision 1.1 or higher, it will then ask for the name of the file to which you would like the compiler listing to be sent.

### L(ink)

This option is used to link together separately compiled and assembled code segments into an executable code file (or library unit).

### A(ssemble)

The assembler is a very important part of the Apple /// Pascal system. It is based on The Last Assembler (TLA) from Waterloo University. TLA is also the only supported assembler on the Apple /// and is not only used to generate specialized code for Pascal programs, but also SOS Interp files such as Business Basic, AppleWriter and invokable modules in Business Basic. For the above reasons, I believe that anyone who wants to do any serious programming on the Apple /// should become familiar with the Pascal system. It provides your only convenient interface to the underlying machine.

### X(ecute)

This is the equivalent of the RUN command in BASIC. You will be prompted for the name of the program you wish to run and then the system will attempt to execute the named program. Many of the Pascal system utilities are not available as system options and must be executed (e.g. LIBRARY.CODE and AIIFORMAT.CODE).

### R(un)

This is a slightly more specialized version of X(ecute). It applies only to the current workfile. If a code file exists for the workfile, it is executed. If not, the compiler will be invoked to compile the text workfile. Assuming that you had to compile and were successful, the system will attempt to execute the resulting codefile, linking if necessary from SYSTEM.LIBRARY.

### U(ser\_Restart)

With this command you re-execute the last program (or system option) executed (or selected).

### I(nitialize)

Initialize is used to reinitialize the file SYSTEM.PASCAL. It does not alter the Prefix, but will reinitialize memory allocation such as the amount of space allocated for graphics.

### H(alt)

This is what might be termed as a "warm boot" by some people. When you enter Halt, the Pascal system terminates execution and transfers control to the Pascal interpreter (SOS.INTERP).

SOS.INTERP starts again from scratch, allocating drivers to Pascal units, opening .CONSOLE for I/O, and then reloads the Pascal system and all of its parameters.

### O(options)

This selection allows you to select certain system options such as the residency of Apple ][ Pascal disk routines and the amount of memory space reserved for graphics.

### M(ake\_Exec)

You use this command to create Exec files. Exec files are AsciiFiles that contain a sequence of steps that you wish to execute repeatedly. For those of you with large computer experience, this is analogous in most respects to a Job Stream or Run Stream.

An example of where you might use an Exec file follows:

Suppose you have a program that generates data files for a graphics program, but you have insufficient memory to execute the program if graphics space is allocated. To work around this you could build an Exec file that looks like this:

```
Xdatagen —execute the data
            generator
O          —select Options
A          —option A (graphics
            space)
D          —suboption D (32 K)
Q          —Quit (system will
            reinitialize)
Xgraphprog—execute the plotting
            program
```

Then you would merely execute the Exec file when you wanted to perform this sequence.

### Q(uit)

This option is new with revision 1.1. It simply issues a SOS Terminate call (after verifying that you really want to quit). When this is done, you will see the screen go blank (and possibly inverse) and a message appear in 40-column mode telling you to insert your system disk and reboot. The Quit option is exactly the same as the BYE command in Business Basic version 1.2.

### MES

As a holdover from its beginnings in UCSD Pascal, the Pascal system looks for certain endings on filenames at various times. Both the Editor and Compiler, for example, expect that input filenames end in .TEXT; therefore, you need not specify the extension when you enter the name because the Editor (Compiler) will add .TEXT to the filename if you did not. This can lead to difficulty if your filename does not end in .TEXT, for example a QuickFile report that was written to disk (these end .ASCII). To get around this, you may end a filename with a period, which is a signal to the system not to add .TEXT (and to ignore the ending period). A similar situation arises when you go to execute a program. If you do not end the name with either a period or .CODE, the system will add .CODE to the name when searching for it on disk. On the other hand, the Filer is very literal about the names you give it.

### Conclusion

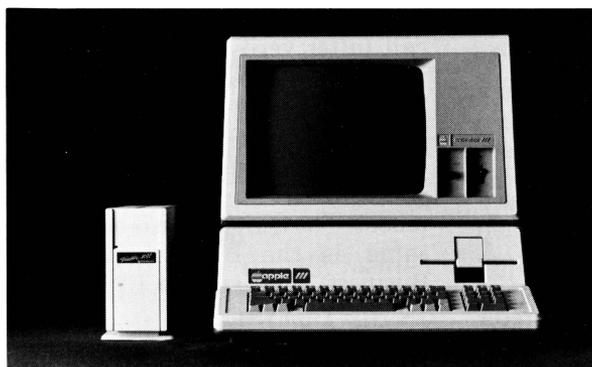
This has not been an in-depth discussion of the Pascal System on the Apple ///, but it should give you an idea of what is happening. The manuals that come with the system tell you all of this (and more) and are pretty well-written. In this respect we are better off than our brethren on the ]['s, since Apple has learned a lot about writing manuals since the Apple ][ version 1.1 release.

(One of the things that folks at Apple do is to rename SYSTEM.FILER to be FILER.CODE and then copy the SYSTEM.STARTUP file from the System Utilities disk to be SYSTEM.FILER, and it is something that you might want to try.)



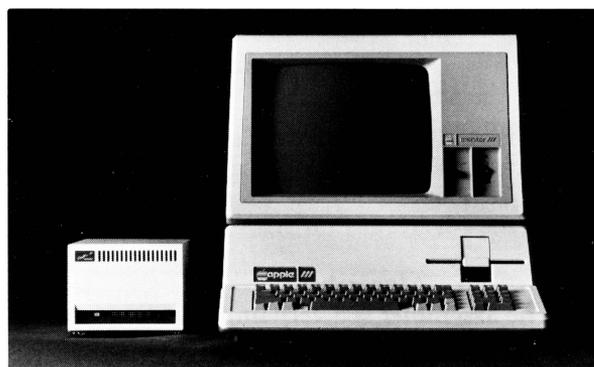
# ON THREE Presents...

## a new line of high capacity Apple /// disk drives



10-20 MegaBytes

or



34 MegaBytes

### **ON THREE has exciting news for you! A brand new line of low-priced hard disk drives for the Apple ///.**

10-, 20- or even 34-MegaBytes (million characters) of very fast hard disk storage can be yours, priced so low you can't pass them up! These drives will allow you to consolidate all your files on a single disk and reduce the time you waste searching through stacks of floppies.

Combined with our **Selector /// Program Switching Utility**, you can place all (see the Selector ad) of your programs on a hard disk and put your floppy disks away forever. Think of how convenient it will be to be able to run any program from your hard disk—in just seconds.

All our hard drives are manufactured by Xebec—A leading manufacturer of hard disks for the Apple II. They come with a full one year parts and labor warranty, another mark of **ON THREE quality**.

#### **Sider 10—Sider 20**

You may have heard of the Sider 10 and Sider 20 for the Apple II. We have modified these drives to work in the Apple ///. They come complete with interface card, cabling, documentation and driver diskette, ready to run on your Apple ///.

The Sider 10 and 20 are attractively styled hard disk drives with a unique daisy-chain option that allows you to attach a second drive to the back of the first, just in case you ever outgrow the 20808 blocks on the Sider 10 or the 41616 blocks on the Sider 20.

Priced at only \$999\* for the Sider 10 and \$1299\* for the Sider 20, these drives are the best hard disk value on the market today!

Added Bonus: How would you like to be able to backup your entire hard disk in a matter of minutes? We will shortly be shipping the **B-Sider**, a high speed, low cost tape backup to attach to the Sider 10 or Sider 20. Call for pricing and availability.

**Xebec 9730** The Xebec 9730 is the Sider's big brother. With a capacity of 69,632 blocks (34-MegaBytes), it is one of the fastest disk drives on the market. If you have very large disk storage needs, the 9730 is the drive for you. Like the Sider drives, the 9730 comes with everything you need to get it running on your Apple ///.

The 9730 is only \$1999\* and is available right now from **ON THREE**.

#### **A Note On Large Hard Drives:**

Since the Apple /// can only work with disk volumes up to 16-MegaBytes in size, each of our large hard drives (Sider 20 and 9730) have been split into two or more sections. Our 20-MegaByte disk is partitioned into a 16-MegaByte volume and a four-MegaByte volume. The 34-MegaByte disk is partitioned into two 16-MegaByte volumes and one two-MegaByte volume. Partitioning simply means you will have two or three disk volumes in one drive box.

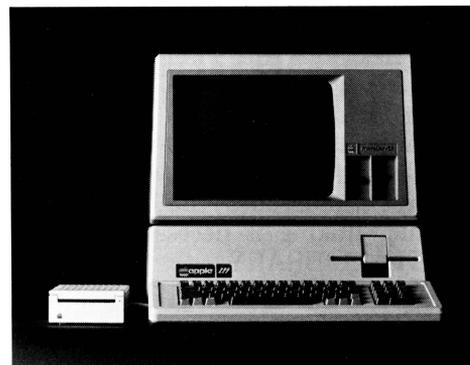
#### **UniDisk ///.5 800K Micro-Floppy**

The **UniDisk ///.5** is an 800K 3.5 inch disk drive for the Apple ///. If you have a hard disk and hate to do backups, the UniDisk ///.5 is the ideal solution. You can backup an entire ProFile with just seven UniDisk micro-floppies. Faster than a normal disk drive, the UniDisk ///.5 is a great time-saver.

Even if you don't have a hard disk, wouldn't it be great to get rid of your regular floppy disks? The new 3.5 inch disks are great! They fit in purses, briefcases, and even shirt pockets much easier than standard 5¼ inch disks. With a hard plastic shell, they can take far more punishment than the easily destructible 5¼ inch diskettes. You can also use your diskettes on UniDisk-equipped Apple //e and //c computers. Since these same 3.5 inch disks are used on the Macintosh, a utility will be coming soon to transfer files to and from the Mac.

The **ON THREE UniDisk ///.5** comes complete and ready to run on an Apple ///, including drive, interface card, cabling, documentation and driver disk. A truly great buy, priced at only \$499\*.

If you already have a UniDisk for your Apple //e, the driver and diskette are available separately at \$50 plus shipping.



\*Shipping charges extra: Sider 10, Sider 20 and Xebec 9730: \$35. UniDisk ///.5: \$10. UniDisk ///.5 documentation and driver disk: \$3.

**To order, call (805) 644-3514 or write:**

ON THREE, Inc.  
Attn: Order Dept.  
Post Office Box 3825  
Ventura, CA 93003

We accept Visa, Mastercard, and American Express. There is a 3% surcharge on orders charged to American Express. California residents add 6% sales tax (on products only).

# Installing the Xebec 9730

john r. sollman

When I first purchased my Apple /// system in December of 1982, my ProFile hard disk seemed big enough to hold the world. After doing a few jobs and storing data on the hard disk, however, it gradually became apparent that space is something you are always running out of. As my system continued to grow and programs were added to the hard disk, more and more demand was placed upon that meager five megabytes of storage capacity.

In order to gain additional disk storage capacity, a MicroSci A143 was acquired. Soon another was added to the chain. Ultimately, my system had grown to the point where most of my worksheets and templates were on floppies, and the ProFile was being used mostly as a boot device and a spooler. My large data bases had to be copied to the ProFile when being worked on, and then copied back to the floppies and deleted from the ProFile to make room for other applications. Enough was enough. The time had finally come to look for some additional hard disk storage capacity.

By coincidence, *ON THREE* Magazine had just come out with a number of new products for the Apple ///, including the Trustor 10 and 30 hard drives, now known as the Sider 10 and Xebec 9730 respectively. After evaluating my anticipated future needs for data and file storage capacity, I decided upon the Trustor 30. Because the drives are identical, all further references will be to the Xebec 9730.

The new hard disk was to be connected to a system which already consisted of a 512K Apple ///, a ProFile hard disk, one Disk ///, two MicroSci A143 disk drives, Catalyst 2.0, Pascal, BASIC, the Quark Discourse spooler, several typical application and data base programs, and, of course, all those drivers which go with Catalyst and MicroSci.

My initial plan was to add the hard disk to my system and keep the old ProFile, which was to continue to serve as the boot device. The Xebec 9730 was to be reserved for on-line file and data storage. Plans, however, have a way of changing, and mine did.

The Xebec 9730 is equipped with an internal formatter and a smart internal controller which gives it an extremely fast seek time. This also accounts for the extremely small driver files needed to run this device. While SOS is limited to about 32 megabytes of disk storage on one volume, the Pascal file limitation is effectively one-half of that, or 16 megabytes (approximately 32,767 blocks of data). The Xebec 9730, as delivered for use with the Apple ///, is divided into three volumes. Available disk space on the first two volumes is 32,766 blocks each; 4096 blocks are available on the third volume. Originally some 800 blocks were spared out to replace any bad blocks which might occur, but *ON THREE's* experience has shown that this is not necessary.

The drive comes with a controller card and all necessary cabling. The unit is shipped with the cabling attached to the back of the drive. The cabling consists of a heavy segment about 30 inches long, and a lighter segment about 12 inches long terminated by a female connector. Also included is a concisely written 18-page user's guide which contains complete and understandable instructions for installation of the device. Included also is an installation diskette containing a copy of the Apple System Utilities Program and a driver code file module. Conspicuous by its absence is the slick three-color brochure which devotes the first several pages to telling you how smart you were to buy this particular product. *ON THREE*

advises that this is one of the frills done without in order to give the customer the best bang for the buck.

The device driver module and the System Utilities diskette are configured for Slot 4 in the Apple ///. If the controller card is to be installed in any other slot, the slot assignment must be modified on the System Utilities installation disk before installation of the software can begin.

One word of caution before starting installation of the controller card. Do not be misguided by the length of the cable when trying to figure out where to put the Xebec 9730. As with the ProFile, effective length of the cable will be only about 36 inches. In addition, the cabling is quite heavy and might not lay well in certain situations. The lighter part of the cable passes through the opening in the back of the Apple /// and plugs into the *front* of the controller card.

Installation of the drive is simple and straightforward. The controller card may be installed in any slot inside the Apple ///. The cable must be passed through the opening and plugged into the card before the card is inserted into the expansion slot. Care must be taken not to bend the pins on the male (controller card) end of the connector. A bit of gentle force is needed to get the cable connector to seat firmly to the controller card. None of the pins must be showing between the card and the cable. If they are, it is not plugged in all the way. Care must be taken where the force is applied to seat the cable, since it is possible to bend the card and damage the circuitry. The cable must be neatly routed behind the controller card before the card is inserted into the slot, so that there is no slack, and no strain in either the cable or the card.

The drive must next be formatted. Two procedures are necessary. First, there is a "hard format," in which the device's internal formatter is formatted. This requires a device driver called .FORMATTER, which is a part of the driver module provided on the installation disk. Next, a "soft format" is performed on each of the three volumes. The *ON THREE* User Guide recommends that the volumes be named /ProFile, /Pro2 and /Pro3. Finally, each volume must be verified.

The driver module on the installation diskette, called /XEBEC.30, occupies only seven blocks. The module consists of four chained drivers called, .PROFILE, .PROFILE2, .PROFILE3, and .FORMATTER. As delivered, the .FORMATTER driver is inactive, and the module is configured for slot 4. If the drive is installed in a slot other than 4, the slot configuration should be changed before the driver module is installed on any of your boot diskettes. This will ensure that you do not forget to make this change on one of your diskettes. The .FORMATTER driver should be left inactive. If for any reason it is necessary to perform another "hard format" on the Xebec 9730, the Utilities program on the installation diskette should be used.

When adding the driver modules to your SOS.DRIVER files, it is recommended that Version 1.2 of System Utilities be used if available. Some of us, especially those of us who use Catalyst 2.0, a spooler and MicroSci disk drives, are blessed with some enormous driver files. Version 1.2 is said to handle large driver files better than Version 1.1, which is the version contained on the installation diskette. Anyone who has installed the *ON THREE 512K Upgrade* should have a copy of Version 1.2.

Those who use the Catalyst program selector must also change the Driver Configuration Block of the .CATALYST driver. The instructions for installation contain the values for each byte of the DCB. The first byte, of course, must be set to the slot number in which the drive is installed. Changing DCB values should be a simple matter for any long-time

Apple /// user. If you use the Quark Discourse spooler, there are a few more changes to be made. The second six bytes, bytes 06 through 0B, of the .SPOOL driver must be set identical to the first six bytes of the .CATALYST DCB. These changes to the .SPOOL DCB were not included in the instruction sheet accompanying my drive, but will be included with drives shipped in the future.

I initially configured only my System Utilities Stage 1 boot diskette. This was to permit me to resolve any problems encountered before configuring my Catalyst and other program or copy-protected diskettes.

The first problem encountered after renaming the old ProFile driver (.PROFILE4) and the volume (/Pro4) (the volume must be renamed before adding the new driver module) and adding the new Xebec 9730 drivers was that I could no longer communicate with the old ProFile while in System Utilities. This problem was solved by reordering the appearance of drivers in the SOS.DRIVER file. A driver module was made consisting of 1) the Xebec 9730 drivers, 2) the MicroSci drivers, and 3) the MicroSci formatter drivers with two turned off. The corresponding drivers were deleted from the System Utilities boot diskette and replaced by the module just created. This put the old ProFile within reach and enabled me to transfer the files. (Thanks to Bob Consorti for his help on this one.) This module was later used for configuring my other boot diskettes.

While proceeding through the installation, it was decided to configure my Catalyst for the Xebec 9730 hard disk instead of the old ProFile as originally planned. I am glad I did.

The old ProFile contained about 4.5 megabytes of programs and data. Using the wild card, all the files were transferred to the Xebec 9730 in about 27 minutes. Most of the time was spent reading from the old ProFile. Following data transfer came the moment of truth—booting up time.

The reconfigured Catalyst worked perfectly with the Xebec, as did the reconfigured spooler. The only

problems encountered were in accessing the old ProFile from Pascal programs. Catalyst 2.0 has a driver module called .QUARKCOM which is chained to six drivers named .UNUSED. These are used for dynamic loading of certain drivers, such as .AUDIO or .GRAFIX. The presence of all the drivers associated with Catalyst and MicroSci places the old ProFile out of reach of Pascal programs, including System Utilities. The old ProFile remains available to programs like Word Juggler and VisiCalc. File copies or backups from the old ProFile would have to be made out of the Catalyst environment. Those who use *Selector ///* instead of Catalyst should not experience this problem, since *Selector ///* does not require the use of special device drivers.

Several attempts were made to bring the old ProFile within reach of Pascal programs in the Catalyst environment. The old ProFile driver was deleted from the boot diskette, copied to the SOS.DRIVER file in the Catalyst subdirectory, and dynamically loaded. It worked for all but the Pascal programs. It seems that the two .UNUSED drivers designed for dynamically loaded block devices are too far down the chain to appear in the Pascal Volumes command. Perhaps later on, some fiddling with the DCB of the .UNUSED drivers to shift their positions within the chain might bring the old ProFile within reach of most Pascal programs. However, the Xebec 9730 seems to have so much capacity that I might just sell the old ProFile and eliminate that problem altogether. (Reordering of the position of the .UNUSED drivers might affect the dynamic loading of the .GRAFIX driver to a Pascal program which needs it.)

The division of volumes on the 9730 seems to be quite fortuitous. Since I am the Adjutant of my American Legion post and the bulletin editor of my Rotary club, I copied the files I use in connection with these jobs to .PROFILE3. The size is just right. .PROFILE contains the Catalyst subdirectory, Pascal, BASIC, and other software. The remaining space will be used for some large relational data base applications

under development. Business files, work aids, templates and worksheets consume about 11,000 blocks on .PROFILE2, with lots of room for expansion.

At the time this article was written, the Xebec had been in use for only a short time. The first thing which impresses me is its lightning-fast speed. Word Juggler documents are loaded and stored in a flash. Advanced VisiCalc loads a tiny bit faster, but not much. Within the VisiCalc program, however, commands requiring program data from the disk are executed noticeably faster. The disk requires only about 20 seconds to come up to speed when first turned on. The device's greatest advantage is the convenience of not having to search through a bunch of floppy disks to find templates or worksheets regularly used in work.

Since getting the drive up and running, I have booted all of my programs. Performance in all programs is significantly speeded where disk I/O is concerned. In my Key-stroke data bases, it is most convenient to have them all on the hard disk. No longer do I have to copy a data file to the hard disk, and then delete it in order to make room for other work. One program, Omnis3, which I pur-

chased to create several relational data bases, will not communicate with .PROFILE3. This is because of the peculiar manner in which Omnis3 selects the volume for data files and libraries. In my particular system, Pascal assigns a unit number greater than 12 to .PROFILE3. Omnis3 apparently has access only to volumes with a Pascal unit number of 12 or less. Possibly this could be cured by shutting off a MicroSci driver.

Limitations discussed above are more associated with the Apple /// operating system or Pascal than with the Xebec 9730. SOS has a limitation of 12 block devices, and Pascal is limited to 6. This is a problem I am told *ON THREE* is working on. The presence of all the block device drivers associated with MicroSci and Catalyst 2.0 quickly use up the Pascal limitation. If one were to consider installing a RAM disk or some similar enhancement, some concessions would obviously have to be made. Those who use *Selector* /// instead of Catalyst would encounter fewer problems related to the number of block device drivers in the system.

Overall, I find that the device is quiet, fast, easy to install and a pleasure to use. The drive provides more than ample disk space on its

three volumes for most program and data base applications. It comes with controller card, necessary cabling already wired, an 18-page installation and user guide, and an installation diskette. Several enhancements are already in the works. The installation and user guide contains several minor errors on page 5. Under a section called "Special Note," it refers to driver names of .PRO2 and .PRO3. Correct driver names are .PROFILE2 and .PROFILE3. Other than this, no fault could be found with the documentation. It is clear and concise, which is certainly refreshing when one compares it with the generally horrible writing encountered in the typical computer manual. Most refreshing is the willingness of Bob Consorti and his staff to work with the customer on any problems which might arise.

*Editor's note: Subsequent to the time this article was prepared, the above errors have been corrected in a manual rewrite.*

The Xebec 9730 hard disk drive is warranted by Xebec for 12 months from date of purchase. *ON THREE* warrants the software media for 90 days. Available from *ON THREE* at \$1,999, the Xebec 9730 hard disk is indeed a bargain. 

## ***ON THREE Presents . . .***

### **• Disk of the Month**

- No need to spend your precious hours typing in long and complex program listings**
- Complete BASIC programs, assembler source and object files from each issue of ON THREE**

**\$14.95** each plus \$2 s/h

two for **\$12.50** each plus \$4 s/h

### **• Numbers 7 & 8 now available**

- No. 7. Contains IMAGEHELPER, a BASIC program to aid in designing graphics images, a heap sorting routine, and a program to preset printer codes.
- No. 8. Contains BASIC and Assembly programs to sort directories, a Pascal program to clean drive heads, and a text file reader.

## **DE CLASSIFIEDS**

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

**Subscriber Discounts:** .50 per word, \$12.50 minimum, subject to the following restrictions:

- Non-commercial ads only
- No items valued over \$100

**SMARTTAX** Prepare federal and California state personal income tax returns simultaneously. See results side by side on your Apple /// system. Prints directly onto standard IRS and California state forms: 1040,A,B,C,D,E,G,SE,W,2441,540,A,B,CEF,D,G. Menu-driven with many additional features. Ideal for professional or personal use. \$89.95. To order, call (415) 454-4357. Officeware Data Systems, 351 Olema Road #15, Fairfax, CA 94930.

# How many dollars have you invested in your Apple ///?

Answer this questionnaire and find out how to extend your computer's life:

- Do you vacuum your office or home less than twice a day?  YES  NO
- Are there smokers in your office or household?  YES  NO
- Are there children in your household?  YES  NO
- Is the computer located in a family room or other high traffic area?  YES  NO

If you answered "yes" to one or more of the above questions, then you need to protect your investment with an *ON THREE* Dust Cover.

**Dust Cover for Apple /// and  
Monitor ///  
\$11.95**

**Dust Cover for Profile,  
Apple /// and Monitor ///  
\$12.95**

**Plus \$2 Shipping and Handling each**

ON THREE (805) 644-3514  
P.O. Box 3825  
Ventura, CA 93006

Send me \_\_\_\_\_ Apple /// dust covers  
Send me \_\_\_\_\_ Apple ///-Profile dust covers

Name \_\_\_\_\_  
Address \_\_\_\_\_  
City \_\_\_\_\_  
State \_\_\_\_\_ Zip \_\_\_\_\_ Phone \_\_\_\_\_

enclosed \$ \_\_\_\_\_  
 M/C  Visa  AE\*  
number \_\_\_\_\_ exp. date \_\_\_\_\_  
Signature \_\_\_\_\_

\*3% surcharge on American Express. California residents add 6% sales tax.

## Hurrah for ON THREE!

ON THREE is America's leading Apple /// support group and independent producer of quality software and hardware products, and ON THREE magazine, America's leading Apple /// magazine is the official publication of ON THREE.

ON THREE magazine contains enlightening articles and programs about Pascal, BASIC, and assembler; technical hints, reviews, material for the novice and the advanced programmer, and in addition, you will always find news of exciting new and current ON THREE products, user group listings and hot line consultants, plus the popular and informative "Three Questions" readers' forum.

*Chances are your ON THREE subscription is about to run out. If so, renew now and don't miss out on any of the interesting articles in future issues nor announcements of new and sophisticated ON THREE products. ON THREE Magazine, your bible of Apple /// information. Twelve issues for just \$40.00, back issues available at \$5.00, postpaid.*

ON THREE (805) 644-3514  
P.O. Box 3825, Ventura, CA 93006

Name \_\_\_\_\_  
Address \_\_\_\_\_  
City \_\_\_\_\_  
State \_\_\_\_\_ Zip \_\_\_\_\_ Phone \_\_\_\_\_

enclosed \$ \_\_\_\_\_  
 M/C  Visa  AE\*  NEW  RENEWAL  
number \_\_\_\_\_ exp. date \_\_\_\_\_  
Signature \_\_\_\_\_

\*3% surcharge on American Express.

ON THREE presents...

## The Uncopyprotect Driver **\$19.95** plus \$2 s/h

ON THREE has not changed its position regarding duplicating copyrighted programs for profit or to give away, but since many Apple /// software products are no longer supported, owners of AppleWriter ///, VisiCalc, and VisiCalc Advanced Version are facing the problem of what to do when a diskette "crashes." After much consideration we decided to proceed with a product to solve that problem. *The Uncopyprotect Driver* will allow you to make back up floppies of the above programs. For the first time, you can put your master disk in a safe place and boot on the duplicate.

Economically priced at only \$19.95 plus \$2.00 shipping and handling, the *Uncopyprotect Driver* comes with full documentation and will work with *Selector ///* so you will no longer require a "key" diskette. The *Uncopyprotect Driver* is sold for legitimate *Archival* purposes only. *ON THREE does not condone* and will not condone duplicating a disk for any other purpose.

# Three Questions

---

---

## Softcard Incompatible

I am interested in buying a church data management program (I am a church treasurer) that is written in CP/M. I bought the demo to see how it would work. When I run the demo the only response I get is "Bdos error on A: Bad Sector" and I cannot read the directory. I am using an Apple /// with 256K and a Softcard ///.

I called the person I bought the demo from and he asked if I could find a place where I can convert the CP/M format he used (TRS-80) to one that is usable by the Softcard ///. My question is; is this translation possible, and if so, how? Included on the demo is a file called RUN.COM for the Apple ///. (I tried to run this demo in the Apple /// native mode with no success).

I am interested in your Selector ///, but do not know how to use a switching utility, or what is needed to use it.

Raithby R. Husted  
San Jose, CA

P.S. Perhaps you know of a church data management program that is written to run on the Apple ///. My main interest is in the accounting for church funds and keeping bank records.

*I am not at all sure if the conversion you need is possible. I am going to have to throw this up to our readers. The problem is in disk formats. Yours is not the same as his, thus you can't get to the program. Even after you get the program on a disk that the Apple /// can read, you may not be able to get it to run due to idiosyncrasies of the Apple /// Softcard system.*

*Regarding the Selector ///, it's very easy to use a program switching utility—that's the whole purpose—to make it easier for you. Right now you are probably booting up your machine for each different program you want to use. With a program switching utility such as Selector, you boot up the system just once each day. After you boot it up, you are presented with a menu that lists all of the programs you can select.*

*You simply choose the one you want and press RETURN. In a few moments the program appears on your screen, just as if you had booted it up from floppy disk. You can use the program just as you have in the past. The only thing that is different is when you quit the program. In the past you probably received the message "INSERT SYSTEM DISK AND REBOOT". With the Selector you will go back to the menu, where you can select another program.*

*The Selector allows you to run most any program from a central menu. The process of 'Running' the program requires that the program be placed on a hard disk system (or a large capacity floppy). The Selector will place most every program on the hard disk for you. All you really need to do is select it from the menu.*

*Using Selector with a hard disk system such as our Sider 10, will allow you to consolidate all of your programs and data. With a single boot disk (the Selector Start-up Disk),*

*you can use all of those programs. Since all of the program are 'Run' from the hard disk, it will take only a second or two to get to any program. Compared to the minute or two it takes to boot some programs, Selector can be a real time-saver for you.*

*One note of caution: Selector allows all native mode Apple /// program to be used from the hard disk. CP/M programs can not be run from Selector. Apple ][ Emulation mode can be selected from Selector, but the Selector does not allow Apple ][ programs to be placed on or run from the hard disk.*

*It may be that a spreadsheet program such as VisiCalc would serve to handle your needs for distributing church funds and handling bank accounts.*

## Dual Power Supply

Dear ON THREE:

I am in the market for a 110 volt power supply for my Apple /// as my Apple is configured for 220 volts. The same applies to my Monitor /// and my daisy wheel printer. My printer is a Brother HR-15XL. Are you able to provide any help? Will be looking forward to a quick reply.

Bill Sharp  
Jonesboro, AR

*Well first of all, ON THREE can furnish a replacement power supply at \$75. However, as far as the Apple /// itself is concerned, it should not be necessary, since the standard Apple /// power supply has a jumper plug for 230 volts/115 volts located on the rear most portion of the power supply board, right in front of the AC plug and power switch. It is clearly marked.*

*As for the other equipment, we cannot advise you, but the most likely solution would be an inexpensive 110/220 volt transformer.*

## Dvorak Keycaps

Dear Val,

I have modified my keyboard to the Dvorak format and changed all my software except my emulator disk. My question is—are the keycaps available for the Dvorak setup? My keys are hills and valleys.

Ken Gerler  
Bel Nor, MO

*The best answer we can provide somewhat skirts the issue. To the best of our knowledge, there are no Dvorak caps available for the Apple ///. However, the Apple /// plus keyboard and that of the Apple //e are identical, and there are sets available for the //e. It is also quite possible that the //e Dvorak set might fit the Apple ///, but we strongly suggest checking out all of the keys, particularly special keys like [ \ ] or [ ~ ] before committing any cash. Please let us know how you make out.*

## State of the Art???

Gentlemen:

I have an Apple /// (128K upgraded to 256K), an Apple ProFile, three 140K disk drives and an OkiData Microline 82A printer. I have had a service contract on this equipment ever since I bought it and have never had a problem with the computer, only the dealers. The first charged \$850 and went out of business before the contract expired. I looked around and found another Apple dealer 20 miles away and got a service contract for \$1200 and he closed shop before the contract expired. Presently I have no service contract and can only pray that no problems occur.

I have *State Of The Art* Inventory and General Ledger that was installed on the ProFile by the first dealer. I back it up once a month but I have doubts that my backup will put copy protected software back on the ProFile. I also have Apple Mail List Manager that I keep on diskettes. I have other software but these two are the only ones I worry about.

The ideal solution would be an inventory and general ledger not copy protected and a mailing list manager that could be installed (by me) on a hard disk. I would like to have some of the items you have for sale but feel they would not help me a whole lot. I would be grateful for any suggestions you might have.

This is the only computer I have ever owned and I haven't seen anything better. I would rather have this one than a Mac or IBM. I suppose you have been getting "Dear John" letters like this from many people but any help you can give me will be greatly appreciated.

Raymond H. Fugere  
Dartmouth, MA

*It sounds to me as if you have been taken by a few different dealers. My first suggestion is to take legal action against whoever. If the two dealers did not complete the contract they should be bound to give you a pro-rata refund. Also, if you bought an Apple service contract (AppleCare), any Apple dealer is obligated to give you warranty service. If they go out of business, it's just like you moved to a new location. Your contract is valid and they owe you however many months more of service.*

*Because the values you list are so high, I doubt if you had AppleCare. It sounds more like a contract with the individual dealers. In this case, if no legal action is possible, you're out of luck. In any event, unless they offered on-site replacement 24 hours a day (and kept in stock a complete back-up system), you were taken by both dealers. For your equipment you should have been charged no more than a couple of hundred dollars a year. For the money you have spent you could have purchased a duplicate system as a backup and had it available for supplemental use. This possibility still exists.*

*Regarding your backup situation, once a file is placed on the hard disk it is not copy-protected. It may require a copy-protected program on a floppy in order to run, but it can not be copy-protected on the hard disk. Your current backup routine will be just fine as long as your State Of The Art floppy disk does not fail. If your hard disk ever fails, you can use your backup copies to make the program work off of the replacement hard disk. Your fears about Mail List Manager are unfounded as it is not a copy-protected program.*

*There is one note of caution about the State Of The Art program. I can't recommend it for the following reason: The program locks itself to the size hard disk you are using. When the program was installed on the hard disk by the State Of The Art mapping utility it also placed information*

*on the size of the hard disk in certain parameter files on the hard disk. If you ever want to get a larger hard disk, you will have to buy another copy of the program. This tidbit of information is not in any of the manuals from State Of The Art.*

*Since there are other accounting packages available for the /// I recommend you get away from State Of The Art if possible. Products such as the BPI accounting system can be loaded onto your hard disk system with the addition of the Selector program switching utility. You will also be able to load the Mail List Manager program onto the hard disk with Selector. Using the Selector will allow you to place all of your programs on the hard disk and quickly switch between them.*

*The Apple Mail List Manager is an old (and slow) program that you may want to replace. I recommend /// E-Z Pieces from Haba Systems. It offers many enhancements over Mail List Manager, including the ability to store your files on the hard disk (Mail List Manager only stores files on a 140K floppy). I hope these remarks have helped you. If we can be of any further assistance, please do write again.*

## Compatibility

Dear Bob:

Congratulations on the resumption of ON THREE. Prior to your suspension of publication I wrote you a letter criticizing ON THREE in its prior incarnation for serving as an outlet for your frustrations with Apple Computer. I wanted to take this opportunity to let you know that I am very pleased with the new ON THREE. All of us who own Apple ///'s are naturally disappointed in Apple's decision to discontinue production. However, that decision is reality. We must all now decide where we go from here. ON THREE should be a great deal of help in the process.

I would be interested in obtaining additional information from you regarding both Draw ON /// and the ON TIME Desktop Manager. Among other things, I would like to know whether these programs are compatible with Catalyst and how they interact with /// E-Z Pieces. If you have additional information which you can send, please do so.

Once again, congratulations. I look forward to the next issue of ON THREE.

John L. Ryder  
Memphis, TN

*Both Draw ON /// and the ON TIME Desktop Manager are compatible with Catalyst and /// E-Z Pieces. It is possible that you might run into a stack overflow problem in a 256K memory machine if Draw ON /// and Catalyst are in memory at the same time. There is no such possibility in a 512K machine.*

*Thanks for the kind words about ON THREE.*

## Looking for Public Domain

Dear ON THREE:

Do you know if there are any groups around the country that handle public domain software for the Apple ///?

Lynn P. Caswell  
Elk River, MN

*I have heard of a number of Apple /// user groups that offer a limited number of public domain programs, also there is some PD software available through some bulletin boards around the country. Please check the latest issue for both user group listings and bulletin boards that may have what you're looking for.*

## Strange You Should Ask...

Dear Bob;

Just received the new ON THREE, and I'm impressed with Val Golding's work. This issue emphasized Business Basic a bit much for this Pascal programmer's tastes, but I'm sure I'll have my turn. I'd just like to add to the list of what's needed for the ///:

1. A native SOS-based Pascal system conforming to the Apple/UCSD syntax, vocabulary, and literary structure, but providing for an adjustable-size data space (more than 64K)...

2. Ditto for a C compiler...

3. A SOS revision permitting more than four expansion slots at a time... it's too easy to fill four slots today! (A ProFile card, a network card, a UPIC, and a UniDisk card... then where do I put my PCPI Z-80 card and my PCPI 8086 card and my PCPI 68000 card and my...?)

4. An external expansion box to go with it...

5. The same SOS revision permitting Megabytes of RAM (I've heard rumors, Bob)...

6. A flexible relational database system (of course, with a C compiler, we could always port Dbase III... Dbase ///?)...

And more companies willing to support our fine machines!

Contrary to Art Schumer's wishful thinking, according to Bill Wasmund at Quark, they have *no* plans for new products for the ///, although they are still giving good support to the products already out... but don't expect to see Catalyst Version 3.0 for us... just the //e and //c! (Although, with a letter writing campaign, *who knows* what could happen?)

Keep up the good work!

Tom Betz  
Yonkers, NY

*Several of the items you mentioned on your "wish list" are actually under consideration. More would be possible if we were to ever get a measure of cooperation from the folks at Apple, but it seems no one there has an interest in exhuming the corpse, so to speak.*

*Rob Turner of our staff is busily at work disassembling Pascal. The hope, if not the promise, is there for a version to run under a 65C802 which, by putting the 802 to work the way it was intended to be used, would allow bank-switched data space and faster running programs.*

*C is not as likely a possibility because of the limited interest (at least on the ///). We will just have to wait and C.*

*Another gleam in Bob Consorti's eye is exactly what you are looking for: a SOS revision to allow more than four slots. Not having "original" source code for SOS makes working on a task like this far more involved than it needs to be. There is one expansion box available from DA Data Systems, but because of the way the slots are phantomed, it can't be used for a peripheral which may generate interrupts.*

*Megabytes of RAM on a ///? Well, we've heard rumors, too.*

*We didn't see where Art Schumer specifically said Quark was planning new products for the ///. However, it is our understanding that Quark now has updates available for Word Juggler which will allow it to run under and utilize the memory of a 512K machine.*

*You're right, the January issue did feature Business Basic. However, our column "Now About this Basic Business..." will not appear in each issue. Similarly, Dr. Astrahan's "Graphically Speaking" column, which we introduced in*

*February, will not always be in each issue. We plan also to bring in columns on Pascal (there you go) and assembly on the same basis. We have to make the most of the limited space we have available, and this seems to be the best solution.*

## In the Driver's Seat

Dear Bob,

Re: Pascal Toolkit, Pascal 2.0 Compiler, Pascal 1.2, Desktop Manager, Latest drivers.

First, it's nice having ON THREE back. I thought we'd lost you for good.

I wanted to write to encourage you to get the above-mentioned Pascal updates and tools together and available ASAP. I had occasion to watch (though not use) Pascal 1.2 and the 2.0 Compiler way back in beta days and I am one more (amateur) programmer who really wants it and will pay for it. The 3's Company BBS (via Tom Betz and Rob Turner's notes) suggest you're packaging the whole thing. I hope that's true and I hope you'll hurry!

I have the Pascal Technical Reference Manual and Apple's very nice data base tool set—called Record Processing Services or RPS. I don't know if anyone's in need of these or not.

Also, I want to put in a 'ready-to-order' order for the Desktop Manager which I understand is now anticipated for Feb '86. For the record, I'm most interested in the macro facility, notepad, clipboard and disk management features, as I am mainly a writer.

Lastly, I have a suggestion for the magazine: How about a listing of SOS drivers and their latest revision numbers? I spent some time sorting out my own drivers to make sure I was using the most current versions I had received from hither and yon, but I don't have any way of knowing whether my set is actually the very latest available. It might be worth setting up an informal 'archive' of the latest drivers, available for a copying fee or whatever.

As a start, here's the best of what I've got:

driver name	supplier	revision
.CONSOLE	Apple	1.30
.AUDIO	Apple	1.00
.PROFILE	Apple	1.30
.PRINTER	Pkaso	3.20
.ONTIME	ONTHREE	1.30
.GRAFIX	Apple	1.30
.RS232	Apple	1.30
.CATALYST	Quark	2.01
.QUARKCOM	Quark	2.00
.FMT	Apple	1.30

Once again, I'm glad you're back.

Jim Redfield  
Costa Mesa, CA

*Thanks for your comments, they spurred me to action. Look in ON THREE for information on an upgrade facility for Pascal and all of the drivers. In addition we have set up a diskette backup service. If you have a bad copy of VisiCalc, AppleWriter or most any other disk, we will fix it for you. Information on this service is located elsewhere in the magazine.*

*I'm glad you would like to order the Desktop Manager, you (and all other Apple /// owners) are the reason we spent so much time and money developing it. The Desktop Manager really makes computer life easy.*

### MacThree Program

Dear Mr. Consorti,

It was a pleasure to receive the January, 1986 issue of ON THREE magazine. I am glad that you plan to continue publishing this interesting and useful journal in addition to your manufacturing and distribution activities. The move to monthly publication is definitely a step forward, as is the hiring of Val Golding as the magazine's editor. I anticipate much of interest from ON THREE in 1986, and eagerly await February's issue.

After having read January's Block\_\_Write column, I must say that I applaud your courageous decision to stand by the Apple ///, despite the fact that Apple has deemed it necessary to throw this particular baby out with the bathwater. Although I use a number of different computers in my capacity as systems consultant, I bear a special fondness for my Apple ///, and view with much amusement Apple's recent efforts to evolve the Apple ][ into a mutant hybrid of the /// and Macintosh. Why didn't they exploit the superior design of the /// in the first place, improving the machine's few but irritating shortcomings on the way to Macintosh compatibility, I'll never know. Had they done so, I think they would have produced a perfect bridge between ][ line and the Mac line. Instead they're now playing catch-up, which is a very funny situation indeed for a company that, at the time it hired me, was the market leader and arrogant almost to the point of insufferability. You prove on a daily basis that the Apple /// is not only a good machine, but a viable one (albeit in a much smaller marketplace than that targeted by Apple's marketers circa 1980), and for that you deserve the congratulations and the thanks of those of us who were involved in the machine's design and early development.

Have you or anyone known to you taken a good hard look at putting a 65C802 in an Apple ///? Having tried it myself, I'm painfully aware of the obvious differences between the 16-bit chip and the Apple's CPU. However, since the exact same chip functions perfectly in an Apple //e, I believe that the 65C802 might be made to work in an Apple /// if problems of bus noise and timing can be identified and solved. I have not met anyone who has investigated the problem beyond the unsuccessful direct-chip-replacement stage, but I would venture to guess that a good hardware engineer could devise a way to mate the 65C802 and the Apple /// inside of a month or two. Perhaps the upgrade might even be limited to a special, chip-sized "buffering board" that could be sandwiched between the 65C802 CPU chip and the CPU socket on the Apple /// motherboard. By producing a 16-bit upgrade for the Apple ///, you could conceivably extend the machine's commercial life for a while, especially if you could also acquire a license from Apple to rewrite SOS and other system software to take advantage of the 65C802's enhanced instruction set. I for one, would be glad to tackle such a software project, and I'm sure you could find others as well. Such an upgrade would render the Apple /// competitive in terms of hardware power with the IBM PC at the very least, and would open the door to many applications that are not now practical on the ///. Please give this idea some thought, and don't hesitate to contact me if you'd like to pursue it further.

In any event, thanks again for resuming publication of ON THREE, and best wishes for the holiday season, not to mention a prosperous 1986!

Jim Merritt  
Morrow Bay, CA

Thanks for taking the time to write those interesting comments about the ///. Your special fondness definitely shows through. With respect to the reasons Apple did not follow through with continued development of the ///, we believe that a combination of circumstances led to its early and regrettable demise. The position of management at that time was, as you indicated, very Macintosh-oriented and the /// group scarcely had a voice in development. This can be attested to by the amount of time and expense that went into the Apple /// plus upgrade, only to have the production discontinued after only some 15,000 had been manufactured. Moreover, the early bad press (on a bad machine, released before it was ready) was never overcome and had to have played a major role in management's decision to drop it.

Nevertheless, we do have the /// to thank for the subsequent development of Apple ][ ProDOS with its compatible file structure and the concept of the MLI, or Machine Language Interface. As to why the SOS concept was not carried over into the Macintosh we will perhaps never know, but Apple again showed their marketing ineptitude with the introduction of an innovative Lisa, foisted upon the public at a unconscionable price. It appears that the recent management changes at Apple and the structural reorganization may at last have paved the road to sensible products at sensible prices.

Still, Macintosh is an orphan, with a non-standard barely compatible file structure. It is to be hoped that the two newest versions, to be released in January and later in the year, will offer substantially more compatibility with both the ][ and the ///. Right now that compatibility is available essentially through the "universal" UniDisk 3.5, which with its own 65C02 and ROM, makes it possible to read and write either 512 or 524 byte blocks. Our file conversion utility is currently taking a back seat until more is known of the new Unidisk-type drive for the Mac.

Yes we have used, and do use, the 65C802 in some of our ///'s and have not, with the exception of interrupt handling under certain circumstances, run into the problems you mentioned. In fact we are well on the way to finishing an assembler written almost entirely in 65C802 on a ///. We would appreciate any input from you, particularly relating to timing and bus noise.

### Titan Address

Dear Sir:

I read about the Titan Apple ][ emulation cards in your magazine, however I was unable to find an address or phone number for them.

Your publication is loaded with great tricks, software, hardware, etc. It would be handy if this great wealth of products and information could be compiled into a catalog/manual format and bound in a ring binder that could be periodically added to and updated.

Michael D. Johnson  
Franklin, LA

The information you requested is:

Titan Technologies, Inc.  
310 W. Ann St.  
Ann Arbor, MI 48104  
(313) 662-8542

An index such as you suggest would be a large undertaking, but we are certainly willing to consider it. You might also be interested in reading Bob Consorti's column in connection with suggestions for future ON THREE projects.

### Old Age Creeps Up

Dear Bob,

I am having problems with *System Failure \$01*. Does this sound like the motherboard or power supply? My SOS Reference Manual calls it an Invalid BRK, but sometimes it occurs when I am not entering anything in the machine. The monitor generally changes pitch from the typical 60 Hz hum, then I get right arrows in 40-column mode (inverse video—I'm using Catalyst and Word Juggler) over the entire screen for a few seconds before I fall back into the screen I was in before the failure, except the error message now appears on the bottom of the screen. Any Ideas?

Also do we still turn to our friendly Apple dealer when we need service or do we go to the trouble of mailing everything to Sun Data in Utah. Last I heard Apple had a 12-year commitment to spare parts and service for the ///. Have they changed policy again?

Kelly McGrew  
Olympia, WA

*The System Failure \$01 is impossible to accurately diagnose, but it would appear to be the main board, and it may well be as simple as advancing age. How long has it been since you have done any preventative maintenance? Using a chip-puller, try pulling all chips and cleaning the sockets with a camels hair brush and just a touch of denatured alcohol (easy does it), then replacing the chips, making sure they are firmly seated. With luck your system should be in shape again for several more months.*

*It appears that Apple has dropped all support for the ///, including service. Most dealers will still service the machine, depending on their supply of spare parts. Sun Data seems to have Apple's complete inventory of spare parts now.*

### Backing Up Backup /// ?

Dear Bob:

I am tired of Backup /// aborting on the last diskette of a backup because of a verify failure. I have two questions:

1. Do you know of an any alternative program that does not die so ungracefully?
2. Is the source code available from any place so that it may be modified to eliminate this trauma?

I am a reasonably proficient Pascal programmer. If the source code is available I would be willing to work on it so that it would be possible to replace a disk that fails verification and restart backing up from the beginning of that diskette. Upon completion I would be pleased to place the program in the public domain for all to benefit.

Any help or suggestions you have would be appreciated.

Milton L. Johnson  
Milwaukie, OR

*We are tired as well. Unfortunately, as much as we like to offer positive responses to reader's questions in this column, this is one case where we can not. There is no alternative program to our knowledge, and Apple is hanging tightly onto any and all Apple /// source code.*

*The reason Backup /// reverts to the beginning upon an abort or verify error is simply that the program makes no provision for saving start-of-disk pointers, a fairly simple thing to do when one has the source code available. Nevertheless, we wouldn't put it past the ingenuity of some of our readers to come up with a fix. How about it, gang?*

### Desktop Manager and Powerkeys

Dear Sir:

I would like additional information on two of your products; *ONTIME Desktop Manager* and *UniDisk ///.5*.

On the *ONTIME Desktop Manager*, do you know if it will work with Powerkeys? The program is part of the .console drivers and is invoked by pressing [closed-apple-tab]. Before I purchase it, I want to be sure there is no conflict and that both programs will function together on my system.

My last question is what is the scheduled release date of your *UniDisk ///.5* Macintosh conversion utility.

P.A. Sikorski  
Norwich, CT

*Our Desktop Manager will work fine with Powerkeys and there will be no conflict. It should be noted also that Desktop Manager has the ability to create macros up to a size of 2K each, considerably larger than Powerkeys. This facility will appear in a module to be released shortly after the main product.*

At this time we have not scheduled a release date for the UniDisk conversion program. We are waiting to see the new directory and file structures which will be used on the Macintosh Plus before commencing work on the conversion program.

### Checking Lexicheck

Gentlemen:

I am in complete agreement with the January article on the superiority of Word Juggler over AppleWriter ///. I also miss the ability to 'print to a file,' but the many advantages more than outweigh that one disadvantage.

The February article on Lexicheck makes me believe that there may have been a version that has been updated over the one I bought. I do not have the option of getting the program to "guess a word" either during the typing or while checking the typed manuscript. Quark has not offered to supply, even for a price, updated versions. I sent in the registration several months ago.

Some users of Word Juggler may have wondered why does "G" appear at the bottom of my screen instead of the *bel* sounding. I wondered for a while until I realized that it was probably because I had inadvertently hit [CONTROL-8] on the keypad instead of [SHIFT-8]. To get your *bel* ringing again, hit [CONTROL-8] again.

R.B. Thompson  
Raleigh, NC

*Thanks for the tip on the bell. We've run into similar situations in other programs. Your question on Lexicheck we'll throw open for discussion. It's quite likely one of our readers has some information.*



### Oops

We goofed. In the February article "Fooling the Apple DMP," we chopped off three important program lines. Here they are:

```
1500 REM * IF NOT A BACKSPACE, CHECK FOR RETURN
1510 IF ASC(KEYPRESS) <> 8 THEN 1580
1520 IF STORE$=" " THEN 1410
```

Sorry 'bout that; hope this fixes any problems you may have.

**ON THREE Presents...**

## **ON THREE O'Clock**

***Now is the Time  
for a real-time clock***

Believe it or not, a lot of folks have plain forgotten (or never knew) that the Apple /// was designed to operate with a built-in clock and that, with a clock chip installed, SOS will automatically time stamp and date all file saves.

When the Apple /// was first released, the supplier of Apple's clock chips could not supply a working clock. As a result, the /// was supplied without a clock of any kind. Now maybe you are wondering when you list a disk directory, how the time and date magically appears.

Not too long ago ON THREE developed a clock for the Apple /// which plugs in right where the never-released Apple clock was supposed to go, and for just \$49.95 plus \$3.00 shipping and handling, this easy to install, SOS-compatible clock can be yours. It comes with comprehensive instructions and ON THREE's limited six-month warranty and does not use any of your precious slots.

With an ON THREE O'Clock installed, whenever you save or modify any type of file, the current time and date will be added to the directory listing so you can always tell at a glance which file you last worked on, and when. But that's not all. Business Basic has two reserved variables, DATE\$ and TIME\$, which return, respectively, the current date and time to your BASIC program. These reserved variables can then be used whenever you want to print the date and/or time in a BASIC program.

### **Special Combination Offer**

There's a great deal more you can do with ON THREE's ON THREE O'Clock if you also have our Desktop Manager. Whenever you want, you can display the current date and time on the screen with one keypress. Since this is a background function, you can be word processing with AppleWriter or entering data into VisiCalc, and with one keystroke you can obtain updated time information. In addition, you can use the Desktop Manager's Appointment Calendar to enter items you want to be reminded of and, like magic, when the time comes, no matter what you are doing, a message will appear on your screen to gently chide you via the Desktop Manager to make that phone call now, etc.

Now The Appointment Calendar is not the only feature of the Desktop Manager, you can read about the Calculator, the Notepad, and the others elsewhere, but since the Desktop Manager requires a clock, we want to offer you a money-saving deal. Purchased together, you can get the ON THREE O'Clock and the Desktop Manager for only \$173.95 plus \$8.00 shipping and handling. Now is the time to take advantage of this special offer.



**\$49.95**  
plus \$3.00  
shipping and  
handling

**Desktop Manager/ON THREE O'Clock Combo**  
**\$173.95** plus \$8.00 shipping and handling

# Taming Timing

dan martin

One afternoon, one of those days when a blank, formatted diskette was nowhere to be found, I booted up the System Utilities diskette to take care of some housekeeping. I popped a new disk into Drive 2, selected the "Device Handling Commands" option from the main menu and prepared, as I had numerous times before, to format a volume.

The rattle and clatter from Drive 2 followed and much to my surprise a beep and diagnostic message appeared: "Device Dependent Error #34; See Manufacturer's Documentation". What manufacturer's documentation? We all know the value of that advice in the often shadowy world of Apple /// support. I called the Apple technical support number (still active at that time) and was advised that my drive was off-speed. My dealer told me that \$22 would take care of the problem.

Being the type of person that doesn't enjoy dependence on dealers of any kind (for example, I change my own oil and filter), I felt that an adjustment of this nature might not be too difficult a task to undertake. I proceeded cautiously to disassemble my, till now, trusty Number Two drive to assess at least the magnitude of the problem.

The procedure which follows should permit any reader to effect a speed adjustment with a mere fraction of the disassembly that I originally went through just to identify the simple (and surprisingly accessible) adjustment for speed control.

The things you'll need to accomplish this task are:

- A System Utilities disk
- A Phillips screwdriver
- A miniature jeweler's-type screwdriver (or virtually any reasonable facsimile—I used my Swiss Army knife)
- A fluorescent light source
- A little patience and fortitude

The following procedure is only for the external Disk /// drive produced by Apple for the ///. My guess is that the internal drive would naturally involve a bit more disassembly and that drives by other manufacturers would be conceptually similar.

## Step 1—Disassembly

**Warning!** *Disassembly should be done with all power to the Apple /// turned off before you so much as pick up a screwdriver.*

Begin by disconnecting the ribbon cable connector from the back of the /// to permit easier handling of the drive during disassembly. Place the disk unit on its top (belly up) on a flat working surface covered with cardboard or paper if you're inclined to be real careful about the finish. Remove the four Phillips head screws from the bottom (and put them in a secure place). Next remove the bottom cover of the disk drive housing. Once that's done, you're ready for Step 2.

## Step 2—Set Up for Timing

Place the unit, still on its back, near the /// in such a way that will allow you to both connect the ribbon cable to the rear apron connector on the /// and also to have access to the rear of the drive. Additionally, the heart of the whole procedure is to accomplish this work using only fluorescent lighting, either via overhead units or a portable desk lamp, since this light source is the primary timing unit as you will see shortly. As implied above, *with the power to the /// still off*, reconnect the ribbon cable to the disk port on the back of the ///. Once you have the drive positioned and correctly reconnected, proceed to the next step.

## Step 3—Timing the Disk Drive

Making sure no conductive material is touching any part of the disk drive or interfering with the drive mechanism, reconnect power to the /// and boot it using the System Utilities disk. Now place a formatted diskette (preferably one you'd be willing to lose, just to be on the safe side) into the external, upside-down disk drive. Remember, if you're using single-sided disks, to turn the diskette upside down as well. Select the device handling choice on the primary System Utilities menu.

Before you proceed, you must locate an important adjustment. Viewing the disk unit on its back from the rear, notice that the ribbon cable leaves the unit on the lower left-hand corner of the back as you're looking at it. On the bottom right, then, is the connector for additional drives. Just above the serial number of the unit and about an inch forward of the metal back panel of the drive housing is a small circuit board with a little black connector in its upper left-hand corner. On the opposite side of the board (the right-hand side) is an integrated circuit chip. The next component to the immediate right of the integrated circuit chip is a long, narrow plastic device with a brass screw head on its right side less than one-half inch from the right side of the metal housing.

That device is a potentiometer (adjustable resistor) which controls the speed of the disk drive motor. We'll get back to it soon. Note: there is no voltage between the brass adjusting screw and the metal case, so a metal screwdriver is alright to use—just don't touch the electronic components. A plastic or wooden adjustment tool (e.g., a filed plastic or wooden toothpick) is safer to use.

Before you grab your little screwdriver, let's review how you're going to be able to tell whether the drive is running off speed. On the top of the unit (as you're looking at it) you'll see a small wheel with a funny black and white checkered pattern going around the surface of the wheel. One track of the pattern has a "60" in it; the other has a "50". These markings refer to the frequency of the A.C. power used to drive the computer.

Virtually all commercial power in the U.S. is 60 Hertz (or cycles per second). Here's where the fluorescent light comes in. Fluorescent lights send out light waves that 'beat' at 60 cycles per second, i.e., the tube sends out light pulses 60 times every second. These light waves interact, or synchronize, with the funny pattern on the wheel to produce an image that will tell you when the drive is running at the correct speed.

Once the drive starts spinning, you will see a pattern that is reminiscent of the 'wagon wheel' effect you used to see on the stage coach in the old cowboy movies when you could swear the wheels were actually turning backwards. That optical illusion was caused by the interaction between light from the spinning wheel spokes and the movie camera shutter clicking away at about 24 frames per second. Many fancy stereos incorporate this identical timing mechanism for audiophiles with unhumanly discriminating ears.

Now we're ready to time the drive. Select the "Verify a Volume" function under the "Device Handling Commands" menu. All this does is read your diskette to identify damaged blocks on its magnetic medium. Verify the medium of volume .D2 for the first 280 blocks, just as the utility offers by default. We don't really care about the diskette, it's just that this utility causes the drive motor to run for a fairly long time.

When you hit the last carriage return, the drive will start to run. Now watch the funny spoke-like pattern on the wheel while the disk is whirring. Remember to have a fluorescent light as the dominant light source for this step or it won't work. The image produced by the spinning checkered pattern (on the outer track of the pattern) will seem to be slowly moving either clockwise or counter-clockwise, depending on whether the drive is running fast or slow. The procedure for timing the drive is to adjust the brass screw on the potentiometer of the circuit board until the wheel-spoke pattern no longer moves in either clock direction, but appears to remain stationary.

The direction of apparent pattern motion will be the same direction that you adjust the brass screw head. If the pattern seems to be running counter-clockwise, turn the adjusting screw counter-clockwise also to offset the motion. Reverse the adjustment if the pattern motion is clockwise. This process will probably require execution of the verify disk function a number of times—just hit a few carriage returns to run it again. Once the pattern appears stationary, you're done with timing.

#### Step 4—Reassembly

Turn off the power to the ///. Replace the plastic cover and reinsert the four Phillips head screws that you can find if you followed my earlier instructions. Don't tighten them down too hard. You don't want to strip the threads on the drive unit. This is a disk drive, not an oil drain plug.

#### Step 5—Gratification

Now go out and buy your spouse a gift with the \$22 you saved. She may yet get to like that computer that chews up all your time. Happy formatting.

*Editors note: If a fluorescent light source is not available, you may be able to adjust your drives in emulation mode. There are several programs for the Apple II which display drive speed to the screen.*



## ON THREE Presents . . .

I ♥ My Apple ///

### Spring T-Shirt Bonanza!

#### • Bright t-shirt colors for sunny spring days

- yellow
- blue
- white
- beige

**\$11.95** plus \$3 s/h

#### • Maintain your bod like your ///

I ♥ My /// Sweat Shirts

- yellow
- blue
- white
- silver

**\$18.95** plus \$3 s/h

sm     med     lg     x-lg

#### • Cap it off with I ♥ My /// Caps

**\$5.95** plus \$2 s/h

For April:

# New Products

For unfathomable reasons new microprocessors or upgrades to them seem to appear on the market shortly after the Spring equinox. This year is no exception as Eastern Design Labs has announced the 65FOO2, an enhancement to the 65C02 with which it is downwards compatible. It is said to operate without timing or other problems in the Apple I, II, III, IV and Commodore Pet.

Included are six new branch instructions, seven load and store instructions, four stack instructions, five logic instructions, seven miscellaneous instructions, one augmented addressing mode, and one new addressing mode. They are detailed below.

### Branch Instructions

- BNV Branch Never
- BRP Branch Reset Pressed
- BSC Branch on System Crash
- BST Branch SomeTimes
- DCC Detour on Carry Clear
- DCS Detour on Carry Set

### Load/Store Instructions

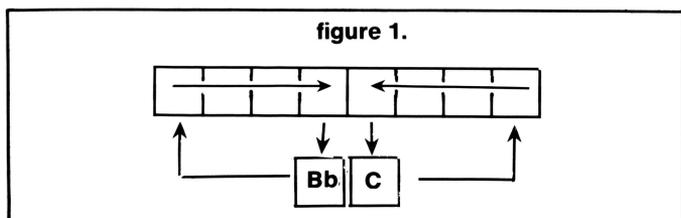
- LNB Load Next Byte
- LAS Load A from mem and Store back
- LXS Load X from mem and Store back
- LYS Load Y from mem and Store back
- SAL Store A in mem and Load back
- SXL Store X in mem and Load back
- SYL Store Y in mem and Load back

### Stack Instructions

- PHS PusH Stack
- PLS PuLl Stack
- TPS Transfer Status (P-reg) to Stack
- TSP Transfer Stack pointer to Status

### Logic Instructions

- LSC Logic Shift Center (see figure 1)
- ALA ALso mem with A
- MYA MaYbe mem with A
- ROD ROTate Down
- ROU ROTate Up
- XAM eXclusive And Mem with A



### Miscellaneous Instructions

- EMF Emulate MainFrame
- IRD Interrupt Request Denied
- LWA Leave Well enough Alone
- MNM Mask Non-Maskable interrupt
- RBR Reset Bits Randomly
- SBR Set Bits Randomly
- WNI Wait for Next Instruction to execute

### Addressing Modes

The immediate mode may now be used with all store instructions. Assembler syntax is shown below.

#### STA # $\$7F$

The new addressing mode is known as *Indirect Indirect*. Assembler syntax is as follows:

#### LDA ( $\$0B$ ),5

The accumulator will be loaded from the location pointed to by  $\$0B,0C$  plus 5 or other value specified.

# SmartTax

T.M.

The fast, powerful, yet easy to use software program that calculates both California State and Federal income taxes simultaneously.

Designed for both personal and professional use, SmartTax allows for easy on-screen editing, displays results side by side on your Apple III screen, and prints directly on to standard IRS and California State forms (1040, A, B, C, D, E, G, SE, W, 2441, 540, A, B, C-E-F, D, and G).

Complete with easy to follow manual and 30 day money-back guarantee, SmartTax is the way to make this year's tax return preparation the easiest ever.

**\$89.95** (+ \$1.50 shipping & handling. Cal. residents please add 6% sales tax).

For immediate shipment, call Officeware Data Systems at:

**(415) 454-4357**

**Officeware Data Systems**

351 OLEMA RD. STE. 15, FAIRFAX, CA 94930

Department of the Treasury—Internal Revenue Service

**1040 U.S. Individual Income Tax Return 1985**

For the year January 1, 1985, or other tax year beginning 1985 ending

Your first name and initial (if joint return, also give spouse's name and initial) Last name

Present home address (number and street, including apartment number, or rural route)

City, town or post office, state, and ZIP code

Use IRS label. Other, please print

Do you want \$1 to go to this fund? For Ph

Does your spouse want \$1 to go to this fund? (only one had income) Security no

©1986 Officeware Data Systems

# 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.

- |  |  |  |   |   |
|--|--|--|---|---|
| <p><b>California</b><br/> <i>Sacramento Apple /// User Group</i><br/>             1433 Elsdon Circle, Carmichael, CA 95608<br/>             (916) 482-6660</p> <p><i>Orange County Apple /// User Group</i><br/>             22501 Eloise Ave., El Toro, CA 92630<br/>             (714) 951-1231</p> <p><i>Los Angeles-South Bay Apple /// Users Group</i><br/>             P.O. Box 432, Redondo Beach, CA 90277<br/>             (213) 316-7738</p> <p><i>Apple /// Users of Northern California</i><br/>             220 Redwood Highway #184<br/>             Mill Valley, CA 94941</p> <p><i>International Apple Core Apple /// S.I.G.</i><br/>             908 George Street, Santa Clara, CA 95054<br/>             (408) 727-7652</p> <p><b>Canada</b><br/> <i>Apples British Columbia Computer Society Apple /// S.I.G.</i><br/>             P.O. Box 80569, Burnaby<br/>             BC Canada V5H3X9</p> | <p><i>Canadian Apple /// Users Group</i><br/>             80 Antibes Dr. Suite 2805<br/>             Willowdale, Ontario, Canada M25R 3N5<br/>             (416) 665-3622</p> <p><b>Colorado</b><br/> <i>Colorado Apple Three User Group</i><br/>             P.O. Box 3155, Englewood, CO 80112</p> <p><b>Connecticut</b><br/> <i>Apple /// Society of Southern Connecticut</i><br/>             34 Burr School Rd., Westport, CT 06880<br/>             (203) 226-4198</p> <p><b>Florida</b><br/> <i>Sarasota Apple /// User Group</i><br/>             c/o Computer Centre<br/>             909 S. Tamiami Trail<br/>             Nokomis, FL 33555<br/>             (813) 484-0421</p> <p><b>Georgia</b><br/> <i>Atlanta /// Society</i><br/>             385 Saddle Lake Drive, Roswell, GA 30076<br/>             (404) 992-3130</p> | <p><b>Kansas</b><br/> <i>Kansas City Apple /// User Group</i><br/>             3800 Cambridge, Kansas City, KS 66103<br/>             (913) 588-6025</p> <p><b>Maine</b><br/> <i>So. Maine Apple Users Group</i><br/>             Casco St., Freeport ME 04033<br/>             (207) 865-4761, X 2249</p> <p><b>Maryland</b><br/> <i>Apple /// SIG Chairman Washington Apple Pi</i><br/>             8227 Woodmont Av. #201<br/>             Bethesda, MD 20814<br/>             (301) 654-8060</p> <p><b>Minnesota</b><br/> <i>Minnesota Apple Corp Users Group</i><br/>             P.O. Box 796, Hopkins, MN 55343</p> <p><b>New Jersey</b><br/> <i>North Jersey Apple /// Users Group</i><br/>             c/o Roger T. Richardson<br/>             P.O. Box 251, Allamuchy, NJ 07820<br/>             (201) 852-7710</p> | <p><b>North Carolina</b><br/> <i>North Carolina Apple /// User Group</i><br/>             2609 North Duke St. #103<br/>             Durham, NC 27704</p> <p><b>Ohio</b><br/> <i>Cincinnati Apple /// User Group</i><br/>             5242 Horizonvue Drive,<br/>             Cincinnati, OH 45239<br/>             (513) 542-7146</p> <p><i>Apple Dayton - Apple /// S.I.G.</i><br/>             P.O. Box 1666, Fairborn, OH 45324-7666<br/>             (513) 879-5895</p> <p><b>Oregon</b><br/> <i>Portland Apple /// Users Group</i><br/>             Portland OR (503) 225-1623</p> <p><b>Overseas</b><br/> <i>Apple /// Owners &amp; Users Group International</i><br/>             c/o Maj. H. Joseph Dobrowski<br/>             AF SOUTH/JLD Box 149<br/>             FPO New York 09524</p> <p><i>Apple User Group Europe e.V.</i><br/>             Box 11 01 69 D-4200<br/>             Oberhausen 11<br/>             West Germany<br/>             0049-6195-7<br/>             3917</p> | <p><i>British Apple Systems User Group (BASUG) Apple /// S.I.G.</i><br/>             P.O. Box 174, Watford Herts<br/>             England WD2 6NF<br/>             0727 73390/72728</p> <p><i>Le Club Apple</i><br/>             43 Avenue de la Grande-Armee<br/>             75116 Paris, France</p> <p><b>Texas</b><br/> <i>Apple Corps of Dallas Apple /// SIG</i><br/>             P.O. Box 5537, Richardson, TX 75080</p> <p><i>River City Apple Corps /// S.I.G.</i><br/>             Box 13349, Austin, TX 78711<br/>             (512) 454-9962</p> <p><i>Houston Area Apple Users Group (Apple /// Division)</i><br/>             P.O. Box 610150, Houston, TX 77063<br/>             (713) 480-5690 or 974-5153</p> <p><b>Virginia</b><br/> <i>Charlottesville Apple /// User Group</i><br/>             216 Turkey Ridge Road<br/>             Charlottesville, VA 22901<br/>             (804) 642-5655</p> <p><i>Greater Tidewater Apple /// User Group</i><br/>             Route 2, Box 216, Hayes, VA 23072<br/>             (804) 642-5655 or 898-3500, ext. 2671</p> |
|--|--|--|---|---|

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.		Modula-2	MU
Business	BB	Pascal	PA
Basic			
Catalyst	CT	ProFile	PR
Cobol	CO	Quark	QU
CP/M	CP	SOS	SO
Data Base	DB	Spread-sheets	SS
Education	ED	Telecom.	TC
Financial	FI	Word Proc.	WP
Fortran	FO	Emulation	AE
General	GE	/// E-Z	EP
		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) 253-2298	Su-Sa	6-9pm	Eastern	BB, PA, MD, WP, MI
Don Loosli	MI	(313) 626-3848	M-F	9am-5pm	Eastern	GE, WP, SS, DB
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	GE, PA, BB, 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
R.B. Thompson	NC	(919) 787-1703	Su-Sa	10am-10pm	Eastern	BB, DB, GE, SS, WP
J. Donald Glenn	NE	(402) 291-9177	Su-Th	7-10am	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
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
Wayne Hale	CA	(619) 450-3856	M-F	7-11am	Pacific	BB, GR, CT
Dennis R. Cohen	CA	(818) 956-8559	Su, M-F, Sa	10am-10pm 7-9pm 12n-6pm	Pacific	GE, PA, MU, WP, DB, SO
Kelly C. McGrew	WA	(206) 943-8533	Su-M, Th-Sa	7-9pm	Pacific	DB, GR, SS, PR, MD, CT

# ON THREE's Desktop Manager gives you a choice:

*The Desktop Manager places all of the utilities you ever wanted . . . Appointment Calendar . . . Notepad . . . Calculator . . . Disk Utilities . . . and more . . . into every single program you own, just like they were part of it. Instantly available from /// E-Z Pieces, VisiCalc, AppleWriter, BPI, and all other programs, the Desktop Manager will clear your desk pronto.*

While word processing, have you ever needed to multiply two numbers? Gotten upset because you have a few thousand dollars worth of computer equipment at your fingertips and still can't multiply two figures when *you* want to? Or, you're entering data in a spreadsheet and can't find either a scratchpad or a pen to jot a note. While you're digging under piles of paperwork, you probably mutter something unprintable under your breath.

Perhaps you are word processing and need to save a file, but aren't sure if the file name you want to save to exists. Too bad the program you're using won't catalog a disk. Similarly, you may need to save a file and discover there is no room left on

your current work disk. You have blank disks, but none are formatted, and if you leave the word processor to format a disk with the System Utilities, all of your work will be lost.

Do you see yourself in this picture? How would you like to clear your desk of that old-fashioned calculator, the pens and paper, your appointment calendar and increase your productivity? *ON THREE's Desktop Manager* to the rescue! It will do these things and more. From within any Apple /// program, a keystroke will suspend your current program and display a window into the *Desktop Manager*. You can stop whatever you are doing, instantly go to the *Desktop Manager* and select any of the following:

- An Appointment Calendar. Enter appointment times or other activities at specific times, like "Call Johnson at 10:30 AM," or "5:00 PM Stop at supermarket. Pick up milk." At 10:30 AM that day a window will appear on your screen and display the first message and at 5:00 PM the second message will appear.
- A full feature Calculator (SIN, COS, TAN, EXP, LOG, LN, memory, base conversions and more). Change from decimal to hex to binary and back. A scrolling paper tape will show your last calculations.
- An easy to use, always there Notepad with full editing capability and jam-packed with features. You can jot page after page of notes to yourself and even print them out. You may never need to use a word processor again!
- An optional Disk Utilities module. Use it to Format Disks, List, Unlock, Delete, Rename and Copy Files. Most of the features of the System Utilities, available in a second instead of a minute.
- An optional ASCII Conversion Table which lists, in an easy to understand form, decimal and hexadecimal values for all the ASCII characters. Useful to determine special character sequences to send to your printer, or for programming.

After noting a forthcoming meeting on the Appointment Calendar, totaling some figures you are working on, making a note to yourself about your upcoming vacation, or copying the files your boss needs, simply press ESCAPE and you are instantly back in your original program and nothing has changed from when you left it a moment ago. Even the cursor is blinking at the same place and you have saved loads of time.

The *Desktop Manager* also lets you use the mouse instead of cursor keys in any program. When you move the mouse around the screen, the cursor will follow, left, right, up and down. Clicking it also acts as an ESCAPE or 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.

Note: The *Desktop Manager* requires 256K memory (512 recommended, since it uses about 40K), an *ON THREE O'Clock*, Apple Clock or compatible Apple /// clock chip.

Note: Clipboard text can not be transferred to Word Juggler documents, as Word Juggler does not use the .CONSOLE driver for reading the keyboard. However, it is possible to transfer text from Word Juggler to the clipboard.

Please call or write for information on the *Desktop Manager Programmers Toolkit*. This package lets you write modules for the Desktop Manager. Full instructions and examples include our routines to put a folder on the screen and move it, our line input routine, the time and date routine and full *Desktop Manager* internal documentation.



“before”



“after”

## Low Prices!

**Desktop Manager . . . . \$129 + \$6 shipping**  
**Disk Utilities . . . . . \$44.95 + \$3 shipping**  
**ASCII Conversion Table . \$9.95 + \$3 shipping**

The *Desktop Manager* also has a clipboard, so you can transfer text from one screen or application to another. You may be doing calculations and want to transfer the results from the Calculator to a word processor. All you do is cut from the calculator and paste to the word processor file. Likewise, you can move an entire section of text from any file to the notepad or vice-versa.

The *Desktop Manager* was designed to be expandable. Here are some of the modules we will offer in the near future:

- Communications Package
- Spelling Checker (for /// E-Z Pieces)
- Graphics Charting
- Keyboard Macros

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

## 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.

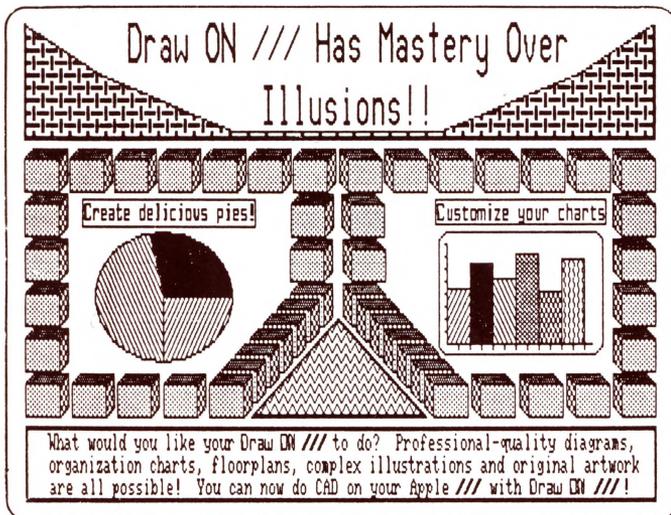
**Draw ON** gives an individual the power of a graphic arts studio. Use it in creating charts, preparation of slides and tables for presentation, and letterhead design. With **Draw ON** you can make changes to the dull graphics

## 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**.

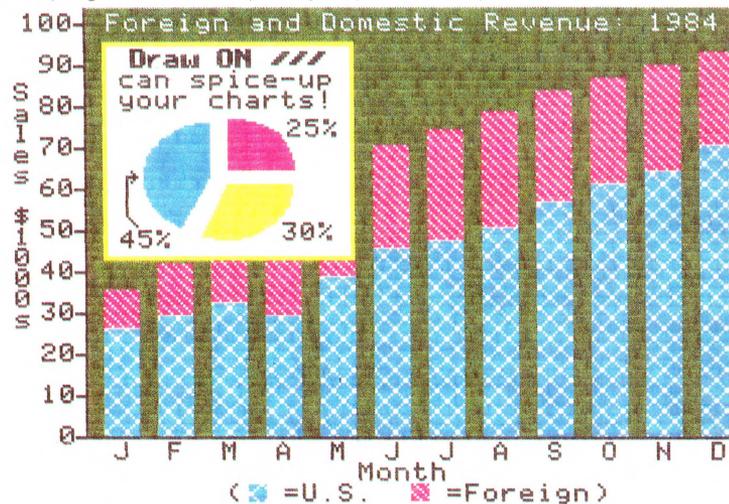
**Draw ON** combines powerful cut and paste facilities with the ability to mix text (in a variety of sizes and styles) with your drawings. If you don't like any of the text fonts or objects that come with **Draw ON** you can design your own! You can label your drawings with these fonts or even use them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