ON THREE™

The Magazine For Apple III Owners and Users

Volume 3 - Number 3

March 1986
$4.00

- **Directories ///**
- **Review ON: Keystroke**
- **Integrated Software For Free!**

- **Questions and answers**
- **and more!**

New High Capacity Drives for the Apple /// Available now from ON THREE

34 MB **10-20-MB** **800K**

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

"before"

"after"

"I've done away with my desk altogether...
All I use now is an Apple /// work station
and **ON THREE's ONTIME Desktop Manager**"
...W. Shakespeare, Stratford-on-Avon

ONTIME Desktop Manager

A background utility to let you cut and paste from or to databases, spreadsheets, etc.

You control the computer, it doesn't control you!

- *The ONTIME Desktop Manager is instantly available from any word processor, spreadsheet, etc. only a keystroke away*
- *Best of all, you can cut and paste from the notepad or the calculator to your current application*
- *Includes:*
  - ✔ *Appointment Calendar*
  - ✔ *Notepad with editor*
  - ✔ *Full function calculator*
  - ✔ *Mouse compatibility*
- *Optional utilities coming soon*

ONTIME Desktop Manager $99 plus $6 shipping
Introductory price thru 3/31/86
available about 2/1/86

- You remember the name, it remembers the time

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

Send me _____ ONTIME Desktop Managers

Name _____

Address _____

City _____

State _____ Zip _____ Phone _____

☐ enclosed $ _____

☐ M/C, Visa, AE*

number _____ exp. date _____

Signature _____
*3% surcharge on American Express

Note: ONTIME Desktop Manager requires either ONTIME or Apple Clock.

ON THREE T.M.

The Magazine For Apple /// Owners and Users

Publisher:
Bob Consorti
Editor:
Val J. Golding

Comptroller:
Joseph Consorti
Logistics:
Janet Schanz
Interior Art:
Virginia Carol
Technical Support:
Rob Turner
Order Dept.:

(805) 644-3514

Typography:
West Coast Typography
Van Nuys, CA
Printing:
The Express
Ventura, CA

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

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

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

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

Subscription information:
U.S.: $40 for 12 issues.
For First Class service, remit an extra $12
Foreign subscription information:
Canada, Mexico, APO, FPO: $20 *additional*, $60 total.
So. America, Europe: $24 *additional*, $64 total.
Pac. Islands, Asia, Australia, Mid-east: $29 *additional*, $69 total.

Back Issues:
Available at $5 each. Please indicate issues desired on the enclosed order form.

Postage for *ON THREE* products:
U.S.: listed on enclosed product price sheet.
Canada and Mexico: double listed US prices.
Outside North America: triple listed US prices.

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

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

Volume 3, No. 3 **Table of Contents** March, 1986

FEATURES

DEPARTMENTS

ON: The Cover

Our March cover displays the best of all worlds, the many high-capacity disk drives now available for the Apple /// available from ON THREE.

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

Products & Services Available for the Apple ///

Current ON THREE Price List

Product	Price	S/H
12-issue subscription to ON THREE	$40.00	
Back issues of ON THREE	$5.00 ea	
Disk Of the Month (DOM)	$14.95 ea	$2.00
Order 2 or more for only	$12.50 ea	$4.00
ONTIME-ON THREE O'Clock Combination	$79.95	$4.00
ONTIME	$39.95	$2.00
ON THREE O'Clock	$49.95	$3.00
Lazarus /// Version 2.0	$49.95	$2.00
Draw ON ///	$179.00	$5.00
Apple /// Fortran	$99.00	$5.00
Selector /// Program Switching Utility	$99.00	$7.00
CROSSWORD-SCRAMBLER	$19.95	$2.00
The Apple /// Fruit Machine (Slot Machine Game)	$19.95	$2.00
The Apple /// Card Machine (Blackjack Game)	$24.95	$2.00
Fruit-Card Game Combo	$39.95	$3.00
Game Card ///	$59.95	$2.00
Micro-Sci Disk Drives		
A3-140K	$299.00	$6.50
A143-560K	$399.00	$6.50
3M DS, DD Disks (10) for A143	$39.95	$3.00
Apple /// Unidisk 3.5 (800K disk) with interface card, Apple /// software & documentation	$499.00	$10.00
Unidisk doc & disk only	$50.00	$3.00
Sider 10 (10 MegaBytes)	$999.00	35.00
Sider 20 (20 MegaBytes)	$1200.00	$35.00
(Siders can be daisy-chained		
Xebec 9730 (34 MegaBytes)	$1999.00	$35.00
512K Memory Upgrade (256K to 512K) *(You remit $449 and we rebate $50 when the old 256K board is returned.)	† $399.00	$10.00
256K Memory Upgrade (128K to 256K)	$200.00	$10.00
256K or 512K upgrade installation by ON THREE (appt. only)	$50.00	
Apple //e Mouse and interface card (can be used with Draw ON /// or Desktop Manager)	$150.00	$5.00
Dust Cover for Apple /// with Monitor ///	$11.95	$2.00
Dust Cover for Apple /// with ProFile & Monitor ///	$12.95	$2.00
I ♥ MY APPLE /// T-Shirts Sm, Med, Lrg, X-Lrg (Color choices: Yellow, beige, white, blue)	$11.95	$3.00
I ♥ MY APPLE /// Sweat Shirts Sm, Med, Lrg, X-Lrg (Color choices: Silver, white, blue, yellow)	$18.95	$3.00
I ♥ MY APPLE /// Caps	$5.95	$2.00

†subject to increase

California residents add 6% sales tax.

To order, call (805) 644-3514 or write:
ON THREE, Inc.
Attn: Order Dept.
Post Office Box 3825
Ventura, CA 93006

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

ON THREE Presents . . .

• Disk of the Month

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

$14.95 each plus $2 s/h

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

- Volumes 7 & 8 now available

Vol 7. Contains IMAGEHELPER, a BASIC program to aid in designing graphics images, a heap sorting routine, and a program to preset printer codes.

Vol 8. Contains BASIC and Assembly programs to sort directories, a Pascal program to clean drive heads, and a text file reader.

ON THREE Presents . . .

LAZARUS ///

File Recovery System

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

$49.95 plus $2 s/h

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

ON THREE Presents . . .

I ♥ My Apple ///

Spring T-Shirt Bonanza!

- Bright t-shirt colors for sunny spring days
  - yellow
  - blue
  - white
  - beige

  $11.95 plus $3 s/h
- Maintain your bod like your ///
  I ♥ My /// Sweat Shirts
  - yellow
  - blue
  - white
  - silver

  $18.95 plus $3 s/h

  ☐ sm ☐ med ☐ lg ☐ x-lg
- Cap it off with I ♥ My /// Caps

  $5.95 plus $2 s/h

The Editor Dishes it Out:

Apple.Sauce

val j. golding

One, Two, Three, Forum

How strange it is, as **Bob Consorti** mentioned in *Block_Write*, that we are writing about spring on Christmas day. Yet that is the way an editor perceives time. In a way, almost like time travel, because February lies in the past and March is all but wrapped up. We have two issues behind us since we've been at the wheel, and we're pleased with the response. Yet *ON THREE* is to an extent in a state of limbo. Bob hired us to guide and produce the magazine, and we are without question going to do just that. But again, we have to appeal to our readers for some guidelines. We want to know the kind of material that turns you on the most.

The Apple /// is in a class by itself, one of the most powerful and exotic computer systems developed, yet (now) totally without support from its creators. That puts us—and a scant few others—in the role of promoting its use and future development. (In this area, we have some great plans we'll tell you about soon.) We have found something we believe to be truly unique and, to our eyes, rewarding. That is *user loyalty*. We have spent several years in the world of the Apple][, a fine machine in its own right, but never have we experienced such perseverance as that of Apple /// users who, notwithstanding withdrawal symptoms by Apple Computer, Inc., have decided that their machine has potential and a future.

ON THREE shares this philosophy. We have committed our time, dollars and brains to prolonging what could otherwise be an abortive lifespan for the machine we all love. *ON THREE* (the company) exists not only to make a profit (which obviously is a necessity), but to pour our resources into new products (the *ONTIME Desktop Manager* is only one such example), and to promulgate the dissemination of Apple /// information. When you purchase, or consider purchasing, an *ON THREE* product, keep in mind that not only do you receive hardware or software value for your hard-earned dollars, but you are also contributing to the future of the Apple ///. (Too bad we can't take an income tax deduction on that.)

Regrouping somewhat, we can't tell you how impressed we are at the loyalty and close-knit camaraderie we have observed since we have been here. This is exhibited in the *Three Questions* reader's forum which has been expanded for this issue. But we'd like to go further than that. In our opinion, there is a basic need to exchange information, questions and answers not necessarily responded to by our editorial staff. What we propose is a new column, a true *reader's forum*, one in which you can share with other readers all those little tidbits of information you have spent so much time discovering. The Apple ///, like any microcomputer, is not a perfect machine, although in our eyes it may come close at times. We suggest *Three Forum* as an outlet, where you can express your frustrations and delights, problems and hints, and sorrows and joys. So we ask, nay urge, for your letters addressed to *Three Forum*. How long have you had an unresolved problem? Someone, someplace out there has an answer. You better believe it!

Internal Medicine

The bent of this column is to acquaint you with the stories in each issue, but we're so excited about what we have coming up, we can't resist the temptation to tell you about that first. Those of you who have come up to the Apple /// via the Apple][route will probably recall the article, published first in *Apple Orchard* and later revised and reprinted in *All About Applesoft* entitled "Applesoft Internals". Some time ago, Apple Computer released—to registered developers only—a similar document relating to Business Basic. On our first reading, we were amazed to discover how close a cousin Business Basic is to Applesoft Basic. Not only do many of the subroutines function similarly (the major difference relating to the third byte, or bank register, of a standard two byte address), but many of the assembler label names are *identical*.

Since Apple has withdrawn all support for the ///, we have thrown caution to the winds, and next month you will be able to peruse (and use) a major article entitled *Business Basic Internals*, containing instructions and descriptions of all essential BASIC subroutines. And if that is not enough, we will be returning with the second installment of our *Now About That Basic Business...*

The Here and Now of it

Surely everyone recognizes the name **John Jeppson**. We have successfully prodded John into allowing us to print a previously unpublished *Softalk* article, *Directories Three*, an exploration into the intracacies of the Appie /// (and ProDOS) directory structure. Beside some really down-to-earth explanations, John has a program that will let you rearrange your existing directories. Take a look at page 5 and don't stop.

Our second major article is by **John Sollman**, a review of *Keystroke Data Base and Report Generator*. Let John's review serve as a guide to your database needs. His review needs no further description; he has covered all bases. Good and bad, *ON THREE* will give you *honest* product reviews. We believe our position is not only to turn you on to good products, but also to warn you of the dogs. (Keystroke is definitely not a dog).

Back with us again is **John Lomartire** (to whom we apologize for the misspelling in the last issue). John has a way with words, and we're sure you will enjoy reading his *Free Integrated Software*, wherein he has come up with a technique to use a spreadsheet to integrate textual material with tables, one more way in which you can increase the power of your *existing* programs.

Our boss, the editor (figure that one out, folks), tells us we have already exceeded our alloted space for this issue. What this means is a quick good-bye. See you around, gang...

Notes from the Publisher:

Block_Write

bob consorti

By now you may have heard that Titan has released their /// plus //e product. While we have not yet had a chance to review a copy ourselves, we know that it takes up two slots. As soon as we heard of this, we thought that people would have to soon make some serious decisions about their ///. Namely, do I keep it?

To me, the idea of turning my old Apple /// into a quasi-Apple //e is disheartening. I'd much rather have my ///, yet there is so much software out there for the Apple][family that it's a hard decision. With the Titan /// plus //e you can now have the best of both worlds, but. . .

Since it requires two slots and I already have all four filled, what must I sacrifice to get //e capability? Do I remove my printer card? How about my mouse card, or should I take out the hard disk interface? What seems like a simple decision gets harder by the minute. Many //e programs now use the mouse, so I don't want to get rid of the mouse card. I may want to use a new modem program on the //e to print-out some files from CompuServe, so I need my printer card. And if I can't use my expensive hard disk, it's of no use whatsoever.

And The Answer Is...

Have you been holding your breath, waiting for the words *We have a slot expansion box?* Well, hold on for a while because we don't. Not yet, that is. Sometimes you know that a product is going to sell, because you just know there is a need for it. Items like the 512K Memory Expansion (still priced at only $399 after rebate) have been needed for many years. I'm not sure that a slot expansion box will sell. What I'd like all of our readers to do is to convince me.

Write in to *ON THREE* and send me a letter, postcard or a singing telegram if you'd purchase a slot expansion box. I'd personally love to see it for the ///, but this could be an expensive project and I want to make certain that all of you (or a reasonable number) want it.

The Rise And Fall

Over the past few months memory chip prices have gone through the roof. After a full year of price decreases (due mainly to falling memory prices), our 512K's were hit towards the end of '85 when memory prices trebled. I won't (hardly) get into the reasons behind the memory price increases, but perhaps your local congressman or senator and the American memory chip producer's lobbyists can explain.

'Cry-baby' attitudes and pushing for protectionist bills has significantly increased our cost for 512K Memory Boards, yet *ON THREE* is going to try to hold the line. We will do our best to keep the price of the 512K Memory Board at only $399 after rebate. I believe memory chip prices will fall again, but it may take some time.

Ordering your 512K Memory Board now will assure you of getting the lowest price available. It will also enable us to keep our prices low. I'll take a few more moments and remind you that all Apple /// native mode software now works with 512K Apples. /// E-Z Pieces, the VisiCalc's, AppleWriter, Word Juggler and all the others now use the expanded memory on a 512K Apple ///. If you've ever run out of memory, it's time to upgrade.

Copying Notes (Again)

A few days ago I went to a small Christmas party (I'm writing this column at the end of December—Aren't magazine schedules funny?). A number of IBM'ers were present, the host included. The subject turned to software and a couple of the partygoers borrowed the host's computer to make copies of some software. I guess I turned an ashen shade, because they were soon asking what was wrong.

After trying to tell them that what they were doing is illegal, their reply was that the software was overpriced and that when it came down under $50 they would buy it. As a software producer I find it hard to understand how two very intelligent people can be so ignorant. I agree that it's hard to justify spending $495 for 1-2-3 or other products. But the software they were copying was under $100 from a company that prides itself on having an excellent cost-performance ratio.

Very little useful software (excluding games) can be written to sell for under $50. It's a fact of life in the software business. The entire concept of overhead went over their heads. For the one dollar cost of the diskette they were copying to, they missed a few things like the costs of disk production, shipping, product support, management and staffing for production, shipping and support, office, production and warehousing space, computer equipment, materials, etc.

All of these items cost money! If a 10% profit turns up in there, many software developers are very happy. Some people think that software houses are rolling in money—maybe some are, but most are just getting by, and if you have been reading or watching the news, you would know that other just plain didn't make it. The rationale for the white collar crime taking place doesn't hold up to scrutiny.

Illegally copying software robs the software producer, forcing higher prices so he can stay even. Then the thief says prices are too high, and so steals again. It amazes me that a person who thinks it's bad to shoplift expensive jewelry from a store doesn't blink an eye when it comes to stealing hundreds of dollars worth of software over cocktails.

I tend to think that once people realize that there is much more to a software product than the price of the disk it is distributed on, copying will slow down. You'll note that I said slow, not stop. I'm not sure if we can ever stop the thefts, but with education we can slow it down.

You may have noticed that I've been spending a lot of column space on the subject of software theft lately. I guess it's just because I'm becoming more acutely aware of just how it affects small businesses. I'm telling everyone because in the small niche that the Apple /// user is in, it may be getting easier and easier to rationalize that theft is all right and doesn't hurt anyone. I trust everyone realizes the absurdity of that intellect.

Page 5
Missing

Page 6

Missing

Page 7

Missing

Page 8

Missing

why Apple /// files do not require contiguous blocks on the disk. Whenever a file overflows any complete block it is assigned another; SOS simply grabs the next free block anywhere on the disk. The block number address of this new block is entered in the list maintained in the file's index block.

Note that data blocks of non-directory files contain only data. Data blocks never contain any linkage information directing SOS from the current block to its successor. All such block number pointers are maintained in the separate index block. It will be seen, therefore, that there cannot be any non-directory files that are exactly two blocks long. A one block file contains one data block. But the next larger file contains two data blocks and one index block, or three blocks in all.

Block number pointers are each two bytes long, so an index block has room for only 256 such pointers. If a sapling file grows so large that it contains more than 256 data blocks then it undergoes another metamorphosis into a "tree file". Tree files contain a master index block which can contain up to 128 pointers to ordinary index blocks, each of which, in turn, can contain pointers to as many as 256 separate data blocks. Thus a tree file can accommodate up to 128 * 256 = 32,768 blocks or 16,777,216 bytes of information, enough to fill a 16 MegaByte hard disk with a single file.

A Tree file can accommodate enough data to fill a 16 MB hard disk with one file.

Directories are quite different. Root directories, in the current version of SOS, always contain four blocks and always occupy blocks [2...5] on the disk. That is fixed. Subdirectories, on the other hand, may contain one, two, or many blocks, and may be located almost anywhere on the disk.

The "seedling, sapling, tree" structure does not apply to directories. There are no index blocks in directories, so each block contains linkage information pointing to the next block. The first four bytes of each block contain the block numbers (two bytes each) of the preceding and succeeding blocks in the same directory. SOS uses these pointers to find its way from block to block within the directory. Since the first block of a directory has no predecessor, its back-pointer is zero. Similarly the last block of a directory has no successor, so its forward-pointer is zero.

```
pOptionList     .word   OptionList      ; a pointer
OptionList      .block  03,00           ; another parameter list
pNameBuf        .word   NameBuf         ; a pointer
NameBuf         .block  10,00           ; buffer for name
 ;
begin   pop     return
        pla                             ; pull 4 dummy bytes for function
        pla
        pla
        pla
        pop     index                   ; pointer to string descriptor
 ;
        lda     #00
        sta     result                  ; initialize to zero
        sta     result+1
 ;
        moveB   addr+1601,index+1601    ; Xbyte of parameter pointer
        lda     #NOTNOW                 ; Basic interpreter routine #13.
        sta     dispatch+3
        jsr     dispatch                ; string data pointer --> index
        beq     exit                    ; b/ string length = 0
 ;
        sta     NameBuf                 ; string length
        ldy     #00
$1      lda     @index,y
        iny                             ; "index" points to 1st character
        sta     NameBuf,y               ;    (not to string length as
        cpy     NameBuf                 ;       required by GETDEVNUM)
        bne     $1
 ;
        lda     #02                     ; fill parameter list for SOS call
        sta     param
        moveW   pNameBuf,param+1        ; pointer to name buffer
        SOS     GETDEVNUM               ; returns with error number in A
        bne     exit                    ; b/ error - no such device
        lda     param+3                 ; device number
        sta     param+1
        sta     result
 ;
        lda     #04                     ; fill parameter list for SOS call
        sta     param
        moveW   pNameBuf,param+2
        moveW   pOptionList,param+4
        lda     #03
        sta     param+6
        SOS     DINFO                   ; the SOS call
        bit     OptionList+2            ; device type
        bmi     exit                    ; b/ yes it's a block device
        lda     #20                     ; else return error #32.
        sta     result
 ;
exit    push    result
        push    return
        rts
 ;
        .proc   ReadBlock,3
        .ref    return,param
        .def    pAddr,pIObuf,IObuf,IObufU
        jmp     begin
 ;
pAddr           .word   addr            ; a pointer
pIObuf          .word   IObuf           ; a pointer
IObuf           .block  100,00          ; our local buffer (lower 1/2)
IObufU          .block  100,00          ;   ditto (upper 1/2)
 ;
begin   pop     return
        pop     param+6                 ; blknum to start read
        pop     addr                    ; pointer to Basic's integer array
        pla
        sta     param+1                 ; devnum
        pla                             ; discard hi byte
 ;
        lda     #SERROR                 ; Basic interpreter routine #15.
        sta     dispatch+3              ; prepare for possible error
 ;
```

The link pointers occupy bytes [0...3] of each directory block. The remainder of the block, beginning at byte 4, contains a number of equal length file entries. In the present version of SOS each file entry occupies 39 bytes, so there is room in the block for the four link bytes and for 13 file entries each 39 blocks long. That totals 511 bytes. There is one unused byte left over.

Table 2.
Structure of File Entries.
Derived from the SOS Reference Manual

Byte#	Bytes	Contents
0	1	storetype & name length
1	15	filename characters
16	1	filetype (Basic program, Pascal Data, SOS, etc.)
17	2	key pointer (block number of first file block)
19	2	blocks used (total blocks in file - including index blocks)
21	3	end of file (total number of readable data bytes in file)
24	4	creation date
28	1	version of SOS used to create file
29	1	minimum SOS version to access file
30	1	access permitted (read, write, rename, destroy, backup bit)
31	2	auxiliary file type
33	4	last modification date
37	2	header pointer (points to first block of this directory)

The first entry in the first block of any directory is special.

The first entry in the first block of any directory is special. It is a directory header which contains the name of the directory and a variety of other information including the length of each file entry and the number of entries per block. The header itself has the same length as a file entry. Root directory and subdirectory headers are similar but differ in a few details; Table (1) shows their present configuration. Note that the bytes in this table are numbered from the beginning of the block, not from the beginning of the header. The four link bytes are included in these numbers.

Every block in a directory has four link bytes, but only the first block has a header. Following the header is a series of file entries. Since the directory header really occupies one entry position, the first directory block can list only twelve files, while succeeding blocks can each list thirteen.

Individual file entries contain the file's name and other information about the file. The structure is shown in Table (2). The first byte contains two pieces of information: the low nibble (lowest four bits) contains the filename length; the high nibble is a code describing the file's storage type. Possible storage type values are 1, 2, 3, and 13 (hexadecimal $D) which stand for seedling, sapling, tree, and directory, respectively. Witn this information, SOS knows what type of file structure to expect and how to go about accessing the file's data. If the file has been deleted then its file entry is considered empty or inactive. The first byte of an inactive entry is zero.

Byte numbers 17 and 18 in each file entry make up the key block pointer. This is the block number on the disk which contains the first block in the file. The order of bytes here is "low byte, high byte" in keeping with ordinary 6502 convention, as contrasted with the "high byte, low byte" order of two-byte integers in BASIC. Depending on the type of file, the key block pointer will point to a data block, an index block, a master index block, or the first block of a subdirectory.

The accompanying BASIC program begins by asking for the name of the directory you wish to rearrange. You must enter a complete pathname. If you want a root directory, then give only the SOS device name containing that volume. If you want a subdirectory then you must begin with the device name followed by "/" and the subdirectory name. There may be several levels of subdirectories. A trailing "/" is optional.

Examples:
.PROFILE .D2/SUB1/SUBSUB1

The name you enter must start with a "." or you will get zapped with an error message. All sorts of errors are possible if you really try. Some will stop the program, but none should destroy a directory.

Table 1.
Structure of Root Directory and Subdirectory Headers.
Derived from the SOS Reference Manual

Byte#	Bytes	Root Directory Header	Subdirectory Header
4	1	storetype and name length	same
5	15	filename characters	same
20	8	reserved	same
28	4	creation date	same
32	1	sos version used to create	same
33	1	minimum SOS version to access	same
34	1	type of access permitted	same
35	1	length of each entry	same
36	1	entries per block	same
37	2	file count (number active files)	same
39	2	volume bit map pointer	parent pointer
41	2	total blocks on volume	parent entry number and parent entry length

The program analyzes the pathname, extracts the device name, and calls function GetNumber in the invokable module to get the corresponding SOS device number. Then it starts loading directories, beginning with the root directory of the volume in the named device. Four blocks are loaded into memory.

If the pathname contains subdirectory names then the program searches through the root directory trying to match the next component name. If found, the program looks up the next subdirectory's key block pointer and loads it, reading a maximum of four blocks. This subdirectory is checked to verify that it is indeed a subdirectory and that you are not trying to pull a fast one. If necessary the program repeatedly loops back, searches for additional component names, and loads them... until the final target directory has been loaded into memory.

In its next phase, the program displays a catalog of the target directory. All available entry slots are numbered and presented in a list on the screen. Entry #0 is actually the directory header and contains the directory's own name. The names of all active files are shown in the positions they occupy; empty slots are represented by "———". Some of the files in the target directory may themselves be subdirectories, and such entries are shown with a "/" following the name. The "/" is added by the program; it is not present in the directory.

You have three options:

(1) press "R" to Restart the entire program.

(2) press "S" to Swap entries. You will be asked for two entry numbers, following which the entries will be swapped. With sufficiently vigorous swapping you can set the files in any order.

(3) press "W" to Write the revised directory back onto the disk. The disk directory is not actually changed until you use this option.

The Write option actually does more than just re-save the directory. If any file entries are subdirectories then certain other pointers must be updated. The file entry of a subdirectory points, of course, to the subdirectory itself located elsewhere on the disk. That subdirectory, in turn, contains in its header a "parent pointer" which points back to the directory you have so carefully rearranged.

It wouldn't be a problem if the parent pointer just pointed to the target directory's key block. But it doesn't, or at least it doesn't always.

(Cont'd on page 12)

```
        lda     #05             ; fill parameter list for SOS call
        sta     param           ;    transfer disk to IObuf
        moveW   pIObuf,param+2
        lda     #00
        sta     param+4         ; transfer 512 bytes
        lda     #02
        sta     param+5
        SOS     DREAD
        beq     $1              ; b/ no SOS error
        jsr     dispatch        ; invokes Basic's error routine
;
$1      ldx     #00             ; transfer IObuf to user
        ldy     #00
$2      lda     #00             ; hibyte (1st byte) of each Basic
        sta     @addr,y         ;   integer --> zero
        incadr  addr            ; increment pointer
        lda     IObuf,x         ; next byte from local buffer
        sta     @addr,y         ; to lobyte (2nd byte) of Basic int
        incadr  addr            ; increment pointer
        inx
        bne     $2              ; for first page of local buffer
;
$3      lda     #00
        sta     @addr,y         ; repeat for second page
        incadr  addr
        lda     IObufU,x
        sta     @addr,y
        incadr  addr
        inx
        bne     $3
;
        push    return
        rts
;
        .proc   WriteBlock,3
        .ref    return,param
        .ref    pAddr,pIObuf,IObuf,IObufU
;
        pop     return
        pop     param+6         ; blknum to start write
        pop     addr            ; pointer to Basic's integer array
        pla
        sta     param+1         ; devnum
        pla                     ; discard hi byte
;
        lda     #SERROR         ; Basic interpreter routine #15.
        sta     dispatch+3      ; prepare for possible error
;
        ldx     #00             ; transfer from Basic array
        ldy     #00             ;      to local IObuf
$1      incadr  addr            ; increment pointer (discard hi byte)
        lda     @addr,y         ; lobyte (2nd byte) of Basic int
        sta     IObuf,x         ; to next byte of local buffer
        incadr  addr            ; increment pointer
        inx
        bne     $1              ; for first page of local buffer
;
$2      incadr  addr
        lda     @addr,y         ; repeat for second page
        sta     IObufU,x
        incadr  addr
        inx
        bne     $2
;
        lda     #04             ; fill parameter list for SOS call
        sta     param           ;      transfer IObuf to disk
        moveW   pIObuf,param+2
        lda     #00
        sta     param+4         ; transfer 512 bytes
        lda     #02
        sta     param+5
        SOS     DWRITE
        beq     $3              ; b/ no SOS error
        jsr     dispatch        ; never returns
;
$3      push    return
        rts
        .end
```

Utility:

Text File Reader

robert c. saunders

READ.TFILE is a text file that can be EXEC'd to read any other text file. (You get a FILES BUSY ERROR when you use it to read itself).

It uses unlikely variables (AAAA$, BBBB$(1000),HHHH,QQQQ,VVVV, ZZZZ) and uses lines 63900 to 63904 so as not to disturb a program in memory. These variables or line numbers can be changed if they conflict with others you are using.

It erases itself at the end.

If you don't view the file, the screen will look the same with the cursor returned to where it was, in most cases.

You will be left with QQQQ lines of text stored in BBBB$(QQQQ).

Some notes on the program:

1. PRINT CHR$(21);0
   is a screen control code that makes the text screen equal one character. It is used here to avoid seeing the prompts that are visible for each line of a text file that has been EXEC'd. This avoids scrolling the existing contents of the screen.
2. OPEN #1,".CONSOLE"
   is used to allow input from the keyboard and avoid taking input from the EXEC file itself. This treats keyboard input as a file that has been opened and so requires INPUT#1 and PRINT#1 statements.
3. PRINT CHR$(29);
   at the end, clears the screen from the position of the cursor to the bottom.
4. When you wish to EXEC READ.TFILE a second time, type in CLEAR (RETURN) first or a REDIMENSION ERROR will occur. Note: this will clear all variables in memory!
5. If the program tries to read a file that doesn't exist or that has no content, all variables will be cleared and you will be prompted again for the Pathname. To avoid this, delete line 63902.

```
63900   TEXT:WINDOW 1,23 TO 80,24:HOME:PRINT C
        HR$(7);:OPEN#1,".CONSOLE":PRINT#1"Path
        name?  :  ";:INPUT#1;AAAA$:OPEN#9,AAAA
        $:QQQQ=0:DIM BBBB$(1000):ON EOF#9 GOTO
         63902
63901   INPUT#9;BBBB$(QQQQ):QQQQ=QQQQ+1:GOTO 6
        3901
63902   IF QQQQ=<0 THEN PRINT CHR$(7);CHR$(7);
        CHR$(7);:CLOSE:CLEAR:GOTO 63900
63903   HOME:PRINT#1"You have read ";QQQQ;" st
        rings of text from a file called ";AA
        AA$;"'.":PRINT"Want to see it now?  (Y
        /N)  : ";:INPUT#1;AAAA$:IF AAAA$<>"Y"
        AND AAAA$<>"y" THEN END
63904   TEXT:HOME:FOR ZZZZ=0 TO QQQQ:PRINT#1;B
        BBB$(ZZZZ):NEXT:PRINT#1:PRINT#1"Want t
        o see it again?  (Y/N)  :  ";:INPUT#1;
        AAAA$:IF AAAA$="Y" OR AAAA$="y" THEN 6
        3904:ELSE END
```

Directories/from page 11

It actually points to the particular block within your target directory which holds the listing. Furthermore, the subdirectory header stores the entry number within that parent block, not the overall entry number in the parent directory. Since you have undoubtedly swapped the entry listing into some other slot in a different block, the pointers will have to be updated. It's particularly exasperating because, as far as we have been able to determine, SOS doesn't even use this feature. Perhaps someone just thought it was a good idea to throw it in.

Updating the pointers is quite simple. The program scans the target directory looking for subdirectory file entries, while carefully keeping track of the block number it is scanning and of entry numbers within that block.

Subdirectory entries are recognized by finding the value thirteen (hexadecimal $D) in the high nibble of the entry's first byte. The appropriate key block pointer is obtained. One block from the subdirectory is loaded into an auxiliary buffer "key%". New pointer values are quickly stored in their proper locations, and the block is written back onto the disk. It takes only about one second.

We hate to admit it, but the program will run even if you omit all our terse, yet pithy, remarks. You will, however, have to type in something for the even hundreds line numbers (1000, 1200, 1400, etc.) since they are the destination of various GOTO and GOSUB instructions; a single ":" will do.

It is, of course, almost impossible to imagine that we, or you, or even this magazine might have made a typing error. But just on the off chance, perhaps you should try out the program on a copy of one of your floppies. Make sure it still catalogs before committing your hard disk.

Review ON:

Keystroke Data Base and Report Generator

john r. sollman

Included in the purchase of my Apple /// system in 1982 was Quick File ///, an interim solution for my data management needs. Potential opportunities to do work which involved cataloging or researching large amounts of data prompted my search for a larger and more flexible program. After evaluating several data bases including Apple File ///, I ultimately settled upon *Keystroke Data Base and Report Generator* by Brock Software Products of Crystal Lake, IL.

The product was purchased with the expectation that there would be some relational capabilities between files for some of the applications I had in mind. Purchase decision was based upon promotional material and about three hours of reading the manuals and experimenting with the program at a dealership. Initial evaluation of the product was hampered by its general non-availability on dealers' shelves, and by dealers' general unfamiliarity with the program.

Keystroke is a comprehensive and powerful data base...

Keystroke is a comprehensive and powerful data base with considerable flexibility. Maximum capabilities, as stated by the manufacturer, are 32,000 records per file, and up to 90 fields and 4,096 characters per record. Up to four key fields may be designated, and up to eight different sets each of search parameters and print specifications may be saved for each file. The Report Generator comes with set-up data for a popular serial printer and the Apple DMP parallel printer. It has the capability to store set-up data for several additional types of printers. Both the Data Base and the Report Generator are completely menu driven.

In addition to its stated ability to create large files, Keystroke offers the ability to convert PFS files to Keystroke, merge data directly from Quick File /// and DIF files, and create files or reports which can be loaded directly into VisiCalc. The revised version of the program (Version A-1) includes also a Word Juggler Form Letter merge capability. The Data Base comes with a loader diskette, a program diskette, a work diskette, and a concisely written manual. A separate manual is provided with the optional Report Generator. The manuals are logically organized and tabbed into sections covering each category of operation. The tabs are numbered, not named, which is an inconvenience. No tutorial is provided (Gott sei Dank!), but there are help menus available throughout the program. The loader diskettes for the current version (A-1) also contain a QUARK.INSTALL file for those who have Catalyst Version 2.0. The Data Base generally retails for $249 and the companion Report Generator for $149.

The program as initially purchased from the dealer had some problems. Certain search routines did not function properly, and there were other problems as well. The staff at Brock was most gracious in working with me on these problems, and they immediately sent me revised loader and program diskettes which seem to have corrected most of the problems initially encountered. At the same time, Brock was shipping replacement diskettes to all dealers and owners of record. The corrected version is A-1, and this version number should appear in the welcome message for both the Data Base and the Report Generator. Purchasers should also catalog their loader and program diskettes to ensure that the following files are dated January 16, 1984, or later:

/KEYDATA.LOADER	/KEYDATA.PROGRAM
KEYDATA.A.LIB	KEYDATA.CODE
	SYSTEM.STARTUP
	KEYDATA.HELP

/KEYREPT.LOADER	/KEYREPT.PROGRAM
KEYREPT.A.LIB	SYSTEM.STARTUP

The people at Brock must be commended for their willingness to deal directly with purchasers of their product. This inspires confidence in the product and the intentions of the manufacturer, and goes a long way toward offsetting any problems a user might initially experience. I wish that more software and hardware developers would emulate Brock's example. (The guys at Quark are great, too.)

...[it] requires 256K and a second drive.

Keystroke requires 256K of memory and an external Disk /// or similar drive, or a ProFile hard disk. Keystroke cannot be booted from the internal drive alone. The program diskette must be installed in the external drive. If a ProFile is connected to the system, Keystroke will transfer its program files to the hard disk when first booted. Thereafter, an external floppy disk drive is not necessary except when running without the ProFile.

Keystroke Data Base and Report Generator are large programs written in Pascal. The loader and program diskettes are almost completely full. There is little, if any, room to reconfigure driver files or add drivers such as for a Micro-Sci A143. The manuals give no information on the creation of a two-stage boot diskette, and without this information such action would be risky because the programs are copy-protected. No backup loader diskettes are furnished until the buyer returns the registration cards.

The program allows for only 36K of memory space for driver files. My system, which includes a ProFile, Catalyst Version 2.0, a print spooler and a Micro-Sci A143 has almost 34K tied up in driver files, even when making maximum use of Catalyst's dynamic driver loading feature. Formatting drivers for the Micro-Sci do not support dynamic loading and therefore must be on the boot diskette. Brock advises that my system is operational but marginal because of the size of the driver file. This leaves little room for further system growth.

The presence of the QUARK. INSTALL file on the loader diskettes made installation of the program on Catalyst a snap. Following installation, however, a message appeared on the screen advising that Keystroke requires the formatting drivers to be dynamically loaded from Catalyst 2.0. Fortunately, this is not an absolute requirement, but only a suggestion as a means of saving space in the driver file. If it were an absolute requirement, it should be stated up front in the manual and not after the user has altered his boot diskette by installing the program in Catalyst.

Keystroke will not work properly with the Micro-Sci A143 when both .Xx and .Dx drivers are active for the same drive. (This is not mentioned in the Keystroke manuals.) Keystroke will not run at all if the .Grafix driver is present in the operating system. (This is mentioned in the chapter on error messages.) My initial attempts to write to .D2 produced Error Message No. 32. The instructions following the error message advise the user to press Return or Escape. Pressing Return enables one to complete the write (creating a subdirectory, or folder, as it is called in the program). Going on to create a file within the newly opened folder produced a stack overflow. Escaping from the stack overflow ultimately invoked a "System Failure" message. The condition was easily remedied (after rebooting) by inactivating the .Xx driver for my Micro-Sci. While it is certainly not necessary that the program recognize the .Xx driver, it should be modified, if possible, so that it does not crash if both .Dx and .Xx drivers are active for the same Micro-Sci drive.

My evaluation of the Data Base and the Report Generator is based upon the creation and manipulation of several files of differing sizes. Size ranged from 38 records to a catalog ultimately consisting of about 2,600 records with two key fields. The small file contained 12 derived or computed fields. The large catalog presently has nine fields on one page, two key fields, and a maximum of 135 characters allowed per record. The typical record contains about 40 characters. This file is stored on a Micro-Sci diskette and consumes 973 blocks of storage space. I estimate this catalog could contain up to 2,800 records before it would outgrow the Micro-Sci diskette.

I have performed most of the functions which would normally be used routinely. This includes creation and maintenance of files, editing of entry forms, copying and backing up of files, establishing and storing of report formats and record selection protocols, and printing selected records to spool, printer and disk. In addition, I have done Quick File, VisiCalc and Word Juggler merges, which would probably not be used in day to day operation of the system. I have not yet merged DIF files, nor have I created Keystroke memory sequences. The system, in spite of its lack of tutorial, is quickly learned and soon becomes second nature.

I performed the functions which would be used routinely.

The system is quickly learned...

For the most part, the system functioned as expected. As noted earlier, both the Data Base and Report Generator initially had some problems with search and sort routines. Following reinstallation of the corrected diskettes which Brock furnished me, all search and most sort routines performed correctly. Following are some of my experiences using the Data Base and the Report Generator.

Creating an Entry Form: The program allows up to nine pages and 90 fields in an entry form. With the ability to create 32,000 records in a single file, and enter as many as 4,096 characters in each record, a truly sizeable data base can be created, limited only by the disk space available to store it. Creating the entry form should be *very carefully planned* if a large file is anticipated. Editing the entry form for a very large file can be quite time consuming.

Creating the entry form should be carefully planned if a large file is expected.

The entry form is created by naming fields and designating field length or specific format (telephone number, etc.). The form can be embellished by adding titles and information lines. Numeric fields may be set to increment automatically as records are added. It is possible to copy the entry form from another file and modify it, or to copy selected information from another file. This is a handy tool when building a system of interrelated files to support a business or cataloging operation. Enhancement menus permit establishment of default responses, designation of key fields, creation of derived fields, and establishment of cross reference to information in another file.

The Keystroke manual describes the ability to cross reference or validate as a feature which makes the data base "completely relational." This, of course, is not so. The relational aspects of Keystroke are quite limited when compared with truly relational data bases, and the data base should not be described as being relational on the basis of this feature alone. It appears that some Madison Avenue hype may have crept into the manual.

Merge of data to or from certain other types of files is one of Keystroke's features. My experience with this function is as follows:

Quick File Merge requires that the user first enter Quick File and write a report to disk using Open Apple-V for each field to be merged. (This writes the field title as well as the data.) Names of fields must match exactly. Quick File merge was first performed with the original version of Keystroke, and repeated several times since receipt and installation of the corrected diskettes. I transferred from a Quick File in .D2 to a Keystroke file in .D2. Field names were identical, but the sequence was not. The data in the first Keystroke field (second Quick File field) did not copy. During the transfer, some spurious information appeared and remained on the screen. The people at Brock felt that field sequence should make no difference, but suggested that this function would probably perform better with the corrected version. Subsequent Quick File transfers worked satisfactorily, except that all text was written to Keystroke in upper case. One such transfer produced a condition which later resulted in the pointer attempting to read beyond the end of the Keystroke file.

VisiCalc Merge, done with the corrected version, performed flawlessly. I merged the small file with 12 derived fields. When transferring to VisiCalc, the program does not write the formulae for the derived fields, but simply performs the calculations and merges the data. Fields are interpreted as columns and records as rows. The program writes field names with a /FR format, and you may elect not to transfer the field names. The limitation in size of file to be transferred should be apparent to any VisiCalc user. You may designate row and column coordinates for the upper left corner, which permits positioning different files in different parts of the VisiCalc worksheet. This allows one worksheet to be loaded on top of another to make a consolidated report. Some reformatting in VisiCalc is generally necessary. However, it is possible to format a blank VisiCalc worksheet with column headings and formulae, and load it over the files merged from Keystroke. You may merge to either Standard or Advanced VisiCalc.

Word Juggler Merge also performed flawlessly. A mailing list of selected records and fields was prepared and stored in Word Juggler format. Following its creation I verified that the field count and fields were correct, and that all the data were present. This feature was not present on the original version of Keystroke, and its addition is a boon for Word Juggler users. Once again, the number of records which can be transferred is governed by Word Juggler's limitation of 1,536 lines in memory. Each field occupies a separate line.

Following is my experience with other features of the Data Base:

Editing of the Entry Form: Entry forms can be edited after data have been entered. The existing file is copied to a work space, then copied back to the revised entry form one record at a time. Entry forms for very large files must be carefully planned. A file consisting of 1,407 records and four key fields took two hours and three minutes to copy back with the screen turned off. Long copy time was due in part to the presence of four key fields. Reducing the number of key fields to two reduced the copy time to 90 minutes. In a file of approximately 100 records, copy time was only a few minutes. Early records copy fairly quickly, but the rate slows as additional records are copied.

Keystroke does not have the flexibility of Quick File ///.

Editing of an entry form is fairly easy, but Keystroke does not have quite the flexibility one finds in Quick File ///. Quick File permits moving single entries up, down, or sideways. In Keystroke, one may delete an entire line or copy it to a new location. Single entries may be moved laterally but not up or down. An editing procedure more akin to Quick File would be most welcome.

Original Data Entry is accomplished by selecting the "Add" option from the menu. The entry form for my catalog, as originally designed, had three key fields. After about the 100th record, each record was saved as it was created. Addition of a new fourth key field slowed the entry procedure significantly. With four key fields and a file already consisting of 1,407 records, data entry in the newly created key field resulted in a save time per record of from 1.5 to 2.5 minutes. At that rate, it would have taken between 40 and 60 hours to make entries in the newly created key field in all 1,407 records. Reduction to only two key fields improved performance markedly. The file was then saved after the addition of about 18 to 20 new records, and saved time was only a few seconds. The manual advises against establishing more key fields than necessary, but does not say why. The more key fields there are, the longer it takes to add or update information.

The more key fields there are, the longer it takes to add or update data.

Find-Update allows records to be located and updated quickly. Search criteria are entered in each field which will serve as a basis for selection. The program provides very simple language and great flexibility for establishing search ranges or record selection criteria. Derived fields are calculated when in "Find-Update", but automatic entries and defaults are not operative. It would be helpful to be able to elect use of established defaults for certain fields when in Find-Update. A user would often want the auto-date to show the last date the entry was *updated*, rather than the date the record was *first created*. Validate/Cross Reference is operative only when in the "Add" mode. When a record is updated, any change in a cross-referenced field will not be carried forward, and must be looked up and entered manually. Find-Update should permit the user to make Validate/Cross-Reference active or inactive.

Backing Up Files: Each file to which new information has been added shows a "+" following the block count in the Keystroke file catalog, until it has been backed up or copied using a Keystroke procedure. The Keystroke procedure for backing up files does not allow the user to select the folder (subdirectory) and file into which the backup is to be made. When backing up to a disk in drive 2, the backup file is written to the root directory rather than to a designated subdirectory (folder) on the diskette. I find it more convenient to use the System Utilities or the Pascal Filer to back up the files.

Printing: The Data Base permits the creation of a "Labels" type of report. The format is selected by using three symbols to designate the fields to be printed and the manner in which the report will be formatted. Sequence of the fields cannot be altered. Record selection capabilities are quite extensive, but sort capabilities are available only in key fields. Up to eight record selections and eight print formats may be saved for each file. The so-called "Tables" report format which can be created in this manner is actually a one-line "Labels" report.

When printing labels from the Data Base, an extra line feed is inserted after every label. If a six line label is to be printed, a five line page must be specified. This was true on two different printers, and even when printing to disk. Single-line labels would appear double spaced when specifying one line per page. This was true even on the corrected version of the program.

The Report Generator allows for the creation of only a "Tables" type of report. It has a rather elaborate series of set-up menus to select records to print and to designate report format. Up to eight record selection protocols and eight report formats can be established and saved for each file.

With one exception, record selection and sorting protocols worked smoothly. On one occasion I sorted a report in descending order on a computed field. (Not a computed column.) The largest number in this field was 1,422, all other numbers being either two or three digits. The sequence produced by the Report Generator was 188, 186, 179, 145, 1,422, 139...

The general format of the program-generated report heading is fixed, in that there will always be a date and a page number at the top of every page, whether it is wanted or not. The user can change the report title, subtitle, and footer. The user should be permitted to omit date and page number, or to locate them elsewhere on the page. The user has considerable flexibility, however, in formatting the content of a report.

It is possible to create reports containing computed columns, as well as reports containing summary data and totals only. The user can adjust the width of columns, determine the order in which the fields will appear in the report (except that a computed column must be to the right of the columns from which it gets its information), establish format within the columns, and designate summary statistics to be printed, such as totals, averages and counts. It is possible to designate a new page at each point where there could be a subtotal, but it is not possible to designate a line break at this point without actually printing a subtotal, subaverage or subcount.

When attempting a summary report which gave only a count, the report generator printed the field being counted for each record before giving the "summary" information. Thus, I had a 1,407 line "summary" report to tell me that there were 1,407 individual titles in my catalog. This problem has not been corrected on the revised version (A-1). When designing a report format using several computed columns, the second, using information from the first to make its calculation, crashes the program. No error message is generated to warn the user that an illegal procedure is being attempted.

As delivered, the Report Generator contains printer set-up options for a popular serial printer and an Apple DMP parallel printer. Set-up options may be stored for several additional printers, or several different sets of options may be prepared and stored for a single printer using different printer names. This gives the experienced user considerable flexibility in printer set-up. These printer option menus also permit the user to select default widths for printing on wide and narrow paper in standard and condensed print. There is an elaborate set of menus to select printing devices, enter printer set-up options, and create report enhancements. The ability to select top, bottom or side margins from a menu, however, is conspicuous by its absence.

Side margins can be set by modifying the default printing width for each printing mode, so that an allowance is made for left and right margins. You then add the necessary escape codes for left margin to the printer set-up string for each printing mode (standard or condensed). A vertical tab setting may also be entered to create a top margin, or the paper may simply be positioned by hand. The program will then print the report heading to the full width selected for the print mode in use and center the report body below the heading. If the report body is wider than the preselected print width, the overflow columns to the right will be printed on a following page. (The original version of the Report Generator simply truncated the overflow columns.) While the steps enumerated above will permit an experienced user to set margins and center his reports, it would be far preferable to accomplish this by menu selection. The menu-driven structure of the data base and report generator suggests that these programs were developed for use by "average" people, not "computer experts."

Some printer set-up options prove to be a bit confusing.

Some printer set-up options tend to be a bit confusing. The user cannot determine, for instance, whether a "yes" response to the options to force line feed or form feed means that the printer forces these actions or that the program is being asked do it.

When designing the report layout, the user is aided by a ruler at the top of the screen which indicates the numbers of columns used. When setting count, total or average functions for a field near the left edge of the paper, care must be taken that the report does not become overly wide. Part of the total or count line can extend into what you think is your margin, effectively moving the entire report slightly to the right. If it moves far enough to the right, the rightmost columns of your carefully formatted report may end up on additional sheets of paper.

Both the Report Generator and the Data Base give the user the option to print to a file on disk. The file to be printed in this manner is a Pascal text file. The program does not incorporate the capability of the Pascal Editor to write an ASCII file. Pascal text files are not readily readable by some word processing programs. The ability to write ASCII files would have been most welcome.

The manual does not provide much detail about printing a report to a file on disk, and there are a few surprises in store for the first-time user.

Once a disk file is designated, the program looks upon it as the default printing device. When selecting a file name which is identical to a file existing on disk, the program produces no warning that a file is about to be overwritten. The program does not ask for a new file name each time a file is to be printed to disk. In order to prevent prior reports in a series from being overwritten, it is necessary to go through a lengthy series of menus in order to designate a new print file for each report.

When writing a report to disk, the Report Generator defaults to "Wide Paper" (132 columns) for the report format. The default for using a printer is "Narrow Paper" (80 columns). If you have been using "Narrow Paper" for writing to the printer or spooler you are not expecting a default to "Wide Paper" for a file on disk. The "wrap around" which this produces consumes extra memory and creates time-consuming editing problems. The alternative is to go back to Keystroke and reprint the report making sure to use the "Narrow Paper" option. Either way is needlessly time consuming. Reports printed to disk are paginated with a header at the top of each page. The user has no option to change this. When printing to disk, the report headers in the "Narrow Paper" format are 80 columns wide. The carriage return character in position 81 causes a wrap-around of every line of the report header. Why it was decided to format the header 80 columns wide is absolutely beyond me. 79 columns would certainly make more sense. The user is powerless to change this.

It requires slightly over an hour to print to disk a three-segment report consisting of 1,314 lines and five fields. This includes the time necessary to go through the menus to select a folder and unique file name for each segment of the total report, to select "Narrow Paper" from the Print Options menu, and to search the entire file for the records to be printed. To print the entire report to disk in one segment would require 36 minutes, but would produce a file too large to load into Word Juggler. Performance, timewise, with smaller files which would fit on a 143K floppy is significantly better.

When returning to the use of a printer, the program does not default back to Narrow Paper. An unsuspecting person could find half of the next report being printed on the platen, which could also damage the print element when it crosses the edge of the paper.

The process of sorting and printing a large file to the spooler is painfully slow, even when the file is sorted on only one key field. It takes about 25 minutes to sort and write a 25 page report (about 1300 records) to the spooler, during which time the computer is not available for other uses.

Sorting and printing... is painfully slow.

Interestingly, it is a bit faster to print directly to the printer, since print and sort routines are performed simultaneously. Intense I/O activity during this process has occasionally caused loss of several lines of data. I no longer use the spooler with Keystroke.

Keystroke Report Generator prints only the first encountered of several identical entries for a sorted field. This is fine for many, but not all purposes. It is certainly not suitable for a catalog. The user has no option to print all entries in a given field, whether identical or not.

The Manual, thankfully, is devoid of those dreary and unimaginative tutorials. This is the manual's strongest point. It is concisely written and for the most part is quite clear. There are, however, some points which can be improved. It would have been most helpful if Brock were to have provided a diskette containing several small files so that the new user could practice what is learned from the manual without first having to create files of adequate length for demonstration purposes. The Work Disk provided with the Data Base could contain sample files which could later be deleted.

The manual should contain a compendium of permissible keystrokes and commands which can be invoked by the user. Nowhere, for instance, does the manual tell what "Open-Apple F" does. (It turns off the screen in order to speed up some operations.) The "F" was intended to stand for "Fast" when turning off the screen. (I think it really stands for "Frog" because of the croaking noise it evokes from the speaker when the screen is turned off. The Brock people tell me that this is to let the user know that the program is working. However, the manual says nothing about speaker noises, and it can come as a surprise the first time.)

Some passages in the manual imply a mandatory requirement when actually only a suggestion is being made. Examples are the numerous references to the specific drives which floppy or ProFile users should use for copying or merging files.

The manual should be more precise in explaining the reasons certain things are done or not done. As an example, the manual advises that no more key fields be set up than necessary. No reason is given. The manual should advise that addition of key fields slows overall program performance. Readers should also be given some suggested rationale for designating key fields.

This leads to my major criticism of the manual. It appears that some marketing or public relations people may have had a hand in its final wording, in that it treads very lightly upon the limitations of the system while extolling its virtues. The description of the system as "completely

It does not detract...to be up front...

relational" is an example. Failure to caution users about slow performance with very large files, or with several key fields is another. It takes careful reading of the manual in order to assess some of the program's limitations. A review of the manual at a software dealership was certainly not sufficient to give me enough information to determine whether Keystroke would serve my original purpose. (It would not, but I had effectively abandoned it for other reasons by the time I purchased the software.)

It does not detract from the merits of a system to be up front about what it will or will not do. Certainly, being more forthright would enable the prospective buyer to determine at the outset whether the system is suitable to his or her needs and compatible with the existing system configuration. It serves no purpose to buy a system only to discover that its hidden limitations or special requirements make it unsuitable for your particular needs. Any reasonable person should realize that no data base program can be perfectly designed. Compromises must always be made in order to achieve a particular design philosophy. Keystroke is an excellent program for many applications, but not for all. Being more forthright about this would certainly help prospective buyers make intelligent and informed choices of software. We need to sell to the *right* market, not to the *entire market*. If software producers do not shape up their documentation and their implied claims, our friendly neighborhood U. S. Government will come down hard on them just as they did on the Hi-Fi manufacturers a few years ago for their inflated performance claims (100 watts/channel ±1 DB).

Summary and Conclusions: The Data Base functioned well with small to moderate sized applications. After a few false starts with the large file, the system functioned slowly but otherwise reasonably well. The chief deterrent to its use in large files is the time it takes to edit and copy files, and print reports. Disk space is a consideration for any large file. My large files are stored on Micro-Sci diskettes. If the largest anticipated file will fit on a 143K floppy diskette, the Data Base should perform superbly. If on the other hand a file of 32,000 records and four key fields is anticipated, Keystroke is not for you. (I am told that the versions for the Mac and the ex-Lisa are quite fast, in view of the microprocessors involved.)

With some exceptions, the system of menu selection was quite convenient and logical. A menu bar is placed over the desired entry and selection is made by pressing "Return" or "Enter." In first entering Keystroke, you move through several menus until the file is selected. The use of Keystroke Sequences (macro commands from which the program derives its name) will speed through the selection procedures for many applications. The Keystroke Sequences have their limitations, however. Because the program is menu driven, any change in the order of entries in a menu may change the number keystrokes needed to position the menu bar. Unless the affected Keystroke Sequences are modified accordingly, some interesting and surprising results could occur.

File creation and data entry worked well...

The file creation and data entry routines worked well and were easy to learn. The use of the key fields in "Find Update" resulted in rapid and almost instantaneous location of files for updating. In the revised edition of the program, record selection procedures worked without a hitch and were a pleasure to use. When working a large file, I find it much faster to copy it to the ProFile for updating or adding new records, then to delete it from the ProFile after the update is copied back to the Micro-Sci. This saves some time, since the periodic saves to disk are accomplished much faster to the ProFile than to the Micro-Sci. My only serious criticism of the Data Base, aside from the time required for certain functions, is its lack of defaults and automatic entries in "Find Update."

...the Report Generator left much to be desired...

In contrast with the Data Base, the Report Generator left much to be desired. The only report option available is a Tables type of report. The report heading format is fixed by the program, and the user has only limited ability to modify it. Some rather subtle and surprising things can occur to the unwary first-time user. These are not explained in the manual. I refer to the 80 column heading format, the widening of the report by encroachment of certain summary information into the left margin, the inability to set margins for the printer by menu selection, the inability to specify line breaks between minor report categories without displaying statistical information, the surprise default to "Wide Paper" when printing to disk, and the cumbersome manner of printing sequential reports to disk.

Also, the user must remember too many little things when printing a file to disk. While Keystroke Sequences can be developed and appropriately chained to do this, quite a bit of thought would have to be expended to get this function just right. Each modification of record selection protocols or report formats could change the order of menu selections and require revising the Keystroke Sequences.

Once record selection protocols and report formats are created and stored, the printing of reports is relatively easy. The use of Keystroke Sequences for these routines would speed them further. Again, the user must be alert to any changes which would alter the content or order of the menus.

(Cont'd on page 20)

Utility:

Disk Drive Cleaning

paul m. stenberg

When I purchased my disk drive cleaner, my dealer told me the easiest way to clean the disk was to go to Apple][emulation mode and do a catalog. This worked but I did not like the idea of having to once again pretend I own an Apple][. Also I thought that the cleaner would last longer, if I used different areas on the cleaner instead of always using track 17 (Apple][VTOC). So I created the program CleanHeads.

The program asks for the drive to be cleaned and which track on the cleaner is to be used. It then prompts you to place the cleaner in the drive. Remember, the head on an Apple /// drive is on the bottom and the cleaning side of the cleaner should face down when inserted into the drive. When you press "ENTER" the program tries to read four blocks off the diskette, which causes the head to touch the surface of the cleaner and be cleaned. The program starts over by asking for the drive to be cleaned. After you have cleaned all the disk drives type 0 and the program will end, but first it will remind you to write the date on the disk drive cleaner so you will know when you last cleaned your drives.

When an invalid drive is entered, the procedure GET_DRIVE issues an error message and then calls itself. This is an example of direct recursion. Recursion is one of the things that PASCAL does best, but when using recursion you must be sure there is a terminator or in theory you will put the computer into an endless loop (on the APPLE /// you will eventually get an error message indicating that the stack is full). In GET_DRIVE the recursion is terminated when a valid drive number is entered. The procedure GET_TRACK uses recursion in the same way.

Once you have entered the program and tested it, follow the procedure on page 185 of the Apple /// Pascal Introduction, Filer, and Editor Manual to make a Turnkey Diskette. Then when you need to clean your drives you will only need one diskette. The program CleanHeads does not use any intrinsics, so you do not need the file SYSTEM.LIBRARY. It also does not use a special character set so you do not need the file SYSTEM.CHARSET.

Now you can clean your drives using the power of the Apple /// and as a bonus your disk drive cleaner will last longer.

Editors note: We do not recommend cleaning disk drives more often than every six months to a year unless they have extraordinarily heavy use. This is because of the [intended] abrasive nature of the cleaning disks. On our own drives, one of which is approaching eight years of service without failure, we have cleaned heads only twice, notwithstanding our heavy use, and have attributed no failures to dirty heads.

```
filename: {$LIST+}
PROGRAM CLEANHEADS;

{ utility CleanHeads
     by Paul M. Stenberg
     This program allows you to select the drive to be cleaned
        and the track to be used on the cleaning diskette. }

TYPE
   DRIVE_TYPE= 1..4;                    { range of drive numbers }
   TRACK_TYPE= 1..30;                   { range of track numbers }

VAR
   ALL_CLEAN: BOOLEAN;
   CH: CHAR;
   DRIVE: DRIVE_TYPE;
   TRACK: TRACK_TYPE;

   PROCEDURE GET_DRIVE(VAR NEW_DRIVE: DRIVE_TYPE);

   VAR
      ANS: INTEGER;

   BEGIN
      WRITELN;
      WRITE('Enter drive to be cleaned? (0=EXIT, 1-4) ');
      READLN(ANS);
      IF ANS = 0 THEN BEGIN      { check for exit }
         ALL_CLEAN:= TRUE;
         NEW_DRIVE:= 1;          { set to a valid number before exiting}
         EXIT(GET_DRIVE)
         END;
      IF (ANS < 1) OR (ANS > 4) THEN BEGIN   { check for invalid answer}
         WRITELN('Drive must be a number between 0 and 4');
         GET_DRIVE(ANS)                  { if invalid, try again }
         END;
      NEW_DRIVE:= ANS
   END;  { GET_DRIVE }

   PROCEDURE GET_TRACK(VAR NEW_TRACK: TRACK_TYPE);

   VAR
      ANS: INTEGER;

   BEGIN
      WRITE('Enter track to be used? (1-30) ');
      READLN(ANS);
      IF (ANS < 1) OR (ANS > 30) THEN BEGIN  { check for invalid answer}
         WRITELN('Track must be a number between 1 and 30');
         GET_TRACK(ANS)                  { if invalid, try again }
         END;
      NEW_TRACK:= ANS
   END;  { GET_TRACK }
```

Keystroke (from page 18)

While the statistics regarding the maximum capacities of the Data Base and Report Generator are impressive as to the volume of records and files possible, the use of this system for very large files is impractical. It simply takes too long to perform search, sort and print routines in the very large files. The system is well suited to files of 143K or less. The system was not designed with the use of high capacity floppy disk drives in mind. Also, allocation of memory space allows only 36K headroom for driver files. This effectively limits the ability to add the high capacity storage necessary to capitalize upon the stated capacities of the program.

The system is well suited to files of 143K or less.

My feeling about the well-meaning people who design off-the-shelf data bases is about like my feelings toward the well-meaning people who design auto expense record books. They don't travel much on business, and do not get input from people who do. What they think will be useful often does not accurately reflect the needs of potential users. Keystroke Report Generator is full of things which one would think would be nice to have. Date and page number on every page, and printing of only the first encountered of several identical entries in a sort field are examples. From a practical standpoint, much additional work is needed to refine user options and make Keystroke the useful report generator it was intended to be.

...additional work is needed to refine user options...

One additional factor must be borne in mind when considering purchase of the Keystroke Data Base and Report Generator. Brock advises me that no further upgrading of the Keystroke programs for the Apple /// will be undertaken because of Apple's unfortunate decision to terminate research and development of the Apple /// line. This means that the program is presently in its final state as far as the Apple /// user is concerned.

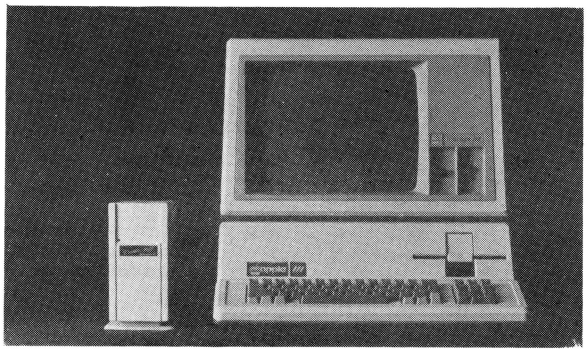
This brings me to my greatest area of concern. The people at Brock are great. They take the time to listen to people who use their products, and welcome constructive criticism. They were interested in my reactions to the Report Generator and Data Base, and expressed a willingness to correct what was feasible in future upgrades of the product. This was truly refreshing. However, Corporate Apple has thrown a wet blanket over the whole thing, not only for Brock, but, I would imagine, for just about everyone else as well. Brock now has no further plans to upgrade this or any other product for the Apple ///. In addition to their willingness to consider such things as available headroom for driver files and better support for the Micro-Sci, Brock had also planned to produce a companion forms generator and graphics program. Because of Apple's on-again-off-again support policy for the Apple /// (which is now off-again), developers like Brock now find it unprofitable to develop, refine and upgrade products for the Apple ///.

Companies like Brock need to be convinced that Apple Computer support or not, there are over 100,000 ///'s out there, still a sizable market with loyal *users who will buy almost anything that runs on a ///. ...ed*

Keystroke has the latent potential to become an excellent data base program. We Apple /// owners will have the privilege of watching companion products from Brock and other software developers continue to be developed and refined for the Apple //e and the MacIntosh. Were it not for Apple's unfortunate decision to cease development work on the Apple /// line, we Apple /// Keystroke users could have looked forward to the development and refinement of additional user options which would have capitalized upon Keystroke's many latent capabilities. Keystroke has great potential, but it still has a lot of growing up to do. Regrettably it now appears that present users will never be able to realize the full potential of this program. Hopefully, the people at Brock will find it in their hearts to throw in a bit of code for us Apple /// owners as they develop and refine the versions for Apple's other products.

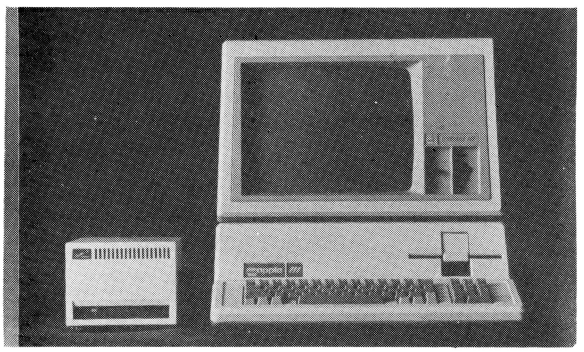
```
PROCEDURE ACTIVATE_DISK(DRIVE_NUM: DRIVE_TYPE; TRACK_NUM: TRACK_TYPE);

VAR
   BLOCK_NUM,
   FIRST_BLOCK,
   I,
   LAST_BLOCK,
   UNIT_NUM: INTEGER;
   ANS: STRING;
   BLOCK_512: PACKED ARRAY[0..511] OF CHAR;  { buffer to hold the disk block}

BEGIN  { ACTIVATE_DISK }
   FIRST_BLOCK:= TRACK_NUM * 8;          { convert track to block }
   LAST_BLOCK:= LAST_BLOCK + 4;          { try to read half the track }
   CASE DRIVE_NUM OF                     { convert drive number to unit number}
      1 : UNIT_NUM:= 4;
      2 : UNIT_NUM:= 5;
      3 : UNIT_NUM:= 9;
      4 : UNIT_NUM:= 10
      END;
   WRITELN;
   WRITE('Place Cleaning Diskette in Drive #',DRIVE_NUM,' and press "ENTER"');
   READLN(ANS);
         { the time required for cleaning is from 10-15 seconds }
   FOR BLOCK_NUM:= FIRST_BLOCK TO LAST_BLOCK DO
      UNITREAD(UNIT_NUM,BLOCK512,512,BLOCK_NUM,12);
   WRITELN('Drive #',DRIVE_NUM,' is now clean')
END;  { ACTIVATE_DISK }

BEGIN  { CLEANHEADS }
   ALL_CLEAN:= FALSE;
   GET_DRIVE(DRIVE);
   WHILE NOT ALL_CLEAN DO BEGIN
      GET_TRACK(TRACK);
      ACTIVATE_DISK(DRIVE, TRACK);
      GET_DRIVE(DRIVE)
     END;
   WRITELN;
   WRITELN('Do Not Forget to write todays date on the Cleaning Diskette');
   WRITELN;
   WRITE('press "ENTER" to continue');
   READLN(CH)          { stop the program so user can read the messages }
END.  { CLEANHEADS }
```

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.

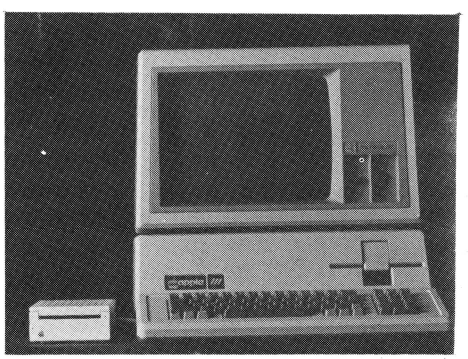
UniDisk ///.5 800K Micro-Floppy

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

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 (415) 383-0203

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 M2R 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

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 /// Owners & Users Group International
c/o Maj. H. Joseph Dobrowlski
AF SOUTH/JLD Box 149
FPO New York 09524

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
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
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-Th	10am-10pm	Pacific	GE
Kelly C. McGrew	WA	(206) 943-8533	Su-Sa	6-10pm	Pacific	DB, GR, SS, PR, MD, CT

Reading the Mail:

Three Questions

Quick Catalog

Dear Sir:

Re Quick File ///, could you please tell me how to obtain a hard-copy of a directory of files on a disk?

Also, do you have a Service Manual including schematics and printed circuit layouts? I do not want the User's Manual, only the Service Manual.

Re ON THREE magazine subscriptions, we paid the subscription price of forty dollars ($40.00) for twelve issues in January 1984, but received only two magazines. Volume 2, Number 1 was received in March 1984 and Volume 2, Number 2 was received in November 1984. Has there been other issues? If so, could you send us the balance of the subscription? We have not received any issues for 1985, though the subscription was paid for. Could we have these issues as well?

Thank you in advance.

Keith Muddle
Chatham, Ont., Canada

P.S.: We would accept Volume 1 as back issues for 1984.

Keith, we're glad you wrote, because it gives us the opportunity to explain again in print about ON THREE subscriptions. You are correct in that ON THREE was not published during 1985 and somewhat haphazardly prior to that time. It is the old problem of perfecting the 25-hour day and one individual spreading himself too thin.

By the time you read this, you will have already received the January 1986 issue and will will be aware that we have hired Val Golding as a full-time editor, which will insure regular monthly publication of ON THREE. All subscriptions have been set up on a 12-issue basis rather than one year, thus you will receive through October 1986 at which time your subscription will be up for renewal.

Back issues are available at $5 each and some are in short supply. However watch elsewhere in this issue for a money-saving special to bring you and other readers up to date.

We have our own copy of a Service Manual for the Apple /// but we do not have extra copies. It may be possible that we could answer a specific question for you from it. Otherwise try finding an Apple dealer with a service department that has been in business for three or four years. It is possible you might be able to see their copy of the manual and perhaps they might even let you photocopy it.

Quick File /// does not have the capability to output a directory to a printer. However, the Quick File disk is catalogable from BASIC, so you may get a hard copy of the directory by entering the following from the keyboard:

```
OPEN #1, .PRINTER [return]
OUTPUT #1 [return]
CAT [return]
```

You may also catalog a Quick File directory by booting on the System Utilities disk, following the prompts and selecting the CATALOG option.

Upgrading Upgrades

Dear Bob:

Enclosed is my Calendar Pak disk which I understand you now have the ability to modify so it will work with the 512K upgrade. Would you be kind enough to provide me with this adjustment. I'm delighted to hear that Tim Gill is finally putting the finishing touches to the upgrade of Word Juggler (and Lexicheck) to work with the Apple /// configuration.

I'll be looking forward to finally having it available for the script writing I do. Also I thought you might be pleased to hear that, based on the fact that Word Juggler was at last going to be available [in 512K version], I decided to really commit to the upgrade and try to use it in some concrete way. Along with the upgrade on /// E-Z Pieces, I also got the .RAM disk up and running. I hadn't really seen much purpose in it until I started to use it. Now it's a different story.

Its use in conjunction with the standard 256K Word Juggler program already shows extraordinary promise. Perhaps I'll do an article for you if you like showing how advanced users of this program can employ it in writing such things as screenplays and other complex format and writing chores. *Yay! . . .ed.* Basically it employs .RAM as a kind of advanced "Power Keys" tool— allowing the "load command" in Word Juggler to insert into a document a variety of format changes as well as characters' names, address labels, or any other material using single letter codes as file names loaded into .RAM.

Anyway, I'm sorry if I had to be so difficult over what I felt was misleading advertising advertising on the part of ON THREE, but my concern was for the future of your own company as well as for the frustrations of your subscribers who had bought the 512K upgrade believing it was ready to work with Word Juggler. I've seen enough good products come out of your shop, I didn't want to see you messed up by some kind of consumer outcry.

We've also been using a very advanced timing program we developed that can be directly loaded as a template into most spreadsheet programs such as Advanced VisiCalc and /// E-Z Pieces. It's been used as the master guide for all of our timings on such television specials as NBC's coverage of the Macy's Thanksgiving Day Parade, The [Pasadena] Tournament of Roses, and the National Emmy Awards.

It currently allows for the insertion of up to 200 different item labels on a rundown with their individual timings. The program automatically provides the operator with an accumulated total of what each item has added to the timing. In addition, one can also set an "OFF AIR" time. Let's say 11:28:47 am. With this inserted the program will, simultaneously with its forward addition, also compute a running backtime for every item in the rundown. This allows a director or others to check at any point in the rundown to see what the actual time should be at that moment in order to have the show end on time. It's a critical factor in such areas as radio and television live production.

I suppose there may be a number of other uses as well.

I must say that not having to do this kind of complicated computations by hand is a real lifesaver for those of us who have used this program over the years. Anyway, It might be something that should have a wider distribution. I'd be curious as to your opinion.

Barry Downes
New York, NY

Thanks for your long and informative letter, Barry; it provides a great deal of food for thought for our readers. Long before you read this, you will have received your Calendar Pak upgrade for 512K. This is just one of several upgrades that ON THREE provides on a cost-only basis, or for return postage only. Since ON THREE is the only supplier of an Apple /// 512K upgrade, it is our obligation, in addition to being an Apple /// support group, to furnish our customers and readers with the very latest in upgrades and related information, an obligation we take very seriously.

Based on your correspondence, there is also a need to address what you felt was misleading advertising on the part of ON THREE. Certainly we don't believe that you—or any of our readers and customers—felt there was an intent to mislead, but sometimes in the preparation of ad copy, pertinent provisos may occasionally be overlooked. In the case of Word Juggler's compatibility with the 512K upgrade, what we intended to imply, but perhaps did not fully succeed in, was that the program would work *with the upgrade but that the additional memory was not utilized by Word Juggler. Of course, the whole subject becomes moot now that there is a specific Word Juggler update that uses a good portion of the 512K memory. Because of the way that Word Juggler utilizes memory, it can never fill the entire amount of memory available. However, the availability of .RAM more than makes up for this deficiency. In terms of the actual number of lines of text available in Word Juggler, the 512K version allows about 3600, while the maximum in the 256K version is 1536.*

We would definitely be interested in sharing with our readers how you have used .RAM in advanced applications with Word Juggler. As to your timing program that can be loaded into a spreadsheet template, we think that it too would be of interest, but we leave the decision up to our readers. How about it? Is there enough interest out there to print an article on this unique application? Remember, your word is our command!

On TIME Needs Current Kernel

Gentlemen:

I need advice regarding the ON TIME package I purchased from you last December. I had no problems installing it on my Applewriter /// disk but delayed trying to use it with my Business Basic until I had made my mind up which Business Basic utility to do without (since there was no space available for the ON TIME drivers until I had deleted something).

So I have just tried it with Business Basic without any success. Installed both the .ONTIME and .CONSOLE with no apparent results—no clock or stopwatch display—nothing. Followed your instructions to the letter and repeated the excercise several times—still nothing. Help!

Also, have ordered Draw ON /// and am planning to get Pascal. Do I need special instructions for them also for ON TIME?

Henry T. Abstein, Jr.
Roseburg, OR

The key to your problem actually involves the versions of SOS used with AppleWriter and Business Basic. Earlier versions of SOS did not handle interrupts correctly and a clock, by its very nature, constantly deals with interrupts. Fortunately, the solution is as easy as writing this letter. Since AppleWriter works correctly with ON TIME, just copy the file SOS.KERNEL to any other disk with which you wish to use ON TIME. (Do make a back-up beforehand, just in case.) Your AppleWriter disk obviously has a copy of SOS version 1.3, the latest, on it.

Incidentally, the combination of Selector /// and a UniDisk ///.5 drive may solve your problem of not having enough space for your drivers on one disk.

Juggling Printer Commands

Dear Val:

I would like to share one trick with my fellow Apple /// buffs. Referring to suggestion #1 in Arthur Schumer's review of Word Juggler, there is indeed a way to use different pitches for printers with infinitely variable horizontal motion index.

Simply select a printer filter which does not support pitch commands for your printer, but which supports the other essentials such as line feed, carriage return, form feed, etc. You can then use the PRINTER CONTROL command to enter the appropriate string to print in any pitch you want.

My first experience with Word Juggler was with a Brother HR1 printer which did not fit into any of the filters. With the Brother, I quickly learned how to set the pitch by sending a control string. I presently use a Fujitsu DPL 24 which uses the DIABLO/XEROX/QUME/NEC filter. If I wish to use a pitch other than the standards, I simply change to another filter. When doing this, depending on the printer and the filter in use, it may not be possible to use other enhancements such as underline or bold print.

Now for a sour note. In May, I purchased Omnis 3 from Blyth Software in San Mateo, CA. There were some problems in getting the software to work with Catalyst. Numerous phone calls to the company produced mostly conjecture and little help. Thanks to *Bob Consorti*, who had no obligation whatsoever to help in this matter, I was finally able to configure the software so that it would come up and quit properly in the Catalyst environment.

Today, I received an unsigned letter from Blyth Software announcing their new "improved" customer service and support policy. After the initial 30-day warranty period, their customers may purchase ongoing customer support. The cost is $150 for 90 days or $500 for a whole year. Those who presently own the software can obtain support by telephone though January 31, 1986 at no charge. I would hope that, for the price they intend to charge, the quality of support would be better than that which I received after my purchase. To me, $500 per year seems totally unreasonable.

Again, it is great to see the magazine reappear. It is also a pleasure to do business with someone who is interested in providing a line of products for us /// diehards, and who is not afraid to spend a little time to help us through some of the rough spots. I enjoy my Trustor 30 and 512K machine very much, and look forward to acquiring some of your software products in the near future. (Especially that Desktop Manager—that is what I have been needing for a long time). Keep up the great work.

John R. Sollman
Silverton, OR

Page 25

Missing

Page 26

Missing

Page 27

Missing

Page 28

Missing

An Integrated Software Package:

For Free!

john lomartire

So-called integrated software packages are the latest rage. Examples would be *Lotus 1-2-3* and *Symphony*[1], */// E-Z Pieces*[2], *The Incredible Jack* and *Jack2*[3]. They are of interest because of these advantages:

A. By booting a disk only once, you get access to a variety of software packages. Ordinarily each program in the package would require a separate boot.

B. Generally one set of commands apply to, and are interpreted in the same way, by all programs in the integrated package. With individual programs, each one has its own set of commands to learn and they may or may not be similar to commands found in other programs.

C. Information, both text and data, is transferable from one program to another within the integrated package group.

Integrated packages differ with respect to the programs they contain, cost, ease of use, comprehensibility of instructions, and efficiency/rapidity of operation. Almost all, if not all, integrated packages come with word processor and spreadsheet programs coupled to other programs. Alternatively, programs such as *ON THREE's ONTIME Desktop Manager* and *Selector ///*, which run in the "background", facilitate, particularly with a large capacity disk, the utility and feel of true integrated packages.

This article describes an integrated word processing/spreadsheet program that is available to anyone who owns a spreadsheet program. Although descriptions, commands, references, etc. noted here apply to *Advanced VisiCalc*[4], this same approach can be used by substituting equivalent commands for other spreadsheets.

1. Trademarks of Lotus Development Corp.
2. Trademark of Haba Systems, Inc.
3. Trademarks of Business Solutions, Inc.
4. Trademark of Visicorp.

No discussion of spreadsheeting, per se, is to be found in what follows. Instead, it is assumed that a user is seeking a means of incorporating spreadsheet information within a word processor document. Obviously there are ways of doing this that involve re-booting each program, so that in some cases, a number of reboots may occur before the final document is assembled. How can this be done without successive rebooting, and using only the spreadsheet program?

QUARTERLY REPORT FOR INLAND DIVISIONS

(All figures in $M)

	JAN.	FEB.	MAR.	TOTALS
Division A				
Sales Vol.	75	82	84	241
Prod. Costs	58	62	63	183
Gross Profit	17	20	21	58
Division B				
Sales Vol.	66	70	72	208
Prod. Costs	55	55	56	166
Gross Profit	11	15	16	42
Division C				
Sales Vol.	81	53	66	200
Prod. Costs	65	48	60	173
Gross Profit	16	5	6	27
Division D				
Sales Vol.	32	35	40	107
Prod. Costs	27	30	33	90
Gross Profit	5	5	7	17

Figure 1. Sample of Tabular Data for a Quarterly Report

Obviously, in making up any such report one would take full advantage of all the features of a spreadsheet, such as summations, percentages, formatting, etc., to simplify the task. When the table was complete, it would then be saved AS A SPREADSHEET FILE. With Visicalc, this means saving it with the /SS command.

Next would come the writing of the text portion of the report that would go along with the tabular data. Instead of quitting the spreadsheet program and booting a word processing program, proceed as follows:

After clearing the screen of the table display, CHANGE THE COLUMN WIDTH OF COLUMN A TO THE NUMBER OF CHARACTERS DESIRED ON A LINE OF TEXT. For example, change column A to a width of 80 (allowing for 80 characters in a line). Now, while still in the spreadsheet program, start typing the desired text. Remember the applicable commands. There are some differences between this procedure and a full-fledged word processing approach, but it will be satisfactory for many applications. For example, there is no word wrap-around but on the other hand, since you are typing in an highlighted line, you can easily tell when you are approaching the end of the line. This proves to be advantageous for hyphenation purposes. Corrections made before the end of the line (before RETURN is pressed) are easily made with the ESCAPE key. Corrections to an accepted line (after RETURN has been pressed) that do not lengthen the line beyond its specified length are also easily made with the line editting feature.

Perhaps the best way to present this concept is to use a practical example and follow it through.

Situation:

You are the director of a company and you oversee four divisions. At the end of each quarter, you forward a report detailing the sales volumes, costs, gross profits and any other pertinent information to your main office. These data are transmitted in a combination report consisting of text and tabular information. Obviously, a word processor could provide the text portion of the report, and a spreadsheet could produce the required tables, but to do both of these tasks using only the spreadsheet program you would proceed as follows:

First, make up the desired spreadsheet table as you would ordinarily. It might look like figure 1.

Date: May 17, 1984

Subject: Preliminary Quarterly Report on Division A, B, C, and D

To: Mr. J. J. Forester

From: E. D. Anderson

This report is for the first quarter of 1984 and is summarized in Table 1. Although the figures are the most accurate available at this time, there is a need for amplification of some of the entries if they are to be placed in proper perspective.

The overall gross profit of $144,000 is gratifying since, in view of the unexpected difficulties that were encountered, it was expected to be much less.

QUARTERLY REPORT FOR INLAND DIVISIONS
(All figures in $M)

	JAN.	FEB.	MAR.	TOTALS
Division A				
Sales Volume	75	82	84	241
Production Costs	58	62	63	183
Gross Profit	17	20	21	58
Division B				
Sales Volume	66	70	72	208
Production Costs	55	55	56	166
Gross Profit	11	15	16	42
Division C				
Sales Volume	81	53	66	200
Production Costs	65	48	60	173
Gross Profit	16	5	6	27
Division D				
Sales Volume	32	35	40	107
Production Costs	27	30	33	90
Gross Profit	5	5	7	17

Divisions A and B continue to operate as they have for the past several years even though the economy has not been as strong as in earlier times. This has been accomplished by extra emphasis on productivity with all workers contributing heavily to a successful operation.

Division C has been struggling to overcome the effects of the large fire in Warehouse #12 and by the end of this first quarter normal output levels were again being observed.

Division D, our brand new venture, is developing sales more rapidly than expected and shows every sign of being our best unit in the near future.

Figure 2. Sample "report" showing combined text and tables

Obviously, in making up any such report one would take full advantage of all the features of a spreadsheet, such as summations, percentages, formatting, etc., to simplify the task. When the table was complete, it would then be saved *as a spreadsheet file*. With VisiCalc, this means saving it with the /SS command.

Next would come the writing of the text portion of the report that would go along with the tabular data. Instead of quitting the spreadsheet program and booting a word processor, proceed as follows:

After clearing the screen of the table display, *change the column width of column A to the number of characters desired on a line of text*. For example, change column A to a width of 80 (allowing for 80 characters in a line). Now, while still in the spreadsheet program, start typing the desired text. Remember the applicable commands. There are some differences between this procedure and a full-fledged word processing approach, but it will be satisfactory for many applications. For example, there is no word wrap-around but on the other hand, since you are typing in a highlighted line, you can easily tell when you are approaching the end of the line. This proves to be advantageous for purposes of hyphenation. Corrections made before the end of the line (before [return] is pressed) are easily made with the escape key. Corrections to an accepted line (after [return] has been pressed) that do not lengthen the line beyond its specified length are also easily made with the line editing feature.

When all this has been done, you have a spreadsheet full of text and another spreadsheet of tabulated information.

Combine these two *during printing*. First print all of the text up to where the table is to be inserted. Then print the table and finally finish up with the rest of the text.

Figure 2 shows a "quarterly report" that has been produced with this technique, entirely with VisiCalc. It is admittedly very simple and short because its only purpose is to demonstrate a concept, so it should not be judged beyond this.

The table of figure 2, starting with the line showing the monthly headings and ending with two blank rows beyond the last entry line, was prepared using standard Advanced VisiCalc commands. The "TOTALS" and "Gross Profit" columns contained formulas to calculate the corresponding cell entries, after "Sales Volume" and "Production Costs" have been entered. When completed, this information was saved to disk with the /SS command as ".D1/TABLE" since the receiving disk was in drive #1. Each of the four columns on the right were nine characters wide for a total width of 36. Since the table was to fit an 80-column format, the leftmost column was set at 44 (i.e., 80—36). Although this was correct, when it was printed it was seen that a slightly narrower table would look better, so the leftmost column was reduced from 44 characters to 40.

All of the text in the report was a straightforward entry into lines of 80-character column width, with blank row insertion as desired. Note the hyphenation in several of the lines. Most of the lines in figure 2 start at the left margin, but the chart headings are centered on the page through use of the label-centering option in Advanced VisiCalc. In a similar fashion, any entry could be positioned at any horizontal starting point by using existing VisiCalc commands and options. The text portion was saved to disk by /SS ".D1/TEXT".

Printing involved sending to the printer all text up to the second line of the table heading - - - "All figures in $M" - - - plus two blank rows. Then the screen was cleared and the TABLE file recalled and printed, making sure that there were two blank rows beyond the last row of figures. The screen was cleared again, the TEXT file recalled and the lower half, starting with "Divisions A and B continue ..." was printed. By totaling the lines in the TABLE and TEXT files, it was possible to keep track of whether a page break was needed. (In this case, it was not.)

Although this seems to be awkward upon reading, in actual practice and with a little experience it all works very well.

Fully integrated programs like to describe their operation as having all your work sheets scattered on a desk top. As you decide to work on something, you bring it forward from the rest of the material on the desk top, replacing the other item you may have been working on before. A large number of items are kept on top of the desk for quick access. In this same analogy, being able to do everything while seated at a desk represents the single boot feature of these programs. If in the analogy it was necessary to leave the desk to do some other operation, then this would correspond to the re-booting of another program. What has been described here is similar to working at a desk with two items in the desk drawer instead of on top of the desk. When you want to do word processing, you "take it out of the drawer" and put the spreadsheet "in the drawer", and vice-versa. You never "leave the desk" (that is, no disk re-booting) for these two systems, so in this respect they are integrated.

It is obvious that the full power of the spreadsheet program is available, since it is the boot program. As a word processor, it is equally obvious that many of the features found in dedicated word processors will not be present, but it works well enough to have merit. Lines can be edited, blocks of lines can be moved or copied to another section of the document, formatting is possible, page breaks can be identified.

USING THIS INTEGRATED PROGRAM APPROACH TO PRODUCE A DOCUMENT IN MULTIPLE COLUMNS

Now here is where we demonstrate a capability that is not usually available with most word processors —— the formatting of text into multiple columns. There are a few fancy word processors that will do this, but not many.

The ability to format in rows might be useful for newletter preparation, for example.

With this approach, you merely decide on the width for each column. In this example, a total width of 80 characters was to be assigned to three columns, with a small spacer column between the text. Therefore a column width of 24 characters was selected, with 4 character spacers. In other words, columns A, C, and E were set at 24 and columns B and D were set at 4.

All of the text was then entered in column A. At the end, the row number provided a good guide to the equal division of text into the three portions.

Before doing this, however, let's try also including a chart. It is a simple chart, just to show that it can be done with this technique.

First, the chart was made up using three columns of 14, 4, and 4 character widths. The rightmost column (C) was calculated from the second column and the overall total. This chart took up 12 lines, and it was saved as TEXT2 on a disk.

If it was desired to enter the chart here, then just leave 12 blank rows

Part Ident.	No.	%
A1256	12	8
A5611	16	11
B06997	22	15
DE254	9	6
X48997	28	19
A5487	19	13
EL548	30	20
X9953	14	9
Totals	150	

plus two more for separation, and continue with text entry. The chart will be printed after all three text lines are on paper. This will require extra effort in positioning of the paper if the chart is to fall in the blank spot set aside for it. This is, again, a simple matter to do and the procedure is not as complex as a description. When formatting the columns, be sure to note the exact position, vertically and horizontally, for the chart placement. With this knowledge the chart can be positioned to print in the exact spot.

This figure was prepared in the fashion described to illustrate the concept.

It is also possible to include a headline that travels across all columns as shown here. It is fitted last, after the columns have been positioned. All text is moved down and the headline put across the top.

Figure 3

Can this proposed package do something better than conventional word processing programs? For the answer to this, continue by reading the text contained in figure 3. When you have read figure 3, return to this point in the text or, as they say in a BASIC program, "GOSUB FIGURE 3".

Now that you are back from reading figure 3 (you did read it, didn't you?), one final example to show the power of this approach. Figure 4 shows how a box of information (text, data, etc.) can be fitted into any segment of a page of text, including printing across several columns. In this case, text was taken from an *ON THREE* ad merely to illustrate the concept.

In summary, a spreadsheet program can be adapted to serve as a pretty fair *integrated* word processor. Although there are certain limitations to the approach of integrating these two operations, they are not sufficient to inhibit the adoption of the concept. It does take a little getting used to, and a knowledge of spreadsheet commands is essential, but it provides an opportunity to go back and forth between these two important programs without re-booting and without changing command structures. In addition, as shown in figures 3 and 4, it is possible to do things with this approach that are not easily, if at all, attainable with regular word processing alone.

The examples shown here have purposely not been detailed step by step because the aim was to present a concept that can be tailored to individual needs. *Experiment*. It is fun to see what you can do, and this article has only scratched the surface. It is not hard to develop expertise in the use of the procedure; it only takes a little time to become very familiar with your spreadsheet. When you have a need for some special arrangement of text, tables, etc., consider using this method. If your document is of standard text format without difficult placement of tables, and you do not mind booting disks more than once, use any regular word processing program.

AN EXAMPLE OF "NESTING" OF INFORMATION

****

You may recognize this text as a portion of the "Apple Sauce" page written by Val J. Golding in the January 1986 issue of ON THREE. This text was placed around a table that appeared on page 2 of the same issue. The main purpose here was to illustrate a particular format that was attainable using the techniques discussed in this article.

ONWARD THREE

Welcome back to the world of ON THREE. We'd like to thank those of you who have stuck around and stayed with ON THREE through its various stages of publication, non-publication. Alas, as we found out some years ago, there isn't always time to do what you want, and when you are operating a business, the business usually comes first. Bob Consorti will vouch for that. Our role then is to assist Bob and guide the magazine. Our introduction to the /// was via A.P.P.L.E. co-founder Bob Huelsdonk who took delivery of one of the very first ///'s delivered in Seattle. Bob was, and still is, very proud of his /// and he always delighted in showing it off to us and to anyone else within earshot. It is truly a powerful machine, every bit as innovative as Woz's original Apple One back in 1976. We have followed the (mis-)adventures of the /// ever since and cried a bit when we heard that the company that gave birth to such a fine machine also attempted to snuff out its life. But you owners out there --- our friends--- knew better. That's why ON THREE is still here. Currently there are some 100,000 plus machines out there. How many of you remember that, although the design guidelines provided for far more, that the first Apple /// was only 96K? Now, through ON THREE you can upgrade to 512K. Not too shabby for a five year old machine, huh? As we followed the progress of the ///, we were surprised to learn that there are still a few new products being produced for the ///, and occasional upgrades as well. One new product we've just recently been made aware of is a real hush-hush earth-shaker. We'd like to tell you more, but we also like our job. Nevertheless, you'll read about it here first, within the pages of ON THREE in our "Apple Sauce" column. Each month we'll fill you in on new products when we find 'em and reacquaint you with some of the existing ones. In addition, we'll take the time to point out some of the stories and features in the current issue and explain how you can benefit from them. We'd also like to make our column a forum for story ideas: the types of articles and programs you most want to see in ON THREE. But be careful what you say, we may call on you to supply the material!

::

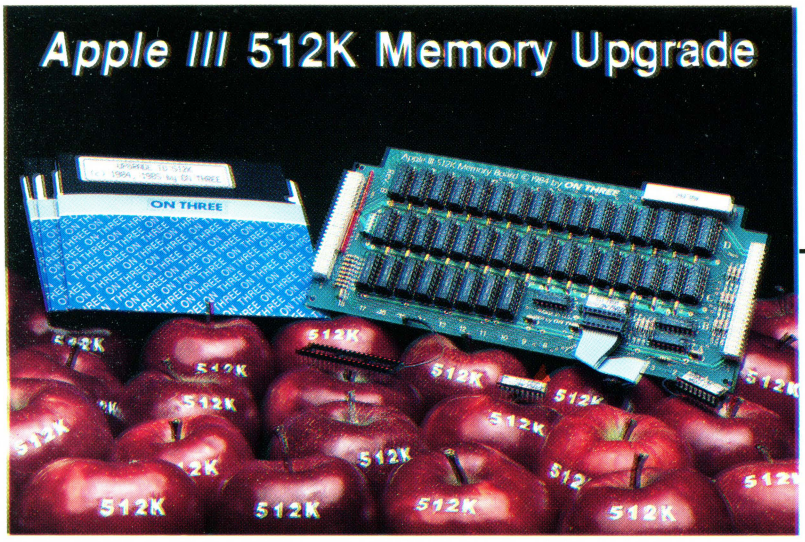
Other Products from ON THREE ...

This flyer is much too small to show you all of our software and hardware products for the Apple /// and Apple /// Plus. Please call or write for complete product information and pricing.

Used 256K Apple ///'s with Monitor ///	$1050.00	$50.00
Apple /// Bus. Basic, v. 1.23	$99.00	$5.00
Apple /// Fortran 77	$99.00	$5.00
256K Memory Upgrade	$250.00	$10.00
Dust Cover for Apple ///, Monitor ///	$11.95	$2.00
I LOVE MY APPLE /// caps	$5.98	$2.00

::

Figure 4. Nesting of information at a particular location on the sheet.



+

= 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

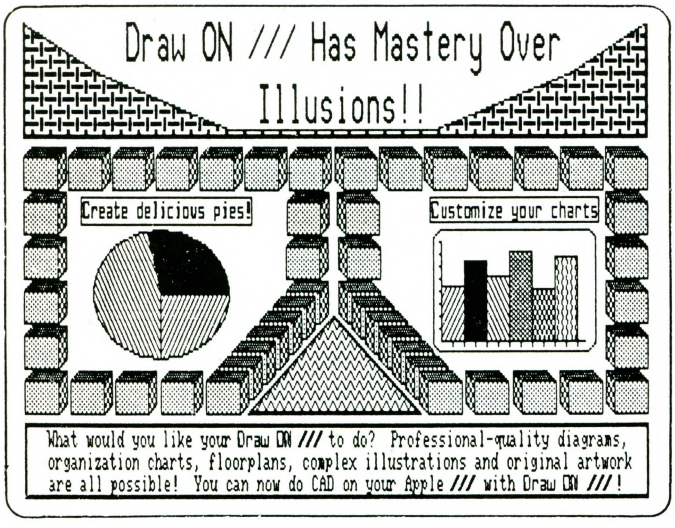
Calif. residents add 6 % sales tax (products only) We accept Visa, Mastercard, American Express†
†*3% surcharge on American Express orders*

ON THREE Presents . . .

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

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

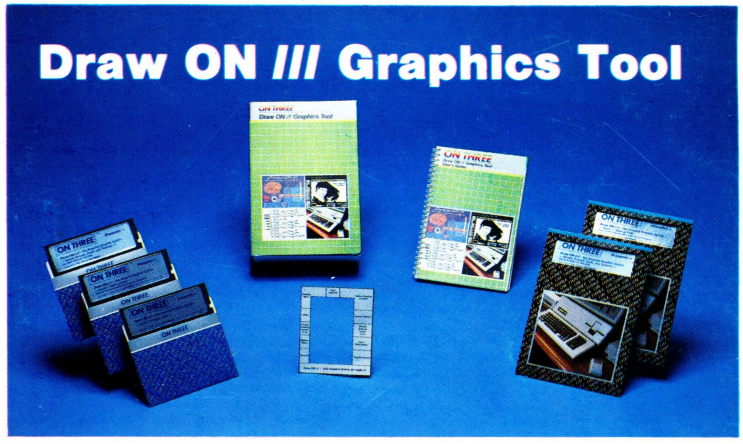
Draw ON gives an individual the power of a graphic arts studio. Use it in creating charts, preparation of slides and tables for presentation, and letterhead design. With ***Draw ON*** you can make changes to the dull graphs



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

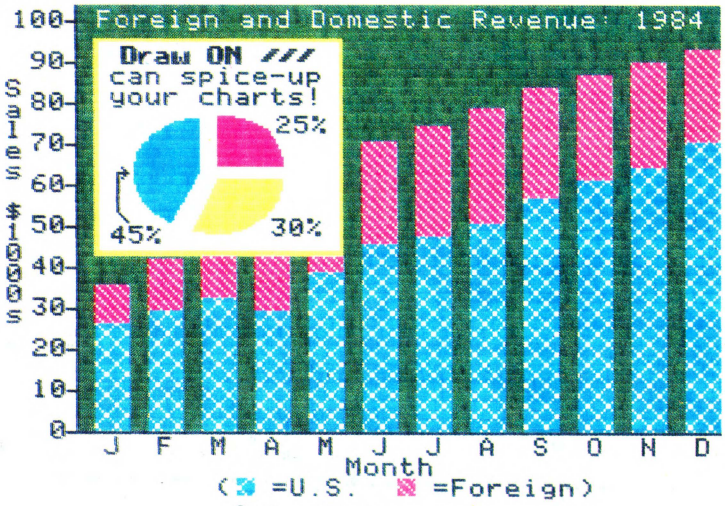
To control ***Draw ON*** either a joystick, mouse or the keyboard is used. Since there are no mice available for the ***Apple ///, ON THREE*** has enabled ***Draw ON*** to use the ***Apple //e*** mouse and interface card. If you would like the ease of use that the mouse provides, purchase an ***Apple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

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



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

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



Optional Equipment

Cursor /// Joystick, Apple Mouse //e or *Apple // Graphics Tablet* (*Graphics Tablet* version costs $50 extra), *RGB Color Monitor, Dot Matrix Printer*

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

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