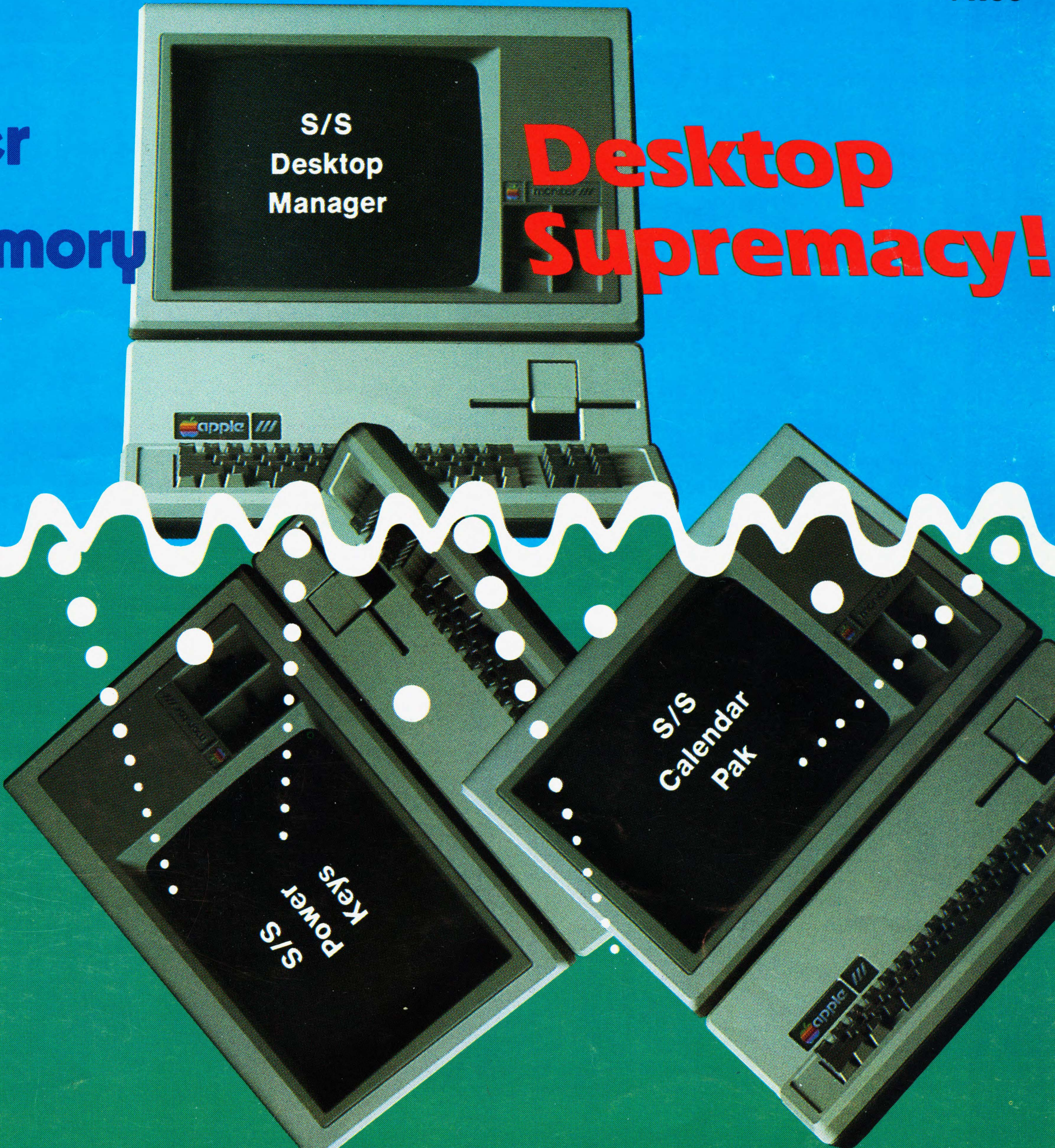
ON THREE®

The Magazine For Apple III Owners and Users

Volume 3 - Number 5

May 1986
$4.00

- Pascal Filer
- Screen Memory Allocation
- RAM disks



Desktop Supremacy!

- Query to Quark

- Basic Internals part II

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.**

"before"

"after"

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.

Low Prices!

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

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.

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

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.

Publisher:
Bob Consorti
Editor:
Val J. Golding
Comptroller:
Joseph Consorti
Logistics:
Janet Schanz
Interior Art:
Virginia Carol
Technical Support:
Rob Turner
Order Dept.:
(800) 443-8877 (toll free)
(805) 644-3514 (in Calif.)

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.

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*, Inc. Apple, Apple][, Apple][plus, Apple ///, Apple /// plus, Applesoft, Apple Business Basic, Disk][, Disk ///, UniDisk, Lisa, LisaDraw, Macintosh, MacPaint, MacWrite and ProFile are registered trademarks of Apple Computer, Inc. Micro-Sci and Gameport /// are registered trademarks of Standum Controls, Inc. Selector /// is a registered trademark of Sabre Software, Inc. ON THREE O'Clock, Lazarus ///, Draw On ///, Desktop Manager and ONTIME are registered trademarks of *ON THREE*, Inc.

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.

SUBSCRIPTION FORM

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

T.M.

The Magazine For Apple /// Owners and Users ®

/// /// /// /// /// /// /// /// /// ///

Volume 3, No. 5 **Table of Contents** May, 1986

FEATURES

DEPARTMENTS

ON: The Cover

A trio of Apple ///'s symbolizes new ON THREE products and market position.

Photographed especially for us, the Apple /// was captured by
David Ojerholm,
H & O Studios, 614 E. Main St., Ventura, CA 93001

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

ON THREE O'Clock

Now is the Time for a real-time clock

$49.95
plus $3.00 shipping and handling

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.

Desktop Manager/ON THREE O'Clock Combo

$173.95 plus $8.00 shipping and handling

The Editor Dishes it Out:

Apple.Sauce

val j. golding

Manager a Trois

A funny thing happened on the way to the meeting... Well, actually at the meeting. We were invited, along with **Bob Consorti**, *ON THREE's* Manager of Operations, and **Rob Turner**, to attend a local Apple /// user group meeting (co-chaired by **Mel Astrahan**, author of our *Graphically Speaking* series). Naturally, we jumped at the chance to show off the newly completed *Desktop Manager.*

It is curious indeed, how one can literally live alongside a developing program for months, offering occasional suggestions and observing new features, etc., and still not be aware of its full power or capabilities. This was our situation at the meeting as we watched first Rob and then Bob demonstrate some of the *Desktop Manager* modules. The Appointment Calendar was wondrous, and the NotePad fair competition to AppleWriter, but what impressed us (and the audience) was the Calculator.

Imagine if you will, a Monitor /// with a couple of file folders open and displayed, and a window with an image of a calculator—a display and all the keys, even an off/on key. Enter a few calculations, total them and press the key that turns on the paper tape... that's when all the ooh's and ah's came from the audience as a paper tape appeared on the screen, indistinguishable from what you would expect to see on a "real" calculator. Enter a few more figures, and the tape scrolls up. No big deal, really, but its stark realism wowed us.

Now for the punch line... while Rob was developing the paper tape concept and had it running for the first time, we got some *real* paper tape, and using scotch tape, attached it to the monitor at the exact point where the *Desktop Manager's* tape image ended, and let it drape over the back of the monitor, down to the floor. About this time, we called Bob: "Rob's got a new feature to show you..." Bob walked in, spotted the tape, and pandemonium broke loose. We fell to the floor laughing. "All in a day's work," we exclaimed.

Now a subject of less levity. Bob's *Block_Write* does not appear in this issue, having been ousted by a matter of some importance. We believe in good advertising, pushing our products, sometimes even, through honest comparisons, knocking others. But we also believe in fairness and not unwarranted personal attacks. If brand X does something better than our product, we'll admit it. But all's *not* always fair in love and war. An advertisement for a product competing with *Desktop Manager* recently appeared in another publication. To learn more about it, we direct your attention to the story *Controversy on the DeskTop* on page 4.

And Another Thing...

A subject those of you who use Word Juggler alone or in conjunction with our *512K Upgrade* board may find of interest is *Query to Quark.* We have been the target of several letters and calls after the February *Block_Write* mentioned that a 512K Word Juggler update was eminently available. Take a look now and read the exchange of correspondence and get the inside story.

Obviously, the whole of *ON THREE* is not filled with our tales of derring-do. For instance, on page 5, you'll find the concluding portion of *Business Basic Internals,* that exploration into the mysteries of BASIC and its interface to invokable modules. In this issue we describe the functions of most major internal subroutines.

Pascal beginners will find part two of **Dennis Cohen's** *ON Pascal* series, looking this time at the Filer and how to use it. Stick with this series each time around and you'll be a "pro" in no time at all.

In **Mel Astrahan's** *Graphically Speaking,* you'll find a learning experience as well. In the first of three parts, Mel describes the complexities of screen memory allocation. It is not, as one might suspect, a bunch of contiguous RAM memory, but a leftover from the original Apple][design (engineered for cost-effectiveness) of seemingly random organization of text lines interposed with groups of unused bytes. But let Mel present the whys and wherefores.

We introduce **Barry Downes** to our pages to tell of his wonderment in discovering a RAMdisk driver buried in our *512K Upgrade Utilities* disk. In *Looking at .RAM* Barry found a whole new world of fast disk I/O to aid him in loading macros into his word processing files. We're sure you'll find it fascinating.

We have a new game for you, *Space Convoy* by **Ron Puckett,** making his second appearance in *ON THREE*. In addition to being just plain fun, it is an interesting demo of how so few lines of BASIC can be used to create a game that will hold your (and your kids) attention. Those of you who program in BASIC will find intriguing Ron's use of special text font characters to create what appears to be graphic images on the text screen.

Sometimes the entire topic of printer interfacing can be confusing, so we have asked **John Lomartire** to step in and define some *Printer Protocols.* The main bent is the myriad connections between the Apple /// or a printer interface card, and the printer itself. You would think each numbered connector pin would be the same, regardless of manufacturer, but in fact there is only limited standardization. John's diagrams show how to "mix and match," required reading for anyone who has just purchased, or is considering purchasing, a new (or second) printer or modem, a subject we will be looking at next month.

We could go on and tell you about still more goodies, but we suspect that this column would end up looking like a "plan ahead" sign. So until next time, keep those cards and letters (and story ideas) coming.

Ciao.

A Note of Importance:

Controversy on the Desktop

on three

The February issue of *The /// Magazine* contained an advertisement by *D.A. DataSystems* that was highly uncomplimentary to *ON THREE* and insulting to the intelligence of Apple /// users in general. The entire staff of *ON THREE* believes and our attorney concurs that a number of statements in that advertisement were erroneous. The following is a statement of true facts that were mis-stated or not brought out in the referenced ad.

1) We resent being called the "Off & On Company." We have worked harder to bring more new and timely products to market for the Apple /// than any other company.

2) *The Desktop Manager*™ by *ON THREE* does not, as stated in the advertisement require "40K minimum plus more for planned modules." It in fact uses a maximum memory of 40K and in some circumstances consumes only 32K of the available system memory.

3) *The Desktop Manager*™ does indeed have many modules available for it. (The NotePad, Calculator, Appointment Calendar, Desktop Controller, cut and paste, mouse control, and built-in help screens are included.) Optional Disk Utilities, and an ASCII Table are among those available now, not the "many planned, call next week" as stated in the ad.

4) *The Desktop Manager* automatically installs itself on any 256K or greater Apple /// system and does not require running the System Utilities to add a driver. It is not, as stated in the ad, "another Pre-Boot (sigh. . .)"

5) In regards to *ON THREE*'s reputation, we take exception to the item stating *ON THREE* will have a reputation "real soon now." The dozen plus products that we have introduced over the last year and the new products that we have planned for production speak for themselves.

6) The statement in the ad describing *ON THREE*'s support as "file a report with missing persons" is ridiculous and defamatory. *ON THREE* prides itself on having the best customer support in the industry. Each *ON THREE* product shipped bears the statement: ***We guarantee satisfaction and full product support. Need help? Phone (805) 644-3514***. Just last evening, two of our people spent almost an hour assisting a caller to get Access /// to run on his Space Coast System hard disk using Catalyst. You'll notice that none of the products mentioned are our own. If this isn't full Apple /// product support, what is?

7) The statement describing people who buy *ON THREE* products as suckers, or there is "apparently one born every minute," may not sit too well with the thousands of Apple /// users who rely on *ON THREE* for all of their needs.

8) The price of the 512K Memory Expansion is not, as stated in the ad ". . .conveniently available from The Off & On Company (T.O.O.C.) for $499." The price of the 512K Memory Expansion remains $399, the same as it has been since December of last year when a price reduction took effect.

We would appreciate it if anyone who has read the D.A. DataSystems ad would write to us at *ON THREE* with their comments. If there are any questions, please do contact us at:

ON THREE, Inc.
4478 Market St., Suite 701
Ventura, CA 93003

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.

*A New **ON THREE** Service!*

☎

Toll-Free Order Line

(800) 443-8877

[In California call (805) 644-3514]

Yes! We accept Visa, Mastercard, and American Express (3% surcharge on American Express)

A new and convenient way to order *ON THREE* products. (A current *ON THREE* price list appears in each issue of *ON THREE* magazine.)

Sorry, *we must restrict this line to orders only. It has been established to offer a new convenience and a faster turnaround for our customers.*

Calls for technical support and all other matters should be directed to (805) 644-3514. Thanks for your understanding and cooperation.

A Look Inside:

Business Basic Internals

part ii

Internal Subroutine Descriptions

Routine 0—DOPAR

DESCRIPTION: DOPAR evaluates (as an expression) the tokenized ASCII data found at the current TXTPTR. The ASCII data must be terminated by either a colon or an EOL ($00). Most of the errors that can be generated in BASIC will cause an error exit when using DOPAR. New variables or arrays will be allocated if encountered during expression evaluation. VALTYP must be set from the values in the table below.

If VALTYP is set to request real data, then any numeric expression will return a real result. Any other type will exit with a TYPE MISMATCH error. The same follows for other variable types. If the any type ($20) is requested, numeric results will return as reals and strings will return as strings.

The results are left in the FAC in the following format:

Expression Type	VALTYP	Result, MSB First
Real (unpacked)	00	FACEXP—FACSGN
Long Integer	40	FAC—FAC + 7
String	FF	Pointer in FACMO, FACLO, FACMOXB

On entry: TXTPTR points to the expression to be evaluated.
VALTYP = requested type of expression ($20 means either type OK).

On return: FAC contains the result.
TXTPTR points to the terminator.
VALTYP set to type found.

Error exits: Almost all, too numerous to mention.

Routine 1—PTRGET

DESCRIPTION: PTRGET searches the variable storage tables for a variable and returns its address. The variable name pointed to by the current TXTPTR is found or created in memory and a pointer to its value is placed in VARPNT. This routine calls DOPAR to evaluate the array subscripts. DOPAR calls PTRGET to locate the variables. (Note the recursion.)

On entry: TXTPTR points to the variable to be referenced.

On return: VARPNT points to the variable's name.
VALTYP and INTFLG determine the variable's type.
ISARRAY will be 0 if simple variable, 80 if an array element.

Variable Type	VALTYP	INTFLG	Result MSB First.
Real (unpacked)	00	00	FACEXP - FACSGN
Long integer	40	00	FAC - FAC + 7
String	FF	00	Ptr in FACMO, FACLO, FACMOXB
Integer	00	80	FACMO, FACLO

Error exits: Since array subscripts can be expressions, most of the DOPAR errors can be generated.

Routine 5—ERRDIR

DESCRIPTION: Validates a program that is running and returns. Checks CURLIN.

On entry: None.

On return: In deferred mode, i.e., executing a BASIC program.

Error exits: If in immediate mode, ILLEGAL DIRECT ERROR.

Routine 6—LINGET

DESCRIPTION: Converts the ASCII line number pointed to by TXTPTR into a 16-bit integer in LINNUM, stored low- high-byte. The entry requirements are met by loading the A-reg via CHRGET or CHRGOT. LINGET will only accept unsigned integers in the range 0 → 63999. There must be a non-digit following the valid digits. LINGET will stop on the first non-digit it encounters and return the accumulated result. Leading spaces or zeros will be ignored.

On entry: TXTPTR points to the first ASCII digit of the line number
A-reg contains the first ASCII digit of the line number
Carry must be set

On return: LINNUM, LINNUM+1 is the 16-bit equivalent, low-high.
TXTPTR points to the terminating non-digit.
(Uses location INDEX as scratch area.)

Error exits: SYNTAX ERROR if number > 63999.
LINGET will only zero LINNUM, LINNUM+1 if carry is clear on entry.

Routine 7—GOTOB

DESCRIPTION: Searches the resident BASIC program for the line number given in LINNUM, LINNUM+1. Searches forward in the program if CURLIN is less then LINNUM, otherwise searches program from the beginning. [Probably uses TXTTAB]

On entry: LINNUM is the line number to locate.

On return: TXTPTR points to the terminator of the previous line.

Error exits: UNDEF'D STATEMENT error if LINNUM does not exist.

Routine 8—GETADR

DESCRIPTION: Rounds an unpacked real in the FAC to a 16-bit address (range 0 → 65535) in POKER, POKER+1, stored low-high.

On entry: FAC contains address as a real with exponent < 216.

On return: POKER holds address rounded to low-high 16-bit integer.

Error exits: ILLEGAL QUANTITY if the real is outside the range 0 → 65535.

Routine 9—FNDLNCO

DESCRIPTION: Searches through the BASIC program for the line number given in LINNUM. LOWTR is set to point at the link byte of the first line encountered with value greater than or equal to LINNUM.

On entry: LINNUM contains the line number to search for.
Y-X-A regs contain pointer to line byte of line to start search at. (Y-X-A = Xbyte, high, low.)

On return: LOWTR points to first line number whose number => LINNUM.

Error exits: None.

Routine 10—FNDLIN

DESCRIPTION: Same as FNDLNCO except that the search always starts at the beginning of the program.

On entry: LINNUM contains the line number to search for.

On return: (See FNDLNCO.)

Error exits: None.

Routine 13—NOTNOW

DESCRIPTION: Given a pointer in INDEX, INDEX+1 and INDEXB to a string descriptor, this routine leaves the string length in the A-reg and a pointer to the actual string data buffer in INDEX, INDEX+1, INDEXB.

On entry: INDEX points to a string descriptor.

On return: INDEX points to the string data.
A-reg holds the length of the string.
X, Y undetermined.
Status register preserved.

Routine 14—ERROR

DESCRIPTION: Raises the BASIC error condition given in the X-register. If ONERR is in effect and execution is deferred mode, then control is transferred to the error handler. Otherwise the error message is printed and control returns to immediate mode. This routine resets the stack and never returns.

On entry: X-reg = BASIC error code.

On return: Never returns.

Routine 15—SERROR

DESCRIPTION: Translate SOS errors given in Table VI to the corresponding BASIC error code if it is found in the table. Otherwise, issue SOS CALL ERROR 22.

On entry: A-reg contains SOS return code.

On return: Never returns.

Table VI

SOS error	BASIC #	BASIC error message
$10	30	FILE NOT FOUND
$25	38	RESOURCE NOT AVAILABLE
$27	25	I/O ERROR
$28	37	DEVICE DISCONNECTED
$2B	27	WRITE PROTECTED
$2E	28	DISK SWITCHED
$40	29	BAD PATH
$43	36	FILE NOT OPEN
$44	31	PATH NOT FOUND
$45	32	VOLUME NOT FOUND
$46	30	FILE NOT FOUND
$47	33	DUPLICATE FILE
$48	34	DISK FULL
$49	39	DIRECTORY FULL
$4D	26	FILE TOO LARGE
$4E	35	FILE LOCKED
$50	23	FILES BUSY
$51	24	NOT SOS
$52	24	NOT SOS
$54	07	OUT OF MEMORY
$57	40	DUPLICATE VOLUME
$58	16	TYPE MISMATCH

Routine 16—SCRUNCH

DESCRIPTION: Requests BASIC to release memory from its unused memory space back to SOS.

On entry: A-reg holds number of pages to release. (If A-reg = 0 then release all available memory.)

On Return: Memory now available via SOS memory management calls.

Error exits: OUT OF MEMORY if none available.

Routine 17—EXPAND

DESCRIPTION: Inverse of SCRUNCH. Requests BASIC to expand its memory by A-reg pages.

On entry: A-reg holds numbers of pages to expand memory.
(If A = 0 then expand to maximum available.)

On return: Memory reclaimed.

Error exits: If A <> 0 and BASIC can't allocate any free memory from SOS, then an OUT OF MEMORY error is generated.

Routine 20—FRECNOW

DESCRIPTION: When a string expression is evaluated and the string result is not assigned to a string variable, then the temporary string descriptor and string space must be freed after use. FACMO, FACMO+1, FACMOXB must point to the descriptor when this routine is called. This routine should be called after using STRCP to allocate a temporary work string.

On entry: FACMO points to a string's descriptor.

On Return: Descriptor, if it was a temporary descriptor.

Error exits: This routine should not take any error exits except that incorrect usage could step on memory anywhere.

Routine 22—OPENIT

DESCRIPTION: Evaluates a pathname and opens a file. TXTPTR points to a pathname or string expression. The file is created if the A-reg <> 1 and is checked to be of the type indicated by the X-reg. Opening a non-block device file must be done with TXTTYP ($04) as the file type or a TYPE MISMATCH ERROR will occur.

The file that is opened is only 'open' in SOS. It is not one of the ten BASIC files and is not accessible from the calling program. In addition, issuing a global CLOSE (no file number) in BASIC will not close the file since BASIC does not issue a global CLOSE_ALL command to SOS.

Thus, opening a file with OPENIT and then taking an error exit back to the calling program could result in the file being left open permanently if proper error processing is not present. A separate entry point just to close such a file is needed to allow the calling program to clean up after an error with the file still open.

On entry: A-reg = 1 requests *no file be created* if it does not already exist.
X-reg = SOS file type.
TXTPTR points to the pathname.

On return: A-reg = SOS reference number.

Error exits: All of the errors for GET_FILE_INFO, OPEN, and CREATE SOS calls (if A-reg <> 1) apply, with the following exceptions:

a) GFI error $58 (NOT BLOCK DEVICE) is handled by trying to open with file type TXTTYP, regardless of requested type.

b) If error $54 (OUT OF MEMORY), OPENIT will call SCRUNCH to free four pages of memory and retry the OPEN once more.

Routine 25—GIVAYF

DESCRIPTION: Converts the A-Y registers, input as a high-low signed 16-bit number, into an unpacked real in FACEXP-FACSGN. This usually is known as a FLOAT function but is not explicitly available through BASIC.

On entry: A-Y is a high-low signed 16-bit integer.

On return FAC is a real with the same value.
VALTYP = 0.
Error exits: OVERFLOW error should *never* occur.

Routine 26—POSINT

DESCRIPTION: Rounds a real and converts it to a positive integer.

On entry: FACEXP-FACSGN contain the value as an unpacked real.

On return: FACMO, FACLO contain the high-low 16-bit value if the real was in the range of 0 → 32767.

Error exits: ILLEGAL QUANTITY error if the real is outside the range for a 16-bit integer.

Routine 27—FIN

DESCRIPTION: Converts the ASCII string pointed to by TXTPTR into its unpacked floating point representation in the FACEXP-FACSGN. FIN will process only ten digits beyond the decimal point but will skip TXTPTR over digits in excess of ten. FIN will always leave TXTPTR pointing to the first non-floating point number that it encounters. There must be a non-floating point number as a terminating character to end the string. Floating point characters include the ten digits, period, plus, minus, and the letter "E".

On entry: TXTPTR points to ASCII representation of a real number.

On return: FAC contains the unpacked binary real.
VALTYP = 0
ANYNUM (location 0D) will be $FF if no digits were encountered and $00 if one or more were processed.

Error exits: OVERFLOW ERROR if string is outside the range of floating point numbers.

Routine 32—DATAN

DESCRIPTION: Searches forward in the program for the end of the current statement. Stops when it finds the end-of-line token ($00) or a statement separator colon ($3A). DATAN does not use CHRGET.

On entry: TXTPTR points inside a statement.

On return: Y-reg holds byte offset to the end of the statement from the unchanged TXTPTR value.

Error exits: None.
However, this routine will loop forever if a $00 or a $3A does not exist within 256 bytes ahead of the starting position indicated by TXTPTR.

Routine 33—STRCP

DESCRIPTION: This routine copies ASCII data into BASIC's string data area and builds a temporary descriptor pointing to the data.

This routine creates the data part of what BASIC calls a string, but it is a temporary one and is not assigned to a variable. If the temporary is to be assigned to a variable, then the INPCOM routine (#34) should be used to assign it. If it is not to be assigned, it must be freed up after use by the FRECNOW routine (#20).

On entry: Y-reg = length of string.
STRNG1, STRNG1+1, STRNG1XB is a pointer to the data to be copied.

On return: Temporary descriptor is built and is pointed to by FACMO, FACMO+1, FACMOXB.
VALTYP = $FF

Error exits: OUT OF MEMORY ERROR is possible.

Routine 34—INPCOM

DESCRIPTION: Assigns a temporary string to a string variable or duplicates an existing string and assigns a copy to the variable. Assumes that the value in FACMO, FACMO+1 and FACMOXB is a pointer to the input string's descriptor. FORPNT, FORPNT+1, FORPNTXB point to the descriptor of the output variable, and the variable's old string value is returned to the free memory pool. VARPNT is normally set by using PTRGET (#1) to locate the desired variable in the storage tables and may be used to fill FORPNT.

On entry: FORPNT, FORPNT+1, FORPNTXB point to the string variable.
FACMO, FACMO+1, FACMOXB point to the descriptor of the new value for the variable.

On return: All registers undependable.

Error exits: VARIABLE ERROR if the variable's descriptor is more than 64K from its storage table origin pointer.

Routine 35—LETP2

DESCRIPTION: Assigns a value to a variable. Takes the value in FAC - FAC+7 and the flags VALTYP, INTFLG and does the assignment into the variable pointed to by VARPNT. (See PTRGET (#1) for the data formats and associated flags.) This routine does not validate its input data and could store it anywhere in memory and therefore must be used with *extreme care*.

On entry: VARPNT, VARPNT+1, VARPNTXB point to the variable.
FAC - FAC+7 has the variable's new value or points to a string descriptor.
VALTYP and INTFLG control the assumed type of the variable and FAC.

On return: All registers undependable.

Error exits: None.
VALTYP and INTFLG do not reflect the actual type of the variable at VARPNT, or if they do not reflect the type that is in FAC, the value will be stored as if it were the proper type anyway.

Routine 39—INT

DESCRIPTION: Converts the real in FACEXP - FACSGN into a real in FACEXP - FACSGN without a fractional part. Same as the INT function in BASIC. Only operates on FAC if its FACEXP has a magnitude < 231 (8,388,608).

On entry: FAC has a real number.

On return: FAC has an integer in real format.

Error exits: None.
Does nothing if FACEXP magnitude invalid.

Routine 44—CONV2STR

DESCRIPTION: Takes the FAC and converts it to an ASCII string for any valid VALTYP. Same as CONV$ function in BASIC. If the FAC is a real or Long Integer then a STR$-like operation is done, leaving a pointer to the string descriptor in FACMO, FACMO+1, FACMOXB.

This routine does nothing if VALTYP is already a string VALTYP ($FF). The FAC can *not* contain a short integer with VALTYP = $00 and INTFLG = $80; use GIVAYF to convert it to a real first.

On entry: FAC has some value determined by VALTYP.

On return: FAC has the value expressed in string form.
VALTYP = $FF.

Error exits: None.
The data in FAC must correctly match the given VALTYP.

```
;*********************************************************
;
;       This procedure will upshift lower case letters in
;             a string to UPPER case letters.
;
;*********************************************************
;
INDEX           .EQU     35          ;Pointer in zero page
INDEXB          .EQU     INDEX+1601  ;Pointer extend byte
DISPATCH        .EQU     0E4
NOTNOW          .EQU     13.         ;Interpreter subroutine # in decimal
;
                .PROC    UPSHIFT,1
;*********************************************************
;
;          Procedure name:  UPSHIFT
;      Parameter word count:  1
;
;*********************************************************
;
                PLA                  ;Pull off and save RETURN address
                TAX
                PLA
                TAY
                PLA
                STA      INDEX       ;Pull & save pointer to string vbl
                PLA
                STA      INDEX+1
                TYA                  ;Restack RETURN address
                PHA
                TXA
                PHA
                LDA      16E9        ;Get pointer extend byte
                STA      INDEXB      ;Save it in the INDEX extend byte
                LDA      #NOTNOW     ;Get the routine number
                STA      DISPATCH+3
                JSR      DISPATCH    ;Puts pointer to string in INDEX
                TAY                  ;Move length of string to Y-reg
                BEQ      RTN         ;If null string, do nothing and rtn
LOOPHERE        DEY                  ;Finished last byte?
                PHP                  ;Save status for test later
                LDA      (INDEX),Y   ;Look at a byte of the string
                CMP      #"z"+1      ;Is it lower case alpha?
                BCS      NOTLC       ;No, forget this byte
                CMP      #"a"        ;Could be...
                BCC      NOTLC       ;No, it's a special chr, forget it
                SBC      #20         ;Make it into UPPER case alpha
                STA      (INDEX),Y   ;  and put it back in the string
NOTLC           PLP                  ;Was that the last byte?
                BNE      LOOPHERE    ;No, go do them all!
RTN             RTS                  ;Yes, return to BASIC
;
                .END
```

For those of you who want to get the most from Business Basic *by writing invokable modules in assembly language, we hope this two part series has been a treasure trove and a source of inspiration. Additional copies of the entire article are available in photocopy form for $5, postpaid.*

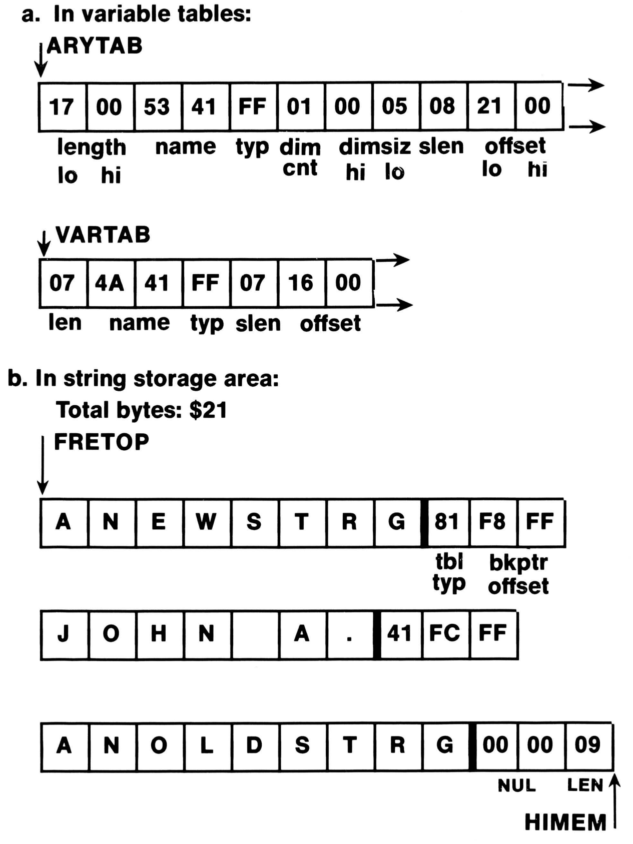
As additional information is developed, it will be published in ON THREE *in whatever form is appropriate. We would like to encourage all who have additional information concerning Business Basic internals, or who have the yen to engage in further research, to pass their information on to us. Additional time spent exploring zero page pointers should result in filling in some of the remaining blanks and question marks from Table V on page 12 of the April* ON THREE. *Any contributions printed or included in updated information will be duly credited.*

Thanks for your help. . . . ed

Table VI. Business Basic Keywords and Tokens

Token(s)	Keyword	Token(s)	Keyword
$80	END	$C0	REM
$81	FOR	$C1	STOP
$82	NEXT	$C2	ON
$83	INPUT	$C3	=
$84	OUTPUT	$C4	LOAD
$85	DIM	$C5	SAVE
$86	READ	$C6	DELETE
$87	WRITE	$C7	RUN
$88	OPEN	$C8	RENAME
$89	CLOSE	$C9	LOCK
$8A	=	$CA	UNLOCK
$8B	TEXT	$CB	CREATE
$8C	=	$CC	EXEC
$8D	BYE	$CD	CHAIN
$8E	=	$CE	=
$8F	=	$CF	=
$90	=	$D0	=
$91	=	$D1	CATALOG
$92	=	$D2	=
$93	WINDOW	$D3	=
$94	INVOKE	$D4	DATA
$95	PERFORM	$D5	IMAGE
$96	=	$D6	CAT
$97	=	$D7	DEF
$98	FRE	$D8	=
$99	HPOS	$D9	PRINT
$9A	VPOS	$DA	DEL
$9B	ERRLIN	$DB	ELSE
$9C	ERR	$DC	CONT
$9D	KBD	$DD	LIST
$9E	EOF	$DE	CLEAR
$9F	TIME$	$DF	GET
$A0	DATE$	$E0	NEW
$A1	PREFIX$	($E1) $FF 80	TAB(
$A2	EXFN.	($E2) $FF 81	TO
$A3	EXFN%.	($E3) $FF 82	SPC(
$A4	OUTREC	($E4) $FF 83	USING
$A5	INDENT	($E5) $FF 84	THEN
$A6	=	($E6) $FF 85	=
$A7	=	($E7) $FF 86	MOD
$A8	=	($E8) $FF 87	STEP
$A9	=	($E9) $FF 88	AND
$AA	=	($EA) $FF 89	OR
$AB	=	($EB) $FF 8A	EXTENSION
$AC	=	($EC) $FF 8B	DIV
$AD	POP	($ED) $FF 8C	=
$AE	HOME	($EE) $FF 8D	FN
$AF	=	($EF) $FF 8E	NOT
$B0	SUB$	($F0) $FF 8F	=
$B1	OFF	$F1	=
$B2	TRACE	$F2	=
$B3	NOTRACE	$F3	=
$B4	NORMAL	$F4	=
$B5	INVERSE	$F5	=
$B6	SCALE(	$F6	=
$B7	RESUME	$F7	=
$B8	=	$F8	=
$B9	LET	$F9	=
$BA	GOTO	$FA	=
$BB	IF	$FB	=
$BC	RESTORE	$FC	AS
$BD	SWAP	$FD	SGN(
$BE	GOSUB	$FE	NOT
$BF	RETURN	$FF	See above

Tokens in the range $E1-EF appear to be unused and are in fact replaced by tokens in the range $80-8F when preceded by a $FF token figure 12. String Storage Method



a. In variable tables:
ARYTAB
17 | 00 | 53 | 41 | FF | 01 | 00 | 05 | 08 | 21 | 00
length lo hi
name
typ
dim cnt
dimsiz hi lo
slen
offset lo hi
VARTAB
07 | 4A | 41 | FF | 07 | 16 | 00
len
name
typ
slen
offset
b. In string storage area:
Total bytes: $21
FRETOP
A | N | E | W | S | T | R | G | 81 | F8 | FF
tbl typ
bkptr offset
J | O | H | N | | A | . | 41 | FC | FF
A | N | O | L | D | S | T | R | G | 00 | 00 | 09
NUL
LEN
HIMEM

CROSSWORD-SCRAMBLER

is a computer program that is educational and makes learning fun. Unlike many software products, **CROSSWORD-SCRAMBLER** lets you 'use your brain'. No, it's not a "shoot 'em up" type of arcade game. although you won't be disapointed| by the graphic displays and musical interludes. Instead, if you like being human and would like to work with a computer (rather than SUBJECT yourself to one), then **CROSSWORD-SCRAMBLER** is what you have been waiting for.

Hundreds of different crossword questions will provide hours and hours of fun for the entire family. With **CROSSWORD-SCRAMBLER** you can turn your **Apple ///** into a true **Personal** Computer - one that is both powerful and entertaining.

CROSSWORD-SCRAMBLER is sold exclusively through **ON THREE** Magazine for only $19.95.

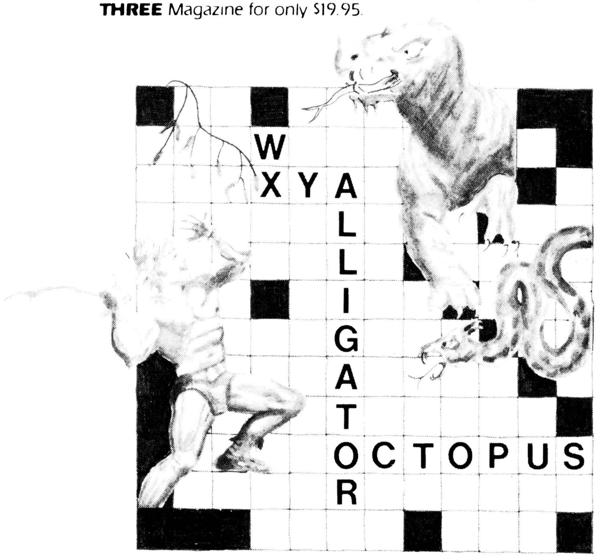
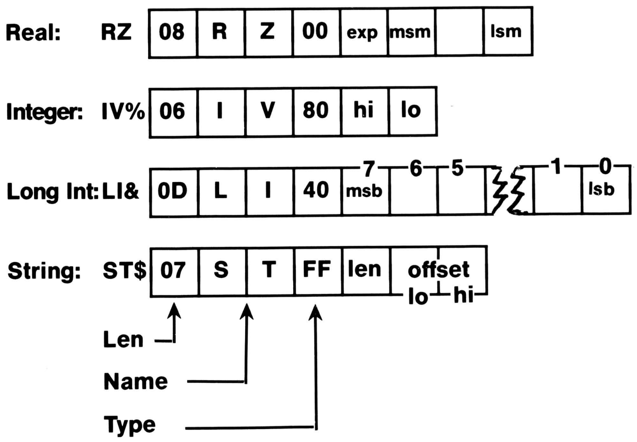
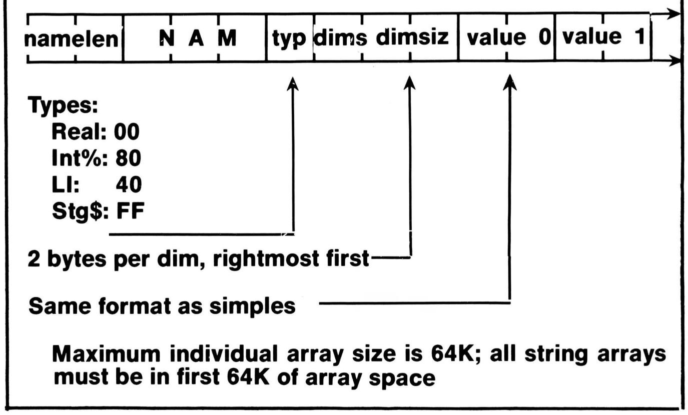


figure 11. Variable Format



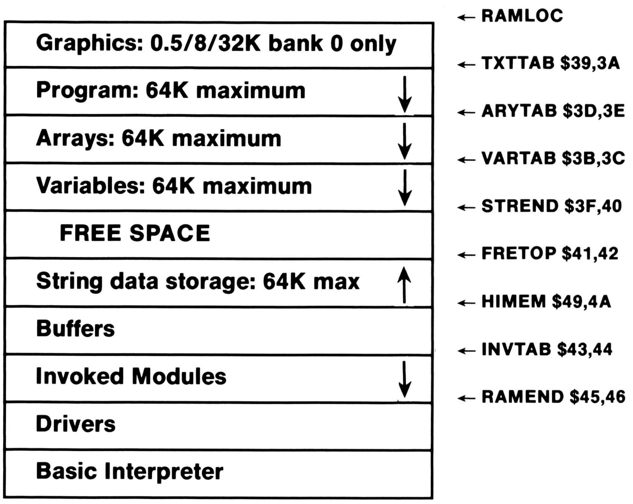
Real: RZ
08 | R | Z | 00 | exp | msm | | lsm
Integer: IV%
06 | I | V | 80 | hi | lo
Long Int: LI&
0D | L | I | 40 | msb | | | | lsb
7
6
5
1
0
String: ST$
07 | S | T | FF | len | offset
lo
hi
Len
Name
Type

figure 13. Array format



namelen
N A M
typ
dims
dimsiz
value 0
value 1
Types:
Real: 00
Int%: 80
LI: 40
Stg$: FF
2 bytes per dim, rightmost first
Same format as simples
Maximum individual array size is 64K; all string arrays must be in first 64K of array space

figure 14. Business Basic Memory Map



A Special Group of Friends:

/// People

richard/lavona rann

Apple /// owners and users have good reason to believe in two old sayings: "Hardship is the fertile soil of friendship," and "You find out who your friends are when the chips are down."

How many of us have found that many dealers, developers and writers who were "sold" on the /// just turned their backs when Apple moved away from the machine? There was a lot of gnashing of teeth, but now that some time has passed, we know that we do have good friends. Think about it. Who has been providing software, hardware, advice, articles and support for us all? Well the group is actually quite large, although much of the support is from private individuals and small companies, rather than those we originally were led to believe were our "friends."

You are reading this in a magazine devoted solely to the Apple /// and published by a group that markets several hardware and software tools designed for the ///. That in itself says a lot.

During the "hard times" a couple of years ago, the word had not gotten out and a lot of /// owners and users had just cause to feel abandoned. For a while it looked like the user and support groups also would and should fold. I personally can remember when /// user groups were compared to Edsel fan clubs. 1985 was a turnaround year for /// owners. The ///community mobilized to serve its own needs and we developed a family of dependable friends.

/// owners and users now have a group of developers (Anderson, Astrahan, Consorti, Turner, and Wade, to name a few), more than one regular publication, several useful user organizations, and an active and highly visible SIG (Special Interest Group) on Compuserve. If you think back, the /// community is better organized and more professional than it was when the Apple /// was an official member of the Apple family.

Third Apple Users (TAU), the largest Apple /// user group, is proud to be among those who emerged unscathed and improved from those dark times.

TAU's roots go back to when the /// was new. A group of people organized a Chicago area users group in 1982. In the early years it never got much bigger than 30 to 40 members, but it held monthly meetings, published a short newsletter and offered classes (including a SOS internals class). By early 1984, the group was beginning to thrive and planned on having "sections" to handle increasing membership and the "large" territory of the Greater Chicago metropolitan area. You all know what happened then, Apple pulled the plug on the Apple ///.

As might be expected, enthusiasm waned. A number of relatively active members decided to disappear and we began to question whether it made sense to continue with a user group for a "dead" machine. The next few months were a major turning point for TAU and for /// owners in general. It would have been easy for the /// and its community to die as expected.

Our true friends, names that come to mind now when you think of active /// supporters, made a decision to support the /// community for as long as it should exist. After all, the /// still did everything it was bought to do and there was no reason to assume that the ///'s (and their users) had become obsolete. We were to learn that we were correct: the /// was much too strong to disappear just because the popular press and Apple declared it to be dead.

TAU made some changes. A decision was made to enlarge the newsletter and emphasize membership benefits that would be of value to all ///'ers irregardless of their geographical location. Membership was opened to other /// organizations and additional services developed. We moved away from being a small local group and consciously decided to provide support and a communications link for the entire community.

Today, TAU is an international organization of /// owners, clubs, and various others interested in the ///. Within the group there are consultants, software developers, and people from Apple.

TAUTALES, the monthly newsletter has been published since 1982 and has not missed a monthly issue since September, 1983. The TAU library contains over 50 diskettes available to members for $3.50 plus shipping.

TAU has a helpline, (much like *ON THREE's Call Three: Hot Line*). Through it, TAU tries to provide a central clearing house for questions concerning Apple /// hardware and software problems and issues for all /// owners and users.

Recently, the *TAU Helpline* got a call from someone who had bought a /// from a friend who later moved out of town. It came with neither diskettes nor manuals. The new owner was trying to find out how to make the machine work. It took very little time for TAU members to gather together what he needed to get going and get his money's worth and more out of his newly acquired ///. He is now ready to help others in a similar position.

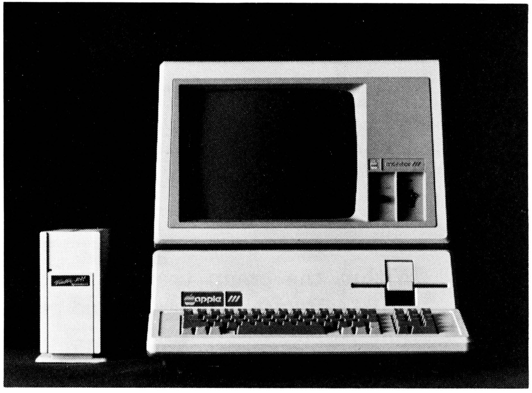
Although members benefit from the newsletter, Helpline, library and group purchases, one of the most valued benefits is the sense of belonging that one gets when dealing with other members. We are a close-knit family whose bonds were forged by the hardships and lonely times of 1984. Here, other /// owners and users can find a group of understanding people who have been through the trauma of desertion.

A belated welcome back to all the folks at *ON THREE* from TAU. We have supported and enjoyed the magazine from its inception and are happy to see our friends meeting with success. /// owners don't have far to look for friends; they are widespread and proven.

For TAU membership information and a sample copy of TAUTALES, send your name and address to:

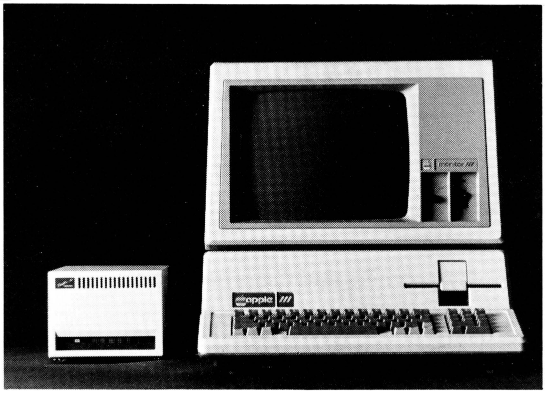
TAU c/o Lavona Rann
1113 Wheaton Oaks Drive
Wheaton, IL 60187

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][. 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][. 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.

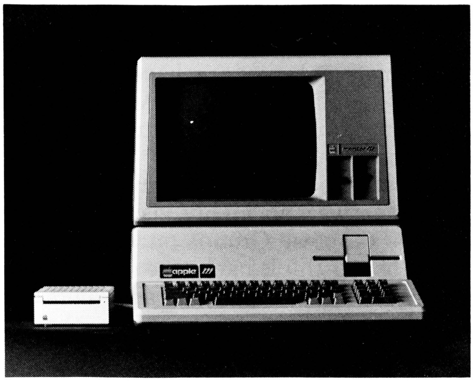
UniDisk ///.5 800K Micro-Floppy

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.**

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).

To order, call (805) 644-3514 or write:
ON THREE, Inc.
Attn: Order Dept.
Post Office Box 3825
Ventura, CA 93003

The Filer:

ON Pascal ///

dennis cohen

Well, we're back with the second installment on how the Apple /// Pascal system will be of benefit to you and how you use it. In the first article we covered (quickly and superficially) what the different options are on the command line. This time we're going to go into more detail on one of those selections, the Filer.

The Filer is probably one of the best examples of the philosophic differences between a system such as Business Basic and a system such as UCSD Pascal. In Business Basic (or most BASICs for that matter) you have a command like CATALOG that lists the files in a specified directory and this command is available from the system prompt or from inside a program; however, the other normal file and directory maintenance operations (like copying files or volumes, renaming files, or deleting files) are absent, requiring that you obtain or write utilities to accomplish them. BASIC also does not provide a pattern matching mechanism (wildcards) as do other operating systems such as CP/M, MS-DOS, or Unix. The Apple /// Pascal System follows the philosophy that all of these functions are related and provides a single utility, the Filer, to allow their operation. Internal to the Filer, there are also some pattern matching conventions which can be used to select a subset of the available files in a directory. When we get further into the series, we will see how Units are provided (or created) to allow you to perform these same functions from within the Pascal programs that you write.

Filer Commands

The Filer is reached by typing an "F" at the "Command" prompt. This tells the Pascal system to load the codefile "SYSTEM.FILER" from whichever disk contains it, so it need only be online somewhere. The Filer allows you to copy files or volumes (Transfer), delete files (Remove), empty a directory (Zero), create and allocate space for new files or subdirectories (Make), list a directory (List & Ext-dir), rename files and directories (Change), modify and/or write-protect files (Alter), list online devices, perform various operations on the "Workfile" or Apple][Pascal formatted disks, and many other functions. Table I contains a complete list of the available Filer commands and a brief summary of what they do. The following will be a more thorough discussion of the Filer's various commands and will conclude with a discussion of "wildcards," a way of abbreviating so as to only operate on a subset of the available files.

L(ist and E(xt-dir

"List" is the equivalent of the Business Basic "Catalog" command, it lists which files are in a given directory. "Ext-dir" gives a list of the contents of a given directory and all of its subdirectories (and their subdirectories, etc.). Some of the differences you will notice in one of these directory listings from those you are familiar with via "Catalog" or the System Utilities are the names of the various file types. What you are used to seeing as PasText is referred to as a TextFile and what BASIC refers to as a Textfile is referred to as an AsciFile. PasCode files are called CodeFiles and BASIC program files are BasProg files. The layout of the listing is also somewhat different, but clearly marked.

You can use wildcards to specify listing only a subset of the files in a given directory. For example, to list all files in the root directory of .PROFILE which start with "S", just answer ".PROFILE/S=" to the prompt, "Directory listing of what volume?" (Try doing that from BASIC.)

TABLE 1 **Apple /// Pascal Filer Command Quick Summary**

Informational Commands:	
L(ist	List the files in the current SOS directory Abbreviated directory listing for an Apple][Pascal disk
E(xt-dir	List the files in the specified SOS directory and all subordinate directories Full directory listing for an Apple][Pascal disk
V(olumes	Tells what drivers are associated with which Pascal device numbers and what volumes are in each drive. Also tells current prefix and what the system volume is.
W(hat	Tells what the current workfile is.
Disk/Directory Maintenance Commands	
K(runch	(Apple][disks only) Collect free space.
X(amine	(Apple][disks only) Create .BAD files containing damaged blocks.
Z(ero	Mark all files in the directory as deleted.
R(emove	Mark a specified file (or group of files) as deleted
C(hange	Rename files/volumes.
A(lter	Change the write-protection, last modification date, and/or the file type of a file (or group of files).
M(ake	Create a subdirectory or file (allocates an initial size).
T(ransfer	Copy files, subdirectories, and/or volumes.
B(adBlocks	Check a disk for bad blocks (Verify a volume)
Miscellaneous Commands	
N(ew	Create a new workfile (clears old one).
S(ave	Save the old workfile and name it -- should be performed before a new if you want to keep your work.
G(et	Treat an already existing file as the workfile.
Q(uit	Leave the Filer and go back to the main command line.

Unless you specify otherwise, the output from the "L" and "E" commands go to the Console device. You can send the output to another device or to a file. If you wanted the directory listing of .PROFILE to go to the printer, you would answer ".PROFILE,.PRINTER" (assuming that your printer driver was named .PRINTER). If you wanted it to go to a file, you would type the filename after the comma. Be sure to specify a ".TEXT" extension to the filename. If you don't, it will be cataloged as a DataFile. It is usually a good idea to list to a file other than one in the directory being listed if you are interested in such things as available space, since a file of size zero is opened at the time the directory is read and the size doesn't get updated until the file is closed.

V(olumes

"Volumes" is a fairly simple command. When you invoke it, you receive a list of the volume and device names associated with the various Pascal unit numbers. The Pascal unit numbers are a holdover from the UCSD p-System from which our system evolved. Certain unit numbers are essentially predefined. Units 1 and 2 are both listed as .CONSOLE; however, in the old days they were CONSOLE and SYSTERM. Unit one is the screen/keyboard combination, unit two is just the keyboard. When we get into Pascal programming (next time), you will see that by using the keyboard alone you can prevent what is typed from being displayed on the screen. Unit three will be .GRAFIX if you have the driver in your SOS.DRIVER file on the boot disk, otherwise it will be unused. Unit four will be the built-in disk drive unless you have run PMOVE or are operating under a CATALYST-type program, but it will always be the "System Volume" (Root volume). Unit five will be the next block structured device (usually the first external disk drive; if you've run PMOVE, it will be the internal drive). Unit six is the Printer driver (if you have one). Units seven and eight are both listed as .RS232 if you have that driver configured, unit seven being the input channel (REMIN: in UCSD) and unit eight being the output channel (REMOUT:). Units nine through twelve are set aside for block-structured devices (extra drives, hard disks, etc). Other devices get numbered from 128 on as the intervening numbers are reserved for future use. Some examples of these other devices might be .AUDIO, a second printer driver, or a format driver. The one thing of which you must be careful is that you don't have more than six block-structured devices active. Nothing terrible will befall you or your system, you just won't be able to access any after the sixth, Unit 12. This is the reason that System Utilities will not operate on block devices after the sixth (it's a Pascal program).

T(ransfer

"Transfer" is the command that I probably use as much as any other. With this simple command, you can copy files from one disk to another, back up disks, and list files to the screen or printer (or even a modem). If you have a sense for the ridiculous, you can even find out how a file "sounds" by transferring it to ".AUDIO". The last is somewhat unusual, but it is a real possibility (I even did it a couple of times for laughs).

The Transfer command is similar to the MS-DOS COPY command, but extends the limited power of that command greatly. With Transfer, you can copy the subdirectories as well (MS-DOS doesn't do that) and includes the MS-DOS DISKCOPY utility as well for copying entire volumes.

R(emove

"Remove" is relatively straightforward. You type "R", the Filer asks which file(s) you want to remove, and you reply. The Filer then notifies you of each file that it has removed (or failed to remove due to write-protection), and then asks for confirmation before the deletion is made final (i.e. that the program, SYSTEM.FILER, writes out the directory changes and has SOS modify the block bit map).

C(hange

"Change" allows you to rename files and/or volumes. The filer will prompt you for the name(s) of the file(s) that you wish to change and to what you want to rename them. Just answer the questions.

The Change command differs from the Transfer command in one significant way. If you do not specify a directory name to the destination file prompt in the Transfer command, the prefix directory (or subdirectory) will be assumed; with the Change command, it will be the path that appears in response to the source file prompt (even if you do specify a path to the destination prompt). What this means is that Change only works within a given directory, it does not copy files to other directories. This demonstrates a difference from the mv command of Unix-like systems in that mv will rename within a directory, but copy and delete between directories.

A(lter

"Alter" gives you the capability to modify file parameters. By this I mean that you can turn write-protection on or off, change the file type as reflected in the directory (such as from DataFile to AsciFile), or change the last modification date as reflected in the directory. The filer will prompt you for the name of the file(s) to be modified. After you have answered that, you will be prompted as to whether you want to change the write-protection status. If you answer affirmatively, you will be asked whether you want protection turned on. After the prompt for write-protection, you will be prompted as to whether you want to change the last-modification date. If your answer is yes, you will be asked for the new date. The date must be in the format dd-mmm-yy (two digit day of month, first three characters of the month, and the last two digits of the year—separated by hyphens) and you will be asked for it over and over until you answer in this format. The final thing for which you will be prompted is whether you wish to change the file type. The types which you may specify are: asci, badf, basicdata, basicprog, code, data, and text. For any of the non-Basic types, you may type in longer names for the type but the filer will look at just the first four characters. If you accidentally invoke alter or want to leave the alter command early, you can do so by pressing [ESCAPE] at any prompt.

M(ake

"Make" is the command you specify when you wish to create a new file or subdirectory, allocating an initial size, but not necessarily putting anything in it. Suppose you have a disk in drive .D2 and you wanted to create a file on it named MYFILE that was 10K (20 blocks) in size. You would invoke the Filer, and type M. The Filer would respond with, "Make what file?" You would then type in, ".D2/MYFILE[20]". Remember, you don't type in the quotation marks—they're just there to delimit what you type. If you had wanted to create a subdirectory named MYSUBDIR, you would have typed in ".D2/MYSUBDIR![n]" where you would replace the "n" with an integer specifying the number of blocks to be allocated for the subdirectory. The exclamation point says that a subdirectory is to be created—if you leave it off, you will just get a file. If you omit the [n], you will get a subdirectory that is 1 block long (it will have room for twelve files). If you omit the size specification for a file, you will get a file that is zero blocks long.

Because of the way a number of parts of the system are written (Assembler, Compiler, and Editor), the Pascal system follows a convention regarding filetypes and naming conventions. A file whose name ends in ".TEXT" is assumed to be a TextFile that follows various internal formatting conventions. One of the consequences of this is that a Pascal TextFile is always at least four blocks in length and is an even number of blocks in length. The Make command protects you somewhat in allocating space for TextFiles, although its error message is far from clear. If you try to create a .TEXT file of less than four blocks, it will not let you, but the error message is that there is no room left on the volume. If you create a .TEXT file with an odd number of blocks, you will get no error message, but will get a TextFile allocated that is one block smaller than you specified (i.e. telling it 17 will get you 16).

D(ate

The "Date" command allows you to reset the time and date. This is especially useful when you don't have a clock or when you need to set your clock (maybe Daylight Saving Time just started or ended) or just when you want to know what time it is. Because of a bug in the clock chip (or SOS's interaction with it), when January 1 rolls around, the year doesn't increment. Daylight Saving Time and New Years Day are the only times that I've used this command since getting my *ON THREE O'Clock*.

P(refix

The "Prefix" command serves a double purpose. It starts out by telling you what the current prefix is and then asks for a new prefix. If you just hit [Return], you are left with the same prefix. If you reply with "*", you get the root volume back as the prefix volume. Any other directory name, subdirectory name, or device designation (if SOS can find it) will make that the default directory.

Z(ero

"Zero" is a command to remove all files in a (sub)directory. The Filer will prompt for verification before it reinitializes the directory. This command (and the Remove command) do not actually erase the data, they just mark the directory entry as "deleted" and let SOS rewrite the block bitmap. This is the reason that a utility such as *Lazarus ///* will frequently be able to restore deleted files. It needs to go to the directory entry which is marked as deleted, find the pointer there to the file's block map and update the volume block bit map appropriately (this is not as simple as it sounds, since there are a number of safety checks that need to be implemented if you don't want to trash your disk or other files).

B(adBlocks

"Bad Blocks" is an alias for "Verify a volume" in the System Utilities. You will be prompted for the name of a device or volume. At this point, the Filer will start to check the integrity of the disk. If it finds bad blocks, you will be notified as to which blocks they are and asked whether to continue after the third, ninth, eighteenth, twenty-seventh, etc., bad block that is located. If at any time, you reply "N", you will be advised of any endangered file(s). Assuming that you found some bad blocks, you should try transferring the good files to another disk and then seeing what you can do to recover the endangered file(s) or parts thereof. After that, the best course is usually to just reformat the disk. If the reformatting shows verification errors, I would recommend trashing the diskette. If it is a hard disk, you should have a utility which "spares" bad blocks and you should run that utility.

N(ew

"New" is how you create a new workfile or specify that the currently associated workfile should no longer be so designated. You would do this if you just want to get rid of all files on the system volume that are SYSTEM.WRK or if you no longer want the currently designated file to be Run, Compiled, or Edited by default.

S(ave

"Save" causes the system to save both the Text and Code files (if they exist) under the currently designated name or under a new name that you specify. If you save into the root directory of the system volume, the system just renames the workfile accordingly. If you save into another directory, the Filer performs a Transfer which means that you still have the SYSTEM.WRK file(s) in the root directory of the system volume. After doing a save, I always do a new. This doesn't occur as often as you would think as I seldom (almost never) use a workfile. If you are working with only one source file, the workfile is not such a bad thing; however, almost any real program consists of more than one source file (the others are called Units and Include Files) and you will frequently be switching between them. This save command should not be confused with the Editor's Save command which I will discuss below.

G(et

When you "Get" a file, you are simply designating that a currently existing file should be used by default when editing, compiling, and running.

W(hat

"What" tells you just that. It gives you the name of the currently designated workfile and whether it has been "Save"d.

All of the workfile commands are holdovers from UCSD Pascal's origins as an academic tool. Student programs tend to be small and they tend to only be working on one at a time, therefore they used the workfile and did not need to remember the name of the file on which they were currently working.

K(runch

"Krunch" is used only with Apple][Pascal format disks. For those of you who are not familiar with the differences, an Apple][Pascal disk does not use the SOS style of scattering the file across the disk, but rather allocates contiguous blocks for each file. This saves a few blocks in overhead since you no longer have to keep track of the block bit-map for the files that are bigger than one block; however, if you start deleting files you begin to get gaps on the disk which are unused. Recently, I was looking at an Apple][disk that had 61 free blocks, but no gap was more than three blocks in length. This rendered the free space essentially useless. When you "Krunch" the disk, all the files are moved toward the "beginning" of the disk unless you specify otherwise. The main reason to specify otherwise is that you want to enlarge a file and there is no free space following it. If you didn't free up the space with a Krunch, you would either get an error message or overwrite part of the following file depending on how you tried to extend the first file. The normal Krunch results in a contiguous block of all the free space at the end of the disk.

X(amine

The other Apple][-specific command in the Filer is "eXamine." It attempts to repair bad blocks as reported by the Bad Blocks command. If it is unsuccessful, it allows you to mark them as bad. What this means is that it removes the file(s) containing bad blocks and creates directory entries by the name BAD.00nnn.BAD where nnn is the block number that was bad. These "files" are not shifted by a Krunch. Because of this, the dangerous areas are rendered effectively harmless. Later in this series, we will write a program that performs a similar function for SOS volumes. It will create a file called BadBlocks (if it doesn't already exist) and append bad blocks to the file. (Bob, when you read this, it would be a nice feature to include in some future version of Lazarus).

Q(uit

The last Filer command is "Quit." As you've probably guessed, this just turns you to the system command line.

Those Filer commands that present you with two prompts, one for a source file and one for a destination file (Transfer and Change), allow you to avoid the second prompt by appending a comma and the prompt to the second prompt to the prompt to the first prompt. For example, if in a Transfer command you want to copy the file PROG1.TEXT to the file .D2/PROG1.TEXT, you could answer the "Transfer what file?" prompt with the response, "PROG1.TEXT,.D2/PROG1.TEXT".

Wildcards

You will frequently need or want to perform the same filer operation on more than one file at a time. The Filer supports a "wildcard" concept which facilitates this (saving you a lot of typing).

The first two characters are not really wildcards, but are special characters that are recognized by the entire Pascal system. They are "*", the shorthand notation for the system volume (root directory), and ":", the alias for the current prefix directory. The second is not particularly useful and is primarily a holdover from UCSD, where it is also not too useful (maybe someone out there who knows the history of UCSD can inform us as to the reason for this shorthand).

Wildcards are of the following two forms:

<str1>=<str2> or <str1>?<str2>

where either or both strings may be omitted. The first form means any file that starts with <str1> and ends with <str2>. Actually, the second means the same thing as the first with the added proviso that you be prompted on each match as to whether to perform the operation. There is another special character to the Filer, "$". This is recognized only in destination filenames (those on the "to" or "as" side of a Filer command). An example sequence would be:

```
T
Transfer what file?=
To what file?.d2/$
```

The text that is underlined are the Filer's prompts to you. This sequence means, "Transfer all files in the current (prefix) directory to .D2, keeping the same name." From this, you've probably realized that the dollar sign means "by the same name."

The question mark may only appear in response to the first prompt. The equal sign may appear in prompt to both prompts, but may only appear in response to the second if either the equal sign or question mark was used in response to the first. As a matter of fact, if a wildcard is used in response to the first prompt, the equal sign must be used in the response to the second prompt (unless you use the dollar sign). If you wished to transfer only the textfiles from the current directory to .D2, you would answer the first question with "=.TEXT". If you wanted to copy all the textfiles to backup files, you would answer the first question with "=.TEXT" and the second with "=.BACK". In this case (and all others), the Filer treats the equal sign in response to the second prompt as meaning "...those characters that were filled in..."

Only one wildcard may be used in a filename specification and may not be used to refer to directory or subdirectory names. This is one of the few places that the Apple /// Pascal System fails to provide a capability offered by UCSD Pascal (a product of Pecan Systems of Brooklyn, NY).

Wildcards can only be used in response to the following commands: Alter, Transfer, Remove, Change, List, and Ext-dir.

Conclusion

This pretty much covers the details of Filer operation. The next installment we'll create our first few small programs, introducing you to the Editor and Compiler in the process.

Page 17

Missing

Page 18

Missing

Native Mode Memory Organization—Part I:

Graphically Speaking

melvin a. astrahan, ph.d.

Introduction

Most personal computers (including our Apple ///'s) are capable of displaying data on their video screens either as characters of text, or in some form of graphic format. Character data is generally represented in memory in the familiar ASCII (American Standard Code for Information Interchange) code or occasionally in EBCDIC (Extended BCD Interchange Code). Most personal computer manufacturers, however, use highly individual schemes to store and display graphic data. It is unfortunate that no equivalent standards exist for image data (even among machines of the same manufacturer!)

A feature most machines do have in common is the raster-scan display (similar to a conventional TV set) in which the image is produced by several hundred horizontal scan lines. Each scan line, or raster, begins at a different vertical coordinate on the left side of the screen and is made up of many picture elements, or "pixels". The pixel data for each scan line is (generally) stored in a contiguous array of memory addresses. Thus, the data in the video memory represents a rectangular array of pixels, and each pixel is (usually) displayed at a unique horizontal and vertical (rectangular) coordinate on the video screen.

New Modes for Old

The Apple /// is capable of displaying graphic data in several different ways. These include four native /// modes, a /// plus interlace mode, and several Apple][modes in emulation. In this series of articles, I will discuss the native modes which include two monochrome (black and white) and two 16-color displays.

The monochrome modes are referred to as mode 0 (an Apple][hi-res equivalent of 280 by 192 pixels stored in 8K of memory) and mode 2 (an Apple //e double-hi-res equivalent of 560 by 192 pixels requiring 16K of memory). The 16-color modes are referred to as mode 1 (280 by 192 pixels, 16K) and mode 3 (140 by 192 pixels, 16K). It is not coincidence that these modes all have one dimension consisting of 192 pixels. That is the number of raster scan-lines displayed by the video hardware of the Apple][and /// family of computers, and is thus the vertical resolution of all graphics modes. The data in the contiguous memory array corresponding to each scan line, however, may be interpreted as a varying number of pixels depending on the graphics mode in use at the time.

Bank Business

The graphics memory of the Apple /// occupies all 32K of bank 0, and is accessible to programs when the XBYTE is set to $8F. Since bank 0 may only be accessed via enhanced indirect addressing, the bank, and thus the graphics memory, is mapped into addresses $2000 through $9FFF. Bank 0 is further divided into two 16K segments which are referred to as display buffers 1 and 2. Display buffer 1 occupies memory addresses $2000 through $5FFF. Buffer 2 occupies the remaining addresses $6000 through $9FFF. Each 16K display buffer is further subdivided into two 8K segments which I will call sub-buffers 1a ($2000-$3FFF), 1b ($4000-$5FFF), 2a ($6000-$7FFF) and 2b ($8000-$9FFF).

When you select a graphics display buffer (using a command such as the Business Basic PERFORM GRAFIXMODE (%MODE,%BUFFER)) you are in fact selecting the memory segment which the video-graphics hardware will display on the screen. When MODE = 1,2 or 3 (the 16K modes), selecting BUFFER = 1 connects the video display hardware to the 16K display buffer 1 ($2000-$5FFF). Selecting BUFFER = 2 displays the 16K buffer 2 ($6000-$9FFF). When MODE = 0 however, only 8K of memory is displayed by the hardware. In this case, selecting BUFFER = 1 displays the 8K sub-buffer 1a ($2000-$3FFF) rather than the entire 16K display buffer 1. Selecting BUFFER = 2 displays the 8K sub-buffer 1b ($4000-$5FFF) (*not* sub-buffer 2a as you might presume!!!) rather than the 16K display buffer 2. This is illustrated in a figure on page 276 of the Business Basic manual.

Try a Byte

It is only after you have determined which buffer is to be displayed and in which mode that the real fun begins. Earlier, I pointed out that the pixel memory mapping is a generally a rectangular array of contiguous memory. Well, that's not quite true in the Apple][and ///...

Consider an image displayed in graphics sub-buffer 1a, mode 0 (280 by 192 monochrome, 8K memory). This is the simplest display mode, and is identical in memory organization to the Apple][hi-res screen. In the fashion of the Apple][, pixels are *bit-mapped* such that a byte of memory represents seven (not eight!!!) horizontally contiguous screen pixels on the same scan-line. The eighth bit is used to control the color palette in the Apple][color modes and the display mode in some newer Apple //e mixed-modes where each byte can have a different display mode (monochrome or color).

Getting back to the ///, if a bit is 1, then the pixel is *on* or white. If a bit is 0, the pixel is *off*, or black. The lowest order bit of each byte maps to the leftmost pixel of that group of seven pixels. Higher order bits map progressively to the right within the group. Within a scan-line, the byte with the lowest memory address represents the seven leftmost pixels of the screen, or pixels 0 - 6. The next higher memory address maps to pixels 7 - 13 and so on... Thus, to display a scan-line of 280 pixels, 40 ($28) contiguous memory locations are required. To display 192 scan lines, 40 times 192, or 7680 bytes are required. Now I've told you that a mode 0 image requires 8K, or 8192 bytes of memory. What happened to the other 512 bytes?

The Missing 512

The sequential memory locations of the 8K mode 0 buffer ($2000-$3FFF) starting with $2000, do *not* map row by row down the screen with the first 40 bytes being scan-line 0 (scan-line 0 is the top scan-line on the screen, scan-line 191 is the bottom of the screen), the next 40 bytes scan-line 1, and so on... as you might expect. Instead, the second 40 bytes map to scan-line 64, one-third of the way down the screen, and the third 40 bytes appear 64 lines below that near the bottom of the screen. To make matters worse, the fourth group of 40 bytes appears on line 8!!!

The reason for this rather bizarre memory mapping scheme lies in some cost-cutting hardware shortcuts in the design of the earliest Apple computers. In order to maintain software compatibility, this video design has been propagated ever since, much to the aggravation of just about everybody.

The basic pattern is as follows. The screen is divided into 3 segments. Within each third of the screen, the screen memory is broken up into eight subgroups of eight scan-lines.

Within each subgroup, subsequent scan-lines are offset in memory by $400 bytes. The beginning of each subgroup is offset by $80 bytes. Each third of the screen is offset by $28 bytes as shown in Table I:

Observe that memory is used contiguously from offset $0000 through $0078 (lines 0, 64, and 128), but between $0078 (the end of line 128) and $0080 (the beginning of line 8) there is an unused gap of eight bytes. For each subgroup then, there are eight such "screen-holes", and there are eight subgroups for a total of 512 unused bytes.

Look Me Up Some Time

Now, lets say we want to plot a point at X,Y coordinate 7,8 on the screen. To do this, we must determine three items. (1) the byte relative to the beginning of a scan line which contains pixel 7, (2) the bit within that byte which corresponds to pixel 7, and (3) we must determine the base memory address of scan-line 8 so that we can compute the final memory address to modify.

There are several schemes floating around for calculating the base address of a scan-line. These may be found in various Apple][type publications. Quite frankly, however, routines which calculate the base address from the Y coordinate are time consuming, and even more so when one is limited only to enhanced indirect addressing. Routines which calculate the byte and bit within the scan-line are also available, but require division by 7 which is also quite time consuming. Since memory is generally plentiful on the ///, my favored approach is to trade memory for speed and do everything via look-up tables. I'll discuss X and Y coordinate look-up tables and high speed assembly language graphics routines in an upcoming article. We'll take a look at the strange world of the Apple ///'s modes 1, 2 and 3 graphics memory organization next time.

Table I

Scan-line		Offset from Base Address	Buffer 1a Address	Buffer 1b Address	Buffer 2 Address
SCREEN SEG 1					
S	0	$0000	$2000	$4000	$6000
U	1	$0400	$2400	$4400	$6400
B	2	$0800	$2800	$4800	$6800
G 1	3	$0C00	$2C00	$4C00	$6C00
R	4	$1000	$3000	$5000	$7000
O	5	$1400	$3400	$5400	$7400
U	6	$1800	$3800	$5800	$7800
P	7	$1C00	$3C00	$5C00	$7C00
S	8	$0080	$2080	$4080	$6080
U	9	$0480	$2480	$4480	$6480
B	10	$0880	$2880	$4880	$6880
G 2	11	$0C80	$2C80	$4C80	$6C80
R	12	$1080	$3080	$5080	$7080
O	13	$1480	$3480	$5480	$7480
U	14	$1880	$3880	$5880	$7880
P	15	$1C80	$3C80	$5C80	$7C80
		to 8 subgroups ...			
SCREEN SEG 2					
S	64	$0028			
U	65	$0428			
B	66	$0828			
G 1	67	$0C28			
R	68	$1028			
O	69	$1428			
U	70	$1828			
P	71	$1C28			
		to 8 subgroups ...			
SCREEN SEG 3					
S	128	$0050			
U	129	$0450			
B	130	$0850			
G 1	131	$0C50			
R	132	$1050			
O	133	$1450			
U	134	$1850			
P	135	$1C50			
		to 8 subgroups ...			

Game:

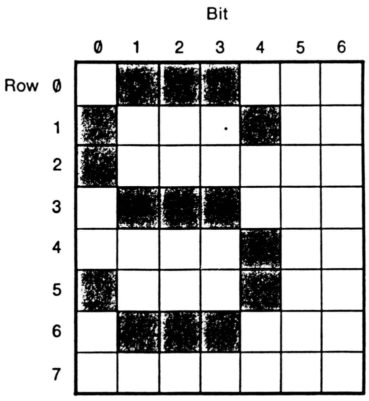
Space Convoy

ron puckett

```
10    REM ******************************
11    REM *   Space Convoy     by       *
12    REM *   ============ Ron Puckett *
13    REM * Copyright (c) 1986 ONTHREE *
14    REM *                            *
15    REM *   Your mission is to lead  *
16    REM *  a convoy of ships through *
17    REM *  an asteroid belt around   *
18    REM *  to plant Mobilia. You     *
19    REM *  have just enough power    *
20    REM *  for ten shots. But be     *
21    REM *  careful. Fast fireing may *
22    REM *  blow up the ship. Good    *
23    REM *  Luck!                     *
24    REM *                            *
25    REM *  The keys are:             *
26    REM *    <--   moves left        *
27    REM *    -->   moves right       *
28    REM *    [space bar] fires       *
29    REM *    [control-Q] quits       *
30    REM *    [escape]    pauses      *
31    REM *                            *
32    REM *    /BASIC/REQUEST.INV must *
33    REM *    be on line (Line 500)   *
40    REM ******************************
50    :
100    TEXT:HOME
105    REM ship$, rock$, stone$ are made so the
       y will output fast in color
110    ship$=CHR$(19)+CHR$(11)+CHR$(18)+CHR$(12
       8)+CHR$(17)
120    rock$=CHR$(19)+CHR$(12)+CHR$(129)+CHR$(1
       1)
130    stone$=CHR$(19)+CHR$(7)+CHR$(130)
140    PRINT CHR$(16);CHR$(1);CHR$(20);CHR$(13)
       :HOME
150    PRINT CHR$(20);CHR$(0):WINDOW 2,1 TO 39,
       24:HOME
160    name$=".CONSOLE"
170    ON ERR GOSUB 500
175    REM Download three new characters 0,1,2
176    REM  which are printed using chr () 128,
       129,130
180    GOSUB 1000:PERFORM control(%17,@clist$)n
       ame$
190    GOSUB 1000:PERFORM control(%17,@clist$)n
       ame$
200    GOSUB 1000:PERFORM control(%17,@clist$)n
       ame$
205    OFF ERR
210    ON KBD GOTO 250
220    gotbyte%=32:score=0:pos=20:shots=10
230    IF GOTBYTE%<>32 THEN 400:ELSE VPOS=1:HPO
       S=RND(8)*38+1:PRINT ROCK$;:HPOS=RND(8)*3
       8+1:PRINT STONE$;:VPOS=20:HPOS=pos:PRINT
        ship$;:score=score+1:GOSUB 320:GOTO 230
240    REM CHECK FOR PROPER INPUT
250    IF KBD=8 THEN POS=POS-1:S=1:IF POS<1 THE
       N POS=1
260    IF KBD=21 THEN POS=POS+1:S=1:IF POS>38 T
       HEN POS=38
270    IF KBD=17 THEN TEXT:HOME:END
280    IF KBD=32 THEN GOSUB 340
285    IF KBD=27 THEN GET Z$
290    ON KBD GOTO 250
300    RETURN
310    REM CHECK FOR COLLISION
315    REM Read text screen, if gotbyte$ = 32,
       all ok
320    VPOS=19:HPOS=POS:PERFORM STATUS(%17,@GOT
       BYTE$)NAME$:GOTBYTE%=ASC(GOTBYTE$)
330    IF SCORE=5000 THEN 600:ELSE RETURN
340    REM FIRE LAZER
350    SHOTS=SHOTS-1:IF SHOTS<0 THEN SHOTS=0:RE
       TURN
360    WINDOW POS+1,1 TO POS+1,19:PRINT CHR$(20
       );CHR$(2);:HOME
370    PRINT CHR$(20);CHR$(0);:HOME:WINDOW 2,1
       TO 39,24
380    RETURN
400    REM COLLISION
410    PRINT CHR$(7);CHR$(19);CHR$(15):VPOS=12:
       HPOS=1:NORMAL
420    PRINT" >>>>>>>>>>>>> COLLISION <<<<<<<<<
       <<<< "
430    VPOS=16:HPOS=16:PRINT"SCORE= ";SCORE;" "
440    FOR X=1 TO 1200:NEXT:PRINT CHR$(7):HOME:
       GOTO 210
450    REM INVOKE REQUEST.INV IF NOT THERE
500    OFF ERR:INVOKE"/P/LANGUAGES/BASIC/REQUES
       T.INV":RESTORE:RETURN
600    HOME:HPOS=1:VPOS=20:PRINT"        The en
       d of the Asteroids":REM 8 spaces
610    GOTO 440
990    REM DATA FOR SHAPES
1000    CLIST$=CHR$(1)
1010    FOR X=1 TO 9:READ A:CLIST$=CLIST$+CHR$(
        A):NEXT:RETURN
1020    DATA 0,119,119,119,99,99,65,227,247
1030    DATA 1,30,18,97,65,33,50,14,0
1040    DATA 2,30,30,127,127,63,62,14,0
2000    FOR X=32 TO 168:PRINT CHR$(X);:NEXT
3000    INVERSE:PRINT CHR$(128):NORMAL
```

Space Convoy is a classic example of how programs like our *Draw ON ///* can let you create your own character sets. Despite their graphic-like appearance, all images are text characters, stored as a 7 by 8 matrix.

Note in the accompanying figure no column for bit 7 is shown, since it is not a part of the actual character. The purpose of the seventh bit is to blink the bits in its associated row. If you press [escape] to pause the program and examine your convoy, you will see just one portion of each spacecraft is blinking.



Format of a Character Cell

***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.

***ON THREE* presents...**

The Unprotect 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 Unprotect 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 *Unprotect Driver* comes with full documentation and will work with *Selector ///* so you will no longer require a "key" diskette. The *Unprotect Driver* is sold for legitimate ***Archival*** purposes only. *ON THREE* ***does not condone*** and will not condone duplicating a disk for any other purpose.

...from **ON THREE**

New!

Keep all of your **ON THREE** *magazines in one place in our handy binder.*

- **Holds 12 issues**
- **Always Available**
- **Practical**
- **Attractive**
- **Convenient**

Blue print with gold trim on sturdy, re-inforced white vinyl.
Only ***$14.95***
plus $1
shipping/ handling

A great gift idea for
your favorite Apple ///'er!

Order now on our new toll-free line
(800) 443-8877 (except California)

Reading the Mail:

Three Questions

Mailbox Undamaged After Barrage

Dear Val:

Like most of your readers, I greeted the latest *ON THREE* when it hit my mailbox with a mixture of amazement, pleasure, and a certain ruefulness. You've got a lot of disillusioned Apple /// lovers out here, Val. First Apple discontinued our machine (it is only a "machine" as my wife tirelessly points out), and then *ON THREE* seemed unilaterally to have canceled our subscriptions.

Have I lost faith in my Apple ///'s, both of them? Good question. Probably I haven't, at least not until they go wrong. And then what? Apple itself no longer "supports" the ///. (Could you tell me exactly what that means—No more products? No more spare parts? No more repairs?) *ON THREE* is offering us some enticing products, but... I'm going to be brutal—*ON THREE* has something of a credibility gap. Let me explain.

Last night I wrote a list of the things you offer that I'd like to have for my ///; it comes to over $4000. Then the doubts begin: Will these products really work? If they don't, will *ON THREE* come through for me? And when my "machines" one day stop working? That's the big one. Maybe I'd better put my $4000 in the bank and wait for the demise of my ///'s. In (let's say) two years time, who knows what may be on the market? Probably no one else but me is contemplating infidelity to the /// in this cold-blooded way. But one thing is for sure: if I am unfaithful, after what Cupertino did to the ///, nothing on earth can persuade me to bed down with another Apple. (Having written all that, I can almost guarantee you won't publish this letter.)

As I said, Val, you've probably got a lot of disillusionment out here to cope with. BUT—it *was* a pleasure to get the new *ON THREE*. I sat down and read it from cover to cover. $4 worth of pure joy. Now all you have to do is stick to the promised schedule and maybe some of that disillusionment will vanish.

Dr. Jim Pinnells
Heidelberg, W. Germany

You make a number of provocative points. As far as ON THREE *is concerned, there is a gap, and it was created (as we have written previously) during a period where one individual was attempting to cover two or three full time jobs. It was, in a larger sense, brought on by Apple's decision to abandon the ///, which in turn caused* ON THREE *to jump in and turn out new products. It was during this period (when* ON THREE *was not being published regularly) that we developed two of our most outstanding products:* Draw ON /// *and the* 512K Memory Upgrade. *Loves of labor almost always have the effect of missed deadlines and sidelined projects. Since we are currently typesetting our fifth issue of 1986, allow us to present you with a complimentary jar of vanishing cream to narrow the gap, along with a flow of new and innovative* ON THREE *products. And yes, these and those in planning stages will be supported fully. Will the products work? Did you really need to ask?*

The Apple /// is not going to roll over and play dead, merely because Apple itself has dropped support. Sun Data has acquired virtually all of Apple's spare parts inventory and there are a couple of dealers around the country who proclaim to specialize in Apple /// service. Above and beyond that some dealers will service the machines when they can obtain spare parts. And a new and revitalized ON THREE, *by publishing service tips and other information previously available only to developers, will further expand our knowledge and extend the life of our machines. New products are being developed not only by* ON THREE, *but others as well.*

Our romance with Apple—although somewhat damaged in the handling—is by no means over, and we hope you feel the same. As we walk through life, certain unforseen events inexorably occur and if our viewpoint is a realistic one we learn to adjust, to take chances. Rather than to start a new courtship, we prefer always to "give a second chance." We think the new management at Apple means business and they deserve to be heard "just one more time." [Hear, Hear! ...Bob]

The letters we select for publication are based almost entirely on reader appeal and interest. Neither brickbats nor bouquets count around here. Thanks for writing, and keep reading.

Draw ON Update

Dear Bob:

Glad to have gotten the report, in the form of your latest publication, that you are still with the /// and doing well. I have a couple of things I sure wish you would help me with. One is the *Draw ON ///* I purchased. I am using a UPIC connected to an Epson MX-80 F/T with Graftrax. I have never bought the Pkaso card because you informed me that you would probably have an update which would make it unnecessary. In reading *the magazine* I believe this has been done. Please advise what you think is the best route to take. I still would like to have a color printer to take advantage of this feature in *Draw ON*.

Next thing is a problem with /// E-Z Pieces. When I start up the underscore, using either method, and set it to stop after a word or book title, it will not stop but instead underlines everything to the end of the paper. I use word processing a great deal and this is the only thing that seems not to be perfect.

Charles B. McClain
Houston, TX

You read correctly. There is now a Draw ON *update, available for $15, which allows the MX-80 and the UPIC card to work correctly. To get your update, send in your* Draw ON *disks, state the version you need, and include your serial number, name and address, and a check for $15. We will do your update and return the disks promptly.*

On /// E-Z Pieces, we would make a guess that the printer control code to turn underlining off is incorrect. Check your printer manual for the correct code and then see how it appears in /// E-Z Pieces.

Apple /// BBS

Dear Sir:

I recently looked through the February issue of *ON THREE* and noticed a section for Apple /// user groups. I don't have a users' group but I do run a BBS which has a subboard exclusively for Apple /// users. It's name is "Capitol Hill BBS" and is located in New Jersey. The phone number is (201) 447-2897 and the hours of operation are as follows: Mon-Fri—6-10 pm EST, Sat-Sun—11am-midnight.

Kevin Scott
Sysop, Capitol Hill
Ridgewood, NJ

Thank you, Kevin. After this appears in print, you'll certainly find some new users. We will be publishing a list of BBS's in this or the following issue. Our own ON THREE *BBS is at (805) 644-1055, 24 hours.*

Dutch Treat

Dear Sirs:

From advertisements and friends I have heard about your company. I would like to receive information about the products your are selling and the monthly magazine. If several people order a product from you, do you offer quantity discounts?

I have also heard about the driver for the UniDisk ///.5. I think it's great you have written the software, but what I long for is to be able to start the Apple /// not from the internal drive, but from the UniDisk. Also, I wonder if there is a possibility to use the two joystick ports to connect other hardware to.

Is there someone in the States who sells upgrades for the Apple /// to an Apple /// plus? I read in *ON THREE* that you can do the upgrade yourself and that the /// plus has better quality on the monitor. I am curious about a 16-bit microprocessor (65802) for the Apple ///.

It's a pity for all the Apple /// users in Western Europe that they get nearly no information about products for the ///. I hope you will go on supporting the Apple /// for a long time. I will use the computer as long as there are new products to buy which enhance its possibilities.

J. Woretshofer
Maastricht, Netherlands

Thanks for writing for information about ON THREE *products. Our catalog has been sent to you. We do offer "group purchase discounts" for two or more of the same product, purchased at the same time and shipped to the same address. Please write or call for specifics.*

The UniDisk ///.5 can not currently be used to boot from. To do so would require major modifications to SOS as well as internal hardware changes. The only significant differences between an Apple /// and an Apple /// plus is the //e-like keyboard and improved video interlace. These kits may still be available from Sun Data, P.O. Box 4059, Logan, UT 84321. They are easy to install.

We are considering selling the new 16-bit 65C802 microprocessor. You will read about it in ON THREE *when we do. We have used them in Apple ///'s (and an Apple //e) without significant problems. The joystick ports may be used for a number of two-way serial applications. We hope this information is of help to you and look forward to hearing from you again.*

Time Out

Dear Val:

Shame on you! You ran an article in the April issue (Taming Timing) about Device Dependent Error 34 and didn't even mention that this error indicates the drive speed is too fast. Error 33 means too slow. Be careful or you will get relegated to page 35.

Bob Consorti
Ventura, CA

To paraphrase Ann Landers (or is it Abby), 20 lashes with a crashed diskette for us.

Trustor Trick

Dear Bob:

I am delighted to have just received the February issue of *ON THREE*. It's great to see the new products you have for the ///. I wish to especially thank Janet Schanz for her extra effort in getting the driver software sent to me. Although I didn't purchase it from *ON THREE*, I will recommend it [the Xebec 9730] to anyone who needs more disk space. Within two hours of the postman delivering the driver, I had formatted and transferred two ProFiles-worth of data to the drive. For anyone else transfering data from a ProFile to a Trustor [Xebec], it is much faster to install the Trustor in slot 4, move the ProFile to slot 3, install the driver files on a copy of the system utilities and then use the copy files function to directly transfer the data between the drives. Remove the ProFile from the ///, remove the ProFile driver from your boot disk and you are ready for a wonderful experience, more empty blocks than you can imagine. The ProFile works great on a //e if you get the //e controller card.

Earl T. Brelje
So. St. Paul, MN

Bob says thanks for the kudos. Our first real working experience with the Sider or any hard disk has been over the last few months. It is truly a thrill to type "S" (for save) from AppleWriter and see your cursor come back in less than the time it takes to blink your eyes. We're sure your hint will prove useful to many.

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

USER GROUP: The Apple THREE Group International, formerly Apple /// Owners & Users Group Int'l, is an independent, non-profit organization for all Apple ///'ers (if you belong to a *local* users group or are connected to one via a modem, great, if not, we'll try to make you feel like you belong). Started in 1983 in Naples, Italy, we publish a monthly newsletter "Apple /// News & Views," containing /// news gleaned from every source possible, attempt to answer or obtain answers to member's questions, and are building a "library" of *every* piece of Apple /// public domain software available. *Cost?* Annual dues are $5 per calendar year in the US ($6 in Canada, $20 foreign). Software (members only) $3 per disk, postpaid (Canada and overseas additional). Interested? Write for an application.

Apple THREE Group International
c/o H. Joseph Dobrowolski
P.O. Box 913
Langley AFB, VA 23665

FOR SALE: Complete 256K Apple /// system, four drives, clock, Z-80 softcard, amber crt with spare, gameport, Grappler printer driver, Wordstar, dBase II, Flight Simulator, and more. Less than 70 hours. $1900 or best offer. Mark (213) 698-9848.

Page 25
Missing

Page 26

Missing

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

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

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

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

Los Angeles-South Bay
Apple /// Users Group
P.O. Box 432, Redondo Beach, CA 90277
(213) 316-7738

Apple /// Users of Northern California
220 Redwood Highway #184
Mill Valley, CA 94941

International Apple Core Apple /// S.I.G.
908 George Street, Santa Clara, CA 95054
(408) 727-7652

Canada
Apples British Columbia Computer Society
Apple /// S.I.G.
P.O. Box 80569, Burnaby
BC Canada V5H3X9

Canadian Apple /// Users Group
80 Antibes Dr. Suite 2805
Willowdale, Ontario, Canada M25R 3N5
(416) 665-3622

Colorado
Colorado Apple Three User Group
P.O. Box 3155, Englewood, CO 80112

Connecticut
Apple /// Society of
Southern Connecticut
34 Burr School Rd.
Westport, CT 06880
(203) 226-4198

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

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

Illinois
Third Apple Users c/o Lavona Rann
1113 Wheaton Oaks Dr., Wheaton IL 60187

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

Maine
So. Maine Apple Users Group
Casco St., Freeport ME 04033
(207) 865-4761, X 2249

Maryland
Apple /// SIG Chairman
Washington Apple Pi
8227 Woodmont Av. #201
Bethesda, MD 20814 (301) 654-8060

Minnesota
Minnesota Apple Corp Users Group
P.O. Box 796, Hopkins, MN 55343

New Jersey
North Jersey Apple /// Users Group
c/o Roger T. Richardson
P.O. Box 251, Allamuchy, NJ 07820
(201) 852-7710

North Carolina
North Carolina Apple /// User Group
2609 North Duke St. #103
Durham, NC 27704

Ohio
Cincinnati Apple /// User Group
5242 Horizonvue Drive,
Cincinnati, OH 45239
(513) 542-7146

Apple Dayton - Apple /// S.I.G.
P.O. Box 1666, Fairborn, OH 45324-7666
(513) 879-5895

Oregon
Portland Apple ./// Users Group
Portland OR
(503) 225-1623

Overseas
Apple THREE Group International
c/o Maj. H. Joseph Dobrowlski
P.O. Box 913
Langley AFB, VA 23665

Apple User Group Europe e.V.
Box 11 01 69 D-4200
Oberhausen 11
West Germany
0049-6195-7
3917

British Apple Systems User Group (BASUG)
Apple /// S.I.G.
P.O. Box 174, Watford Herts
England WD2 6NF
0727 73390/72728

Le Club Apple
43 Avenue de la Grande-Armee
75116 Paris, France

Texas
Apple Corps of Dallas Apple /// SIG
P.O. Box 5537, Richardson, TX 75080

River City Apple Corps /// S.I.G.
Box 13349, Austin, TX 78711
(512) 454-9962

Houston Area Apple Users
Group (Apple /// Division)
P.O. Box 610150, Houston, TX 77063
(713) 480-5690 or 974-5153

Virginia
Charlottsville Apple /// User Group
216 Turkey Ridge Road
Charlottsville, VA 22901
(804) 642-5655

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

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 then 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 inadvertantly 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 Lang.	AL	Modems	MD
		Modula-2	MU
Business Basic	BB	Pascal	PA
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 Pieces	EP

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

Save more on 512K Upgrades!

Now you can save even more when you purchase the *ON THREE 512K Upgrade*. If you've read our ads, you know the final cost is $399 plus shipping, etc., but you remit $449 plus at the time the order is placed and $50 is rebated when we receive your old board back.

Effective immediately, we are offering our upgrade customers a new money saving option. As before, you may choose to receive a $50 cash rebate *or* you may now elect to receive a credit voucher from *ON THREE*, worth $60 on any future *ON THREE* hardware or software product purchases! This will effectively make the cost of your upgrade just $389, saving an additional $10, making our upgrade even more attractive to you.

The choice for additional savings is yours. With each *512K Upgrade* we ship, we will enclose a form for you to complete and return with your old board. If you want to save $10, just check the $60 credit box or, if you have a need for ready cash, mark the $50 cash box.

We hope our new policy will be of benefit to you.

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 '\' 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 excercises 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.

Page 29

Missing

Page 30

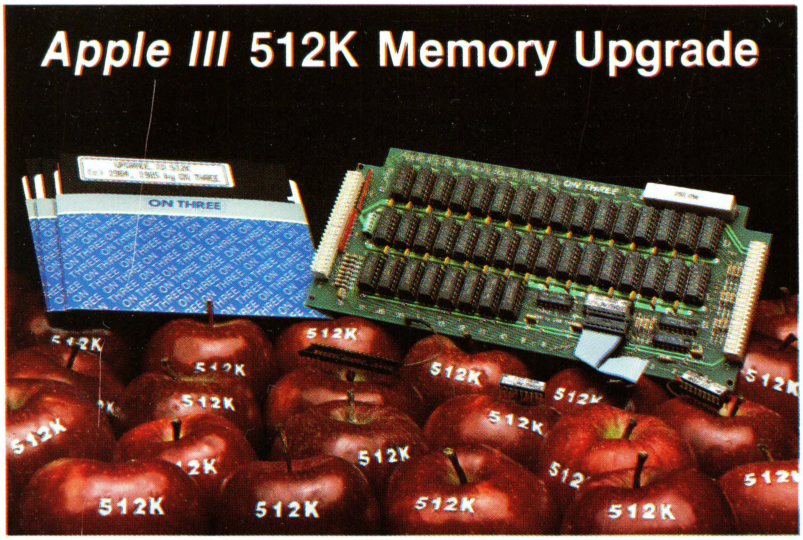
Missing

Page 31

Missing

Page 32

Missing



+

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

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

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

Look forward in 1986 to more file capacity for your applications programs like VisiCalc (regular and advanced versions), /// E-Z Pieces, Selector ///, 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 ///* 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 ///* 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 /// or Word Juggler.

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

* 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 /// and many other programs will automatically have about 450K of memory to work with.

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 /// 512K memory board with *state-of-the-art* 256K memory chips.**

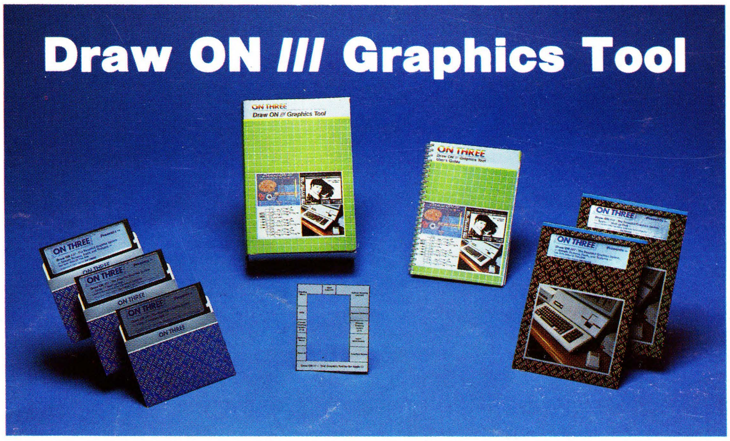
ON THREE (805) 644-3514
P.O. Box 3825, Ventura, CA 93006

Calif. residents add 6 % sales tax (products only) We accept Visa, Mastercard, American Express†
†*3% surcharge on American Express orders*

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

ON THREE Presents . . .

Draw ON /// Graphics Tool



- ■ Use *Draw* ON /// directly with Apple //e mouse and interface, joystick, keyboard, or Apple Graphics Tablet (Graphics Tablet version $50 additional)
- ■ *Draw* ON /// can spruce up dull graphs with its many typefaces or by creating fancy borders and textured images
- ■ *Draw* ON /// comes complete with easy to follow menus, a durable spiral-bound instruction manual and tutorial, keypad overlay, and unprotected diskettes which will install on *Selector* /// or Catalyst
- ■ *Draw* ON /// is compatible with all monochrome monitors as well as NTSC (standard) and RGB (hi-res) color monitors
- • Multiple help screens
- • User-adjustable grids
- • Zoom in for detailed work
- • Rubber-banding of lines

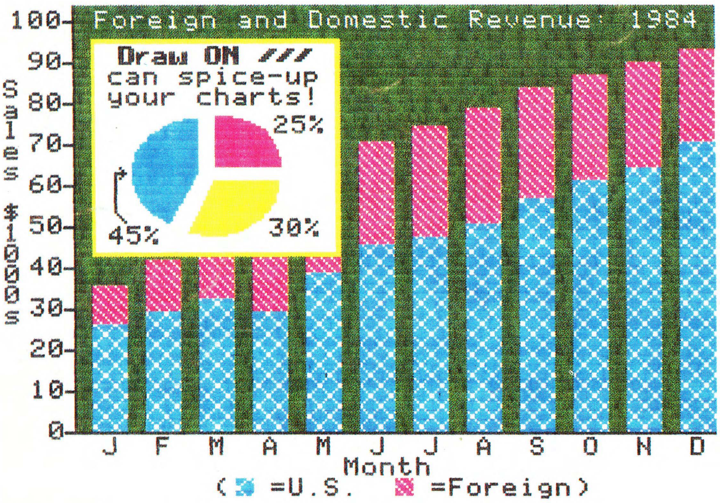
The most versatile Apple /// graphics tool ever designed!

$179 . . .plus $5 shipping and handling

What? A computer graphics program that is powerful and easy to use, has the resources of a complete graphics art studio, creates professional-quality charts and diagrams, complex illustrations and original artwork, letterheads, slides and tables for presentation? Don't you believe it! . . . **unless you're talking about Draw ON ///™**, from **ON THREE!**

Draw ON /// transforms your Apple /// into a drafting table, easel and sketch pad, all rolled into one, like MacPaint with color. Computer Aided Design (CAD) applications such as circuit layouts and flowcharts are childs play for **Draw ON ///.**

Draw ON /// comes with a wide selection of text fonts and objects which can be supplemented with those of your own design. Mix and match with drawings and charts, using **Draw ON** ///'s powerful cut and paste facility. You can use **Draw ON** ///'s many fonts to label your own drawings as well as those in other applications, and you can pick up objects, expand, shrink, rotate, invert, and texture.



Draw ON /// requires 256K minimum memory

Look!

You can print Draw ON ///screens with all of these popular printers:

- Apple DMP
- Epson FX, MX, RX series
- ImageWriter
- ProWriter

plus, with a PKASO/PKASO-U interface:

- Centronics
- IDS Prism, Color Prism*
- NEC
- Okidata
- . . .and others

*required to print color drawings

Specify printer and interface when ordering