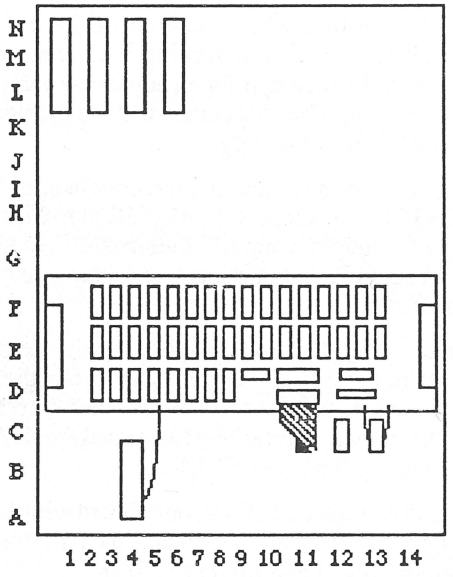
ON THREE®

The Magazine For Apple /// Owners and Users

Volume 5, No. 9/10 September/October 1988

Expanded *Double* Issue



A Quick Peek At Upgrading To 512K!

In This Issue...

ON THREE News & Views

From The Desk of Bob Consorti

Another fall season in the wonderful world of the Apple /// is upon us. Being the last developer to create new products for the ///, we've been asked repeatedly why we do it. The answer is quite simple. The /// for all its potential greatness really never took off as some of the other machines did. Most of the newer software available for the more recent computers was never written for the Apple ///.

Over the last six years we've dedicated ourselves to bringing you the best quality software and hardware possible for the Apple ///. With our memory expansions, hard disk drives, desk accessories and other powerful utility and productivity tools we've been helping keep the Apple /// up to date with the newer machines at a fraction of their cost.

In that spirit I'm pleased to announce five new products and a fall clearance sale with products up to 50% off our regular low prices. This month we're introducing a low cost color printer for the ///, a sideways printing utility for spreadsheets, low cost 512K memory expansion boards and a new game for the /// - BattleFleet, just like BattleShip but done in graphics!

I encourage you to look over this flyer and pick out one of our great new products or one of our older products that is on sale. We have something for everyone, so please phone or mail in your order today.

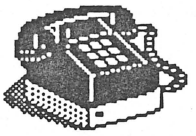
Below I've listed some of the products that are currently on sale. Make sure you order early as our fall clearance sale ends on October 31, 1988 and this sale may never be repeated!

Fantastic Sale Prices!

Product	*Regularly*	*Sale Price*
Draw ON ///	$179	$99
StemSpeller ///	$69.95	$59.95
StemWriter	$99	$79.95
Comm. Manager	$49.95	$39.95
Graphics Manager	$39.95	$29.95
Lazarus ///	$49.95	$39.95
Selector ///	$99	$69.95
The Desktop Mgr.	$129	$69
TDM + any 2 accs.	$200+	$99
Disk Manager	$44.95	$29.95
Macro Manager	$44.95	$29.95
Super Accs. #1	$39.95	$29.95
Super Accs. #2	$24.95	$19.95
Super Accs. #3	$29.95	$24.95
Apple /// w 256K	$599	$499
Apple /// w 512K	$999	$899
Sider 20 Hard Disk	$949	$899
Titan ///+//e	$399	$349
1200 Baud Modem	$159.95	$129.95
2400 Baud Modem	$259.95	$229.95

(All Modems Now Include Comm. Manager)

Special Fall Clearance Sale!



Phone in an
ORDER TODAY!

Orderline: (800) 443-8877

In Washington: (206) 823-0516

For Support/Questions: (818) 701-1355

New! - Low Cost 512K Upgrades

You might be asking what's new about the 512K upgrades. Well, in view of the ram chip price increases of the last year we've decided to offer a couple new varieties of the 512K upgrade. The first comes with no memory, the second with 256K of memory.

You can purchase the additional memory needed by the 512K (32 4164K ram chips and 8 41256K ram chips) from a variety of vendors yourself, thus reducing the cost. We'll even supply a list of the places where you can get ram chips from.

You can order a 512K memory board with zero K of memory, take the 32 4164K ram chips from the 256K memory board in your Apple /// and then purchase 8 41256K ram chips and get a fully functional Apple /// with 512K of memory for under $300!

You can also now get a 512K memory board with 256K of memory (the 8 41256K chips) and take the ram from your 256K memory board and fill up the 512K.

All of the memory boards come complete with instructions for installation and use, diagnostic and update programs - Everything you need to turn your Apple /// into a 512K powerhouse.

The 0K (zero K) 512K is priced at only $159.95 and the 256K 512K costs $299.95. Add $10 for shipping, handling and insurance for each memory board purchased.

Apple /// Technical Support

Technical support is FREE on any product that you've purchased from ON THREE within the last 90 days. After that time, or on any non ON THREE product there is a modest fee for technical support. $5 for the first three minutes and $1 per minute thereafter with a $15 minimum.

The support line is open Monday through Friday 9 AM to 5 PM Pacific time. Please have your product serial number (if applicable), purchase date, your credit card number and a list of the problems you're having ready *before* calling.

Technical Support (818) 701-1355

Editor/Publisher:

Bob Consorti

Controller:

Joseph Consorti

Design Consultant:

Cathy Consorti

ON THREE - The Apple /// Magazine is published bi-monthly by ON THREE, Inc. 8920 Yolanda Avenue, Northridge, CA 91324. ISSN# 0889-6249.

For a copy of our author guidelines, please send a stamped, self-addressed envelope to the above address. The current per page rate is $12.50.

Return postage must accompany all submitted manuscripts, diskettes and drawings if you would like them returned. All manuscripts longer than one page must be accompanied by a diskette to be considered. We can take no responsibility for unsolicited materials.

All letters received by *ON THREE* will be considered as unconditionally assigned for publication and as such are subject to *ON THREE*'s right to edit and comment. Your full address will be published only when specifically requested.

Subscription Information:

U.S.,APO, FPO: $30 for one year. First Class Service: $12 additional, $42 total.

Canada, Mexico: $20 additional, $50 total.
South America, Europe, Pacific Islands, Asia, Australia, Middle-East: $44 additional, $74 total.

Back Issues are available for $5 each.

Postage for ON THREE products:

U.S.: Listed next to product price.
Canada and Mexico: Double U.S. postage.
Elsewhere: Four times the U.S. postage.

All funds must be remitted in U.S. dollars drawn on a U.S. bank. We also accept direct wire transfers and U.S. Postal money orders.

ON THREE®

The Magazine For Apple /// Owners and Users

Table of Contents

Volume 5, No. 9/10 September/October 1988

FEATURE ARTICLES

DEPARTMENTS

ON THE COVER:

A quick peek at one of our articles, upgrading to 512K using our new 512K upgrade kit.

Cover by Bob Consorti

Call Three: Hotline

The Call Three Hotline is a service whereby Apple /// users can contact fellow Apple ///er's who have expertise in a particular subject(s) pertaining to the Apple ///. These individuals are not compensated for their efforts so we all owe them a resounding **Three Cheers** for their generosity.

If you have questions, feel free to call the consultants listed below, but please observe the calling times listed. It would be a good idea to check the time zone so as to *not* wake someone up in the dead of night. If you are willing to spend a little time and help your fellow Apple ///er's, write ON THREE giving the necessary info. and we will be happy to add your name to the Hotline.

Name	Area	Telephone	Days	Hours	Zone	Subjects
Coville Woodburn	NH	(603) 863-5590	M,Tu,Th,F	7-8 PM	Eastern	BB,CT,GE,MI,QU,WP
Ken Johnson	MA	(413) 536-7502	Su-Sa	6-9 PM	Eastern	BB,PA,MD,WP,MI
Don Loosli	MI	(313) 626-3848	M-F	9-5 PM	Eastern	GE,WP,SS,DB
Harry T. Hanson	NJ	(201) 467-0712	M-F	6-9 PM	Eastern	GE,PA,BB,CT
Edward Gooding, Sr.	VA	(804) 747-8751	Su-Sa	6-9 PM	Eastern	CO,SS,PR,MD,CT
Jeff Fritz	WV	(304) 599-7010	M-Sa	8-11 PM	Eastern	BB,DB,GE,MI,SS,TC,EP
Al Johnson	FL	(904) 739-1600	M-F	9-6 PM	Eastern	GE
Paul Sanchez	FL	(305) 266-5965	Su-Sa	10-4 PM	Eastern	SS,PR,CT
R.B. Thompson	NC	(919) 787-1703	Su-Sa	10-10 PM	Eastern	BB,DB,GE,SS,WP
David Galliher	TX	(214) 462-0346	Su-Sa	5-9 PM	Central	Corvus
J. Donald Glenn	NE	(402) 291-9177	Su-Th	7-10 PM	Central	GE
Scott Weddel	NE	(402) 572-7543	Su-Sa	4-10 PM	Central	GE,TC
Jim Ferencak	IL	(312) 599-7505	M-F	10-5 PM	Central	GE,EP,DB
Paul Thomas	MS	(601) 494-4476	Su-Sa	6-10 PM	Central	GE,AC,BB,CP,DB,FI,MI,PA, PR,SO,SS,TC,EP,WP
Earl T. Brelje	MN	(612) 455-6405	M-F	4-9 PM	Central	CT,DB,WP,GE,QF,O3
Ron Maupin	TX	(512) 280-0144	Su-Sa	8 AM-10 PM	Central	AL,CO,CT,EP,MD,PA,QU, SS,TC,EP,WP
Terri Wiles	CO	(303) 850-7472	Su-Sa	10-6 PM	Mountain	PA
William Prince	OR	(503) 254-6465	M-F	9-4 PM	Pacific	GR,TC,Corvus
Karl La Rue	WA	(509) 582-6459	F-Su	6-10 PM	Pacific	MD,GE,EP,WP,TC,SS,CP
Pat Holwagner	CA	(415) 433-2323	M-F	10-6 PM	Pacific	GE,SS,WP,CT,DB,SO,AE,EP
Vincent F. Latona	CA	(818) 703-0330	M-F	9-5 PM	Pacific	GE,WP,BB,SS,AE
Dennis R. Cohen	CA	(818) 956-8559	Su M-F 7-9 PM	10-10 PM Sat Noon-6 PM	Pacific	GE,PA,MU,WP,DB,SO
Kelly C. McGrew	WA	(206) 943-8533	Su-M,Th-Sa	7-9 PM	Pacific	DB,GR,SS,PR,MD,CT
Larry E Kalland	AK	(907) 272-4968	Su-Sa	12 Noon-11 PM	Alaska	AC,CT,GE,SS,EP,WP
H. Van der Straeten	Belgium	(015) 516561	Su-Sa	7-10 PM	Europe	BB,CT,DB,GE,PA,PR,SS
Arnaud Trache	France	(21) 030421	Sa-Su	10-7 PM	Europe	AC,BB,DB,AE,FI,GR,MD,PR,SS, TC,EP,OT
J. Woretshofer	Netherlands	(043) 611704	Su-Sa	9-12 AM	Europe	CT,FO,GE,PA,QU,SO,WP,AE,EP
Salvador Garcia	Spain	(91) 234 5068	Su-Sa	7-10 PM	Europe	BB,GE,PA,MD,CT

Subject Table

SUBJECT	CODE	SUBJECT	CODE	SUBJECT	CODE	SUBJECT	CODE
Accounting	AC	Agriculture	AG	Assembly Lang.	AL	Business Basic	BB
Catalyst	CT	Cobol	CO	CP/M	CP	Data Base	DB
Education	ED	Emulation	AE	Financial	FI	Fortran	FO
General	GE	Graphics	GR	Micro-Sci	MI	Modems	MD
Modula-2	MU	Omnis 3	O3	Pascal	PA	ProFile	PR
Quark	QU	Quick File	QF	SOS	SO	Spreadsheets	SS
Telecomm.	TC	/// E-Z Pieces	EP	Word Proc.	WP	ON THREE	OT

Apple Sauce

by Bob Consorti

Something Bad...A Good Deal Of Good

Well folks, there's a little bad news a a great deal of good news. I'll get the bad news over with first. With less than 500 subscribers left (from a high of 6,000 a couple years ago), we simply can't continue publishing the magazine on a monthly basis. We are therefore going to publish on a bi-monthly basis from this point on. The next issue you will receive is the November-December double issue at the beginning of November.

As a consequence we're cutting back our advertising so it occupies just the center two pages in the magazine and the cover space. This way we can include much more in each issue so you'll get more out of each copy of ON THREE. We don't want to stop the magazine as many might see that as the death knell of the ///. We plan on continuing to produce ON THREE Magazine and creating new products for the /// well into the foreseeable future. Our bi-monthly status will give us the financial relief that will let us keep on helping you.

Bad news aside, our commitment to the Apple /// is stronger than ever. Lest anyone think that might not be the case, this month we're introducing a number of significant new products for the Apple ///. I can't decide which one is more important so I'll talk about them in alphabetical order...Well sort of.

Color Printouts Anyone?

How would you like a color printer for your Apple ///? I thought you might like to get a hardcopy of the great color graphics that Draw ON /// or some of your other Apple /// graphic programs create. Well I'll bet you don't want to pay the $600+ a new ImageWriter II color printer costs. How about $500, or $400, or better yet $300. No, how about $$259 for a new Seikosha 700A Color Printer complete with parallel interface card and a special copy of the Color Graphics Manager? I think I see a few hands going up!

The Seikosha color printer combined with its parallel interface card and Color Graphics Manager software is a fantastic steal at only $259. Just another brick in the foundation of our continuing commitment to the ///.

How About Sideways Printing?

One of the other products we're releasing this month is **Side Print ///™**. How many times have you wanted to print a wide spreadsheet sideways? Well, our new Side Print /// program is just what you've been waiting for. It supports the Apple DMP, ImageWriter, Epson and OkiData series of printers and best yet is only $29.95!

Want Some Entertainment?

After a hard days work why don't you kick up your heels and try our new BattleFleet game! At only $19.95 it is a great bargain. Just like the board game BattleShip, BattleFleet lets you hunt out enemy vessels and even do a graphics screen dump at any time!

Fall Clearance Sale

Located on pages 14, 15 and the inside back and back cover is product information on our fantastic fall clearance sale. Never before have prices on almost all of our products been this low. We're running a number of combination specials so please check them out. The fall clearance sale ends October 31, 1988 so please look over the product information and place an order today!

In This Issue...

The Ranns' explore connecting the Apple /// to various printers and take a look at how long your floppy disks will last while Dr. Allan Bloom describes the new and improved version of ThreeWorks.

Yours truly shows you how to do ultra-fast text screen saves/restores from Basic with the help of a medium sized invokable module in the Business Basic Extensions column. I also take up some space this issue with a look at upgrading to 512K with our new 512K upgrade kits. Finally I talk about using your modem to dial up an information service such as CompuServe.

I think you'll agree that we've really packed a good deal of information into this issue. I hope you like it and come to enjoy our new expanded bi-monthly schedule. ///

Apple /// User Groups

If you want to meet other Apple /// owners and exchange ideas, tips and hints about the Apple ///, contact one of the user groups listed below. They're all willing and able to help new and old Apple /// users with the trials and tribulations of owning an Apple ///.

If you recently formed an Apple /// group or know of one that is not listed here, please contact ON THREE and we will include the updated information in this section.

California

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

Orange County
Apple /// User Group
22501 Eloise Avenue
El Toro, CA 92630

L.A.-South Bay
Apple /// User Group
P.O. Box 432
Redondo Beach, CA 90277
(213) 316-7738

Apple /// Users of Northern CA
P.O. Box 1528
Mill Valley, CA 94942

International Apple Core
(Apple /// SIG)
908 George Street
Santa Clara, CA 95054
(408) 727-7652

Canada

Apples British Columbia Computer
Society (Apple /// SIG)
P.O. Box 80569
Burnaby, B.C.
Canada V5H 3X9
(416) 839-7779

The Astronic Club
1453 Highbush Trail
Pickering, Ontario
Canada L1V 1N6
(416) 839-7779

Colorado

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

Connecticut

Apple /// Society of So Connecticut
34 Burr School Road
Westport, CT 06880
(203) 226-4198

Florida

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

Georgia

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

Illinois

TAU c/o Lavona Rann
1113 Wheaton Oaks Drive
Wheaton, IL 60187
(312) 665-6319

Kansas

Kansas City Apple /// Users Group
5533 Granda
Roeland Park , KS 66205
(913) 262-3355

Maine

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

Maryland

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

Minnesota

Minnesota Apple Corps User Group
P.O. Box 796
Hopkins, MN 55343

New Jersey

New 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 /// SIG
P.O. Box 1666
Fairborn, OH 45324-7666
(513) 879-5895

Oregon

Oregon Apple /// User Group
1001 S.W. 5th Ave. #2000
Portland, OR 97204
(503) 645-6789

Texas

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

River City Apple Corps
Apple /// SIG
Box 13349
Austin, TX 78711
(512) 454-9962

Houston Area Apple User Group (H.A.U.G.)
Apple /// Division
P.O. Box 610150
Houston, TX 77063
(713) 480-5690 or
(713) 974-5153

Virginia

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

Virginia (Continued)

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

Apple THREE Group Intrnl
c/o H. Joseph Dobrowolski
P.O. Box 913
Langley AFB, VA 23665
(804) 865-7520

Overseas

Apple /// Users
Belgium/Netherlands
c/o/ H.Van der Straeten
Mechelsebaan 116
2850 Keerbergen
Belgium
(015) 516561

Apple /// User Group
Belgium/Netherlands
c/o J. Woretshofer
Ganzerikweerd 22
NL-6229 TG Maastricht
The Netherlands
(043) 611704

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 /// SIG
P.O. Box 174
Watford Herts, England
WD2 6NF
0727 73390/72728

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

Apple /// User Group
c/o Canberra Acct Services
P.O. Box 42
Duffy A.C.T. 2611
Australia

The Beginning ///

by Richard and Lavona Rann

The Apple /// And Printers

The most difficult problems for beginners and experts alike, are those encountered when hooking up a new printer to an Apple ///. Years ago, when we got our first Apple ///, it was the most difficult problem we had to solve. There we were, with years of experience using and programming computers (though we admit very little in microcomputers at that time), and we could not make the printer work.

Though we hate to admit it, it took several days of trying before the printer finally worked. Part of the problem we brought upon ourselves because we bought a "nonstandard" printer. Still, although it usually doesn't have to be that hard, setting up printers for the Apple /// can be difficult and involve a lot of trial and error.

Most computers have a rather limited choice of printers or printer types that can be used with them. Many computers, especially those made by IBM or those pretending to be IBM's, require special hardware boards to provide the needed interface between printers and the computers. This situation may make it easier to install a printer, but it also limits the possible configurations.

Sometimes, it is a problem to get a board and printer to match, and for the two of them to fit the software selected. Since this approach is "hardwired" (on the board) there is very little that can be done to adjust to changes in system software and new program capabilities as they become available. The computers printing capabilities are frozen in time, once the board is selected unless a new board is purchased.

Also, in many operating systems, the software includes the detailed instructions on how to "talk to" a specific printer or board and must be modified by the developer before newer devices may be used. Many people are locked out of using new programs because they would have to change their interface card, and perhaps, both their interface card and their printer.

In theory, at least, the Apple /// is different. The /// can be adjusted by software to fit almost any new hardware peripheral situation. It was designed with great flexibility to allow people to select external devices. When it was designed and built, it had considerably more flexibility than any other microcomputer had ever had.

The /// uses driver files to match the computer and printer and allows individual programs to ignore the task of special printer problems. While this method allows for tremendous flexibility, it means that having a printer without the appropriate driver makes the printer virtually, if not totally, useless. Without a matching driver installed, a printer will not work on the ///.

When the /// was an active product of Apple, driver files written by Apple and hardware developers generally accompanied the products that required them. The problem now, is that a lot of printers and interface cards were intended only for other computers and were introduced after the /// was no longer a factor in the printer market. No Apple /// drivers were written specifically for them. Many of these will still work with the Apple /// by use of prior drivers, but some will not.

Some that can be made to work with drivers intended for other printers may not be able to access all the special features, like subscripting, or bolding that the printer and the /// can normally handle. In any event the search for the right driver can be a tedious affair of trial and error. Unfortunately, because writing a drive is such a complex form of programing it is virtually impossible to get anyone to wrote a new driver. There are very few people still programming on the /// in assembler at all.

Among that group are even fewer, if any, that would take on the task of creating a new driver "just for the fun of it." Even if a programmer could be found willing to write a new driver for the Apple ///, the cost would be prohibitive since the market would be so small. This situation

leaves the /// owner with no recourse other than using the drivers already written, and selecting printers that will work with them.

The easiest solution still available new, is the Apple ImageWriter. Apple's ImageWriter, DMP's, and older LQ printers are /// compatible and most still have instructions for the Apple /// in their manuals. Beyond that there are many other printers that will work well with the Apple ///. The major danger signal to watch for in selecting a new printer is technology that was not in existence while the /// was being sold by Apple or a special card required for use.

Obviously, these will not be very likely to have a driver written specifically for them for use on an Apple ///. If they can be made to work, at all, they will likely not be able to use all their features.

SELECTING A PRINTER

When the Apple /// was introduced there were few printers on the market. The Apple /// was specifically set up for using the Apple Silentype printer. This printer was obsolete by the time most Apple ///s were sold, so few people have ever seen one. The Silentype printer used thermal paper that came in rolls and tended to brown with age or exposure to light. This printer is handled differently than any other printer because it connected to the Apple /// through joystick port A, and the /// came from the factory set to use the Silentype.

Even thought they have been out or production for 6 or 7 years, Silentype printers are still available. They don't make bad little draft printers and are especially low priced if you don't consider that the paper is fairly expensive.

Other printers generally used on the Apple ///, fall into one of two categories depending on how they transmit data between the /// and the printer. Serial printers work much like a telephone in which the data is transmitted in a stream, one small piece at a time. A serial interface is built into the Apple /// and serial printers may be connected to an Apple /// without any additional equipment (you have to have a proper cable, but those are available).

Parallel interface printers require data transmitted in data packets. It presumes the cables contain a number of transmission lines, and requires an interface card be added to the /// to convert the serial output of the computer to parallel format. The names themselves are descriptive of just how the data is transmitted. Either in order, singly in serial one after another, or as groups arranged in bunches of data sent over a group of parallel line at the same time.

There have been lots of technical arguments over which is better. We won't enter into that argument, but will tell you that the standard on newer Apple's is Serial, and that is becoming common on other machines as well.

The serial interface requires less equipment to operate on an Apple /// than does a parallel interface. The serial interface is accessed through the built-in RS-232-C port. Since, on the Apple ///, the RS 232 port is designed as a communications port, it must either have a modem between the computer and the printer or have a modem suppressor between them.

The reason for needing the modem suppressor, is to allow the /// and the printer to communicate. This is called a, "Hardware Handshake." It is accomplished agreeing which lines and through which pins on the 24-pin RS 232 connector. The standard for communication is different from than the standard for a printer.

A modem suppressor is a cable with a female RS 232 connector on each end that transfers physically the signal from one pin to another so the computer is sending and receiving over the right pin combination. In this case the Data Terminal Ready and Request to Send are pins 20 and 4 respectively.

The modem suppressor is inexpensive, and in most cases the people selling the printer will supply it with no additional charge because so many computers require it with a serial printer.

The parallel setup is a different story. Data conversion is done via a hardware card that is usually installed in slot 1 or 4 in the Apple ///. These cards originally cost at least a hundred dollars and new, they are sold separately rather than as part of the printer. This is not as bad as it seems because most of the data conversion is done in the card so the parallel printer does not need as much circuitry as

the serial printer and consequently is usually cheaper.

Most of the time, parallel printers use broad ribbon connectors like the type that connect disk drives or the Profile to the Apple ///, rather than the round cable like the RS 232.

SETTING UP A PRINTER

Hooking the printer up physically is fairly simple because, for the most part, there is only two types of printers, and they couple together in only a couple ways. The problem of which printer is more than physical hookup, and it is still not all that easy to solve. There are two areas that must be in agreement before the printer will work.

First there is the problem of the right driver. The rule of thumb here is never to buy a printer that you aren't certain has a driver that works with the ///, and that you know where that driver can be found. While the rule is not always practical, finding a driver that will work with a particular printer is sometimes difficult.

Remember, not all printers have drivers written that will work with them. If you have a printer that does not have a specific driver known to work with it, then you must use the Trial and Error method. Here start with the most common drivers, generally the ones used by the greatest number of printers.

A few of those are the Apple Parallel driver (designed for the Apple Parallel Card and a bunch of "standard" printers), Qume, Diablo, or the "Standard" Apple Serial driver.

There are several good sources of drivers and hundreds available, including most user groups that still support the ///. By and large the majority of drivers have worked their way into the public domain and can be ordered for a small fee once found. Another source for some special purpose drivers is Sun Remarketing which also offers a selection of the more commonly available drivers.

Installing or changing a printer driver on the Apple /// is like changing any other driver. First you bring up Apple /// System Utilities, and then choose the third option on the main menu (S) System Configuration program (SCP). Next you have to read in your standard drive file using (R) Read a Driver File. Once you have your driver file in the computer you can edit the printer drive. If you need to delete the printer driver in your system and replace you use option (D) Delete a Driver. For a more detailed description of the way this portion of System Utilities works, see the March 1987 issue of ON THREE.

There must be agreement between the printer and the Apple /// on just what manner and speed will be used for transmitting data. In most cases it is the Apple /// driver that must be adjusted to meet the conditions of a particular printer. In come cases, there are toggle switches on the printer that may be used to set options. These toggle switches are also often called dip switches. For our purposes, the titles are interchangeable.

To set either the toggle switches, or the driver for the Apple ///, it is vital to understand just what speed and in what protocol the printer is expecting data. That brings us to the second don't. Avoid buying a printer without an operators manual. The manual is a key source in determining what protocols and capabilities are available and-or necessary for using the specific printer. Once again finding out which settings work for a particular printer by trial and error can be tedious, time consuming and very frustrating.

Remember that many printers have dip switches. These toggle switches allow some of the printer functions to be turned on or off, and certain others to be adjusted. Without an operators manual there is usually no way of telling what any given combination of settings of these switches mean. On some printers, it can be difficult to even find the switches without a manual to help.

We will assume that you are familiar with System Utilities, and are able to install a driver file. If you aren't, we suggest you review your manual, or the March 1987 issue of ON THREE.

With a manual, the Apple /// can be adjusted to fit most printers specifications for data transfer. Next month we will cover using the Edit Configuration Section of System Utilities to set up the standard variables for different drivers and printers, and also cover the special printer setup requirements of Word Juggler, Three Easy Pieces, and the Apple II Emulation mode. ///

Continued From Page 10

There is only so much that can be displayed on an 80-column screen. The article titles are truncated so the rest of the basic bibliographic data can be displayed — publication, date, volume, and issue. There is more.

You get a hint that there is more to see because the keyword BLOOM isn't on the screen anywhere. It has to be in the record somewhere, yes? You're sure when you zoom in (Ö-Z) on a record, as in Figure 2.

```
File: APPLE3.BIB.87        FIND RECORDS        Esc: Review/Add/Change

Find all records that contain BLOOM
Press @-F to change Find.

Record 67 of 384

TITLE: Getting There From Here: Number 9: Mail Call

MAGAZINE/NEWSLETTER: TAUTALES
AUTHOR: Bloom
SUBJECT: Column B09

PG(S): 6-7                MONTH: September        COLLECTION: -
  VOL: 5                    DAY: -
ISSUE: 12                  YEAR: 1987

NOTE: Bridge programs revisited: Mail List Manager (MLM) Utilities,
-: Merge ///.
-: -
-: -

Type entry or use * commands                                *-? for Help
```

Figure 2

As you can see, the full record contains much more than you see in the default column format. And you can see the value of Dave's annotations. You certainly would never guess from the title that the PFS mail/merge program MERGE /// was in that article. Never assume what you see on the column-format is all there is to a THREE WORKS file. Zoom in to get all the goodies.

In "zoom" mode, you can page through a file (or a "found" subset) to see what all is there. Just press Ö downarrow to see the next record, or you can back up a screen with Ö uparrow. This is a normal EZP feature.

As with any "keyword" search, you have to know half the answer before you can ask the question. You tell EZP to look for a keyword related to a topic. If it is a good keyword, you get one or more references. If the keyword isn't any good, you get frustrated. Remember you can always leaf through a data base file sequentially in "zoom out" mode. It will also help with your keyword selection. You'll see what kinds of words Dave tends to use to mean what kinds of things.

Don't blame Dave for the keywords in the CompuServe library files. He copied the file synopses verbatim as posted by the original contributors. BTW (That's "by the way" in CompuSpeak), it is a whole lot cheaper to get the contents of the data libraries via THREE WORKS than to download them yourself. Besides, the THREE WORKS versions are in the ever-so-useful EZP DB format already.

There is a special frustration in searching the bibliography files. You look for references for a topic of intense personal interest. You find a mother lode of references on that topic. Every one of those references is out of print. Unavailable. The publisher no longer exists. What THREE WORKS doesn't tell you is there is at least one person in the Apple /// community who does have a copy of that article or that program. There is nothing in THREE WORKS that says who to yell at to get what you need.

I have a modest suggestion. Write or call Dave. His address and home phone are in the documentation. In the first place, he's a collector. He probably has what you want. In the second place, he is buddies with a lot of other collectors. Remember, please be a sweetie and at least reimburse copying and mailing costs.

THE BOTTOM LINE

What can I say? There is so much good information consolidated in THREE WORKS that every Apple /// user could benefit from having it. It is so cheap that no Apple /// user needs to be without it. That is some kind of cost/benefit ratio, yes? If you have the original THREE WORKS, consider that Dave has reworked and improved every file, in addition to adding the 1987 bibliography. Buy it. I'd say that even if Cousin Vito weren't looking over my shoulder. ///

Review ON: ThreeWorks

by Dr. Allan M. Bloom

A Look At The *New & Improved* ThreeWorks

David Ottalini, co-chair of the Apple /// section of the Washington Apple Pi user group and real-life producer for Cable News Network and all around good guy, has a new (June 1988) edition of THREE WORKS, what he calls "a living record of the Apple /// Community."

As has been said, THREE WORKS is an Apple /// encyclopedia. If it doesn't have something about the /// you need, you maybe don't need your Apple ///. You're going to have to watch me during this review. I really like THREE WORKS. I fear I will gush over it. I fear I have already gushed. It is a good product, priced unreasonably low. I suggest that you buy it, lest Dave ask his cousin Vito to break your kneecaps.

Let me try to describe THREE WORKS in a nutshell. It is eight disk sides full of useful information related to the Apple /// in "/// E-Z Pieces" (EZP) data base format. And I mean full. I could not save the two Word Processor files to disk after modifying them for my printer. The "useful information" cannot be readily pigeonholed, although the Apple /// bibliography — an annotated list of articles referencing the Apple /// from day one through December 1987 — takes up the biggest chunk. After that, THREE WORKS gets very varied.

There are annotated lists of Apple /// terms, SOS completion codes (fatal and non fatal), monitor commands, file types, BASIC peeks and pokes, BASIC keywords, books and clubs and periodicals and vendors past and present, the latest versions of programs and device drivers, addresses of firms that still repair Apple ///'s, public domain programs and their sources, consultants and their specialties, contents of the CompuServe Apple /// libraries.

WHAT YOU GET AND WHAT YOU NEED

The eight disk sides contain 37 EZP files, two Word Processor and 35 Data Base files. The THREE.WORKS WP file on disk side one is a seven-page introduction to THREE WORKS. The other WP file is a two-page catalog of the disks, with its contents duplicated in a DB file if you prefer an on-line table of contents.

THREE WORKS contains four Apple /// bibliography files, one for 1980 through 1984, and one for each year thereafter. There are additional files for Apple /// books, periodicals, clubs, and bulletin boards. At the system level are files of monitor commands, error codes, file types, a glossary of Apple /// terms, and a list of firms that do repairs. The BASIC programmer can find a list of BASIC keywords and a list of peeks and pokes and calls.

There are files of all known commercial programs and devices drivers, plus the latest version numbers of each. There are also lists of public domain programs and how to get them. I'm sure I missed something. Let's just add "et cetera" to be safe.

THREE WORKS is not copy protected, and the author begs you to back up the four "flippy" disks onto eight single-sided volumes. He also notes that he has kept the price low, so there is little excuse for piracy. Considering the amount of work he put into THREE WORKS, screwing him out of his tiny royalty would be at best tacky. Also despicable. Besides, his Cousin Vito might find out.

It shouldn't need saying, but since THREE WORKS consists of EZP files, you need /// EZ Pieces to use it. This also implies a minimum 256K system unit and two disk drives — one of which may be a high capacity hard disk. If you don't have EZP, you are in the distinct minority of Apple /// users. I suggest that THREE WORKS is so useful that you might want to buy /// E-Z Pieces just to use THREE WORKS. Oops, I'm gushing again. Sorry about that.

Because THREE WORKS consists of EZP subfiles, the user documentation is mainly a brief paragraph on the contents of each file. The author assumes you know how

to use EZP and lets that program's documentation speak for itself. If you aren't fond of the EZP manual, buy the Campbell & Redden book "Working With Apple Works" (Hayden, ISBN 0-8104-6759-3). It's my favorite. Just watch out for the Apple Works features that its older brother doesn't have.

WHAT I THINK

I think THREE WORKS is too much. I doubt any single person would care about all the files on the eight disk sides, much as no one cares about every article in Encyclopaedia Britannica. The various THREE WORKS topics are in separate files, so you need not wade through everything to see the topic you're interested in. My main interest is the bibliography. I have a lot of the stuff printed about the Apple /// over the years, so much that it is a drag paging through articles and books trying to find a particular topic. I keep the four bibliography files on my hard disk for quick and easy access.

I keep copies of the other 33 files handy on floppy disk, with the printed catalog stuffed into the first disk's sleeve. Access is not as quick, but I cherish my hard disk space. There's an even better way. If you have ON THREE's The Desktop Manager, put the WP version of the file on your disk. Then invoke TDM's "File View" (Super Accessories Disk #2) to look at the catalog any time. Without bothering to bring the catalog into your desktop.

If you have hard disk space to spare and put all the THREE WORKS files there, you don't need the catalog file in any form. EZP's catalog (list of files) facility will tell you everything that the THREE WORKS catalog does.

THREE WORKS covers a wide variety of Apple /// topics, but it does not cover everything. If you looked only at THREE WORKS, you might assume that the Apple /// supported only the Business BASIC programming language. There are no sections on Assembler, COBOL, Forth, Fortran, or Pascal.

Dave is pretty sure those programming languages exist, but he doesn't know anything about them. He doesn't know much about SOS calls either. He does not devote specific sections to them. He does reference them in the annotated bibliography.

I keep saying "annotated" when talking about THREE WORKS files. That is the real key to their being useful. Every word in an EZP data base record can be searched for. In the case of the bibliography files or the CompuServe data library files, a title is probably not enough to really pin down its contents. The annotations that Dave put in the "scribble space" of each data base record are the keys to finding an article of interest on any given topic. If the text in the scribble space is good, you can use the EZP "find" command to select all records that match the particular topic you are interested in. I've found that Dave is very good at annotating THREE WORKS records. He is not perfect, but he is awfully good.

Let's see an example. Figure 1 is an example of the 1987 "bibliography" data base. I admit that I requested EZP to find (Ó-F) all occurrences of the last name of my favorite author. That eccentricity aside, each file looks about like the Figure 1 example.

The "column-format" view doesn't tell you much, except maybe that BLOOM showed up too much in 1987.

Continued On Page 8

```
File: APPLE3.BIB.87        FIND RECORDS        Esc: Review/Add/Change

Find all records that contain BLOOM
Press @-F to change Find.

TITLE                                        MAGAZINE        MONTH    VOL NO
=============================================================================
From BASIC to Pascal                         On Three        Mar      4   3
Things to Come                               TAUTALES        Dec      6   3
Getting There From Here: Number 1: We'       TAUTALES        Jan      5   4
Getting There from Here: Number 2: Bri       TAUTALES        Feb      5   5
Getting There from Here: Number 3: Me        TAUTALES        Mar      5   6
Getting There from Here: Number 4: The       TAUTALES        Apr      5   7
Getting There from Here: Number 6: II        TAUTALES        Jun      5   9
Getting There From Here: Number 7: Int       TAUTALES        Jul      5   10
Getting There From Here: Number 8: Wir       TAUTALES        Aug      5   11
Getting There From Here: Number 9: Mai       TAUTALES        Sep      5   12
Getting There From Here: Number 10: Pu       TAUTALES        Oct      6   1
Getting There From Here: Number 11: DI       TAUTALES        Nov      6   2
Getting There From Here: Number 12 Chr       TAUTALES        Dec      6   3
On The Trail of the Apple ///                WAP Journal     Feb      9   2
Disk /// Backup for the Apple ///            WAP Journal     Nov      9   11

Type entry or use * commands                                 *-? for Help
```

Figure 1

Business Basic Extensions

by Bob Consorti **Ultra-Fast Screen Save/Restores**

After publishing Taylor Pohlman's Challenge in the July issue of ON THREE, one of our readers complained that it's impossible to write a program for the contest. He stated that a program for the contest needed to do screen saves and screen restores. Not impossible, but incredibly slow using the REQUEST.INV invokable module.

Well, that weekend I wrote an invokable module for Business Basic that saves and restores arbitrary regions on the text screen. You simply tell it where the region is (window) and it handles saving and/or restoring it at ultra-fast assembly language speeds.

The Save/Restore routine can handle up to 256 screens of information at a time and has sophisticated memory management techniques that allow you to simply ask it to save or restore a screen. It completely handles finding a chunk of memory to store the requested screen in.

After writing the routine I realized that with it anyone who can program in Basic should be able to write a program to demonstrate the Mac-style pull down menus and dialog boxes. All it would take is a little effort. I hope there are some people who are now willing to make an effort and submit an entry for the contest.

The Programs

There are two program listings that follow. The first is the Business Basic test program that shows how to use the screen management routines. The second is the actual source for the invokable module.

The short test program draws a small box on the screen and allows you to move it around by pressing the arrow keys. It uses the invokable routines to manage saving and restoring the screen of the box and what's underneath.

The invokable is intended to speed up the display of information using a technique called caching. This involves first printing the text once the regular old slow way. Then you call the invokable to save the screen image. The next time you want to display the text, simply call the invokable to restore the screen in an instant!

A program that would demonstrate a Mac-style interface with pull down menus, dialog boxes and help screens would first print each and every screen image that could be displayed by the program. After it printed a screen it would use the invokable to save the image. When needed it would call the invokable to restore the screen.

This way you can instantaneously display those dialog boxes, help screens and pull down menus on command. With the screen management routines provided by this invokable you should be able to do very sophisticated programs that have a **modern interface**.

The invokable provides five routines that your Business Basic programs can use. The first is called like this:

```
PERFORM SaveScreen(%x1,%y1,%x2,%y2,%screen.num)
```

Give it the upper left (x1, y1), the lower right (x2, y2) coordinates of the region on the text screen you would like to save, the screen number (0-255) and the routine will quickly save the screen.

```
PERFORM RestoreScreen(%x1,%y1,%x2,%y2,%screen.num)
```

Call `RestoreScreen` with the screen number to restore and the coordinates you would like the screen restored to. Please note that the coordinates don't have to be the same as when the screen was originally saved. This way you can move regions of the screen around.

The dimensions of the restored region don't have to be the same as the saved region. As long as the number of characters saved/restored remain the same, the routine will restore the screen with the coordinates you request.

As you save screens the invokable will claim memory

from Basic to save the screen images into. If you decide you aren't going to use a screen anymore you should return its memory. That is done with this command:

```
PERFORM ReturnMemory(%screen.num)
```

Note that all of the coordinate and screen numbers ***must*** be integers. If you try to pass the routines real numbers they will respond with an appropriate error message.

There are two other routines that will aid in the initial saving of the screen images. When you're in the beginning of your program creating and then saving screens you can turn the screen *off* so the user doesn't see them being created. Simply call the `ScreenOff` routine. Likewise, when you want the user to see what's on the screen, call the `ScreenOn` routine.

The invokable allows the *dynamic* creation, change and removal of up to 256 text screens. There are a couple of things that you should do to help speed things up. If you have created and saved a large number of screens, changing the ones in the middle or near the first ones you saved will take more time than later ones you saved.

Say that you've saved 100 screens. The first one you saved you gave the # 0 the last one # 99. If you changed the contents of screen 0, the invokable would first delete screen 0 and then move the contents of all the other screens up in memory to fill the hole caused by deleting that screen. This can take a second or so depending on the number of screens. Then it would save the new screen number 0 at the bottom of memory.

If you have a number of static screens (ones that don't change), save them first. Then the ones that change will be at the bottom of memory and can be changed faster. Doing this will speed the saving of information. If you follow these guidelines it should never take more than a 1/4 second or so to save or change any screen.

Before you exit your program you should always call the `ReturnMemory` routine with the numbers off all the screens you've saved. If you don't do this and simply run another program, Basic will not be able to reclaim the memory used to hold the text screens and you will eventually run out of memory.

When calling this routine you should return the last screen that was saved first and the first screen saved last. This will assure that the invokable will quickly return the memory used by the screens back to Business Basic. If you don't it will take more time than necessary to return the memory to Business Basic.

That's about it folks. Have fun with this new routine and I hope to see some exciting contest entries soon. I'd like to have the contest finished by December so hurry! ///

The programs presented in this article are available on disk for only $9.95 plus $2 for shipping and handling. Order **Business Basic Screen Manager** today!

Program Listing #1: Business Basic Demonstration

```
10    REM ************************************************************
20    REM *                                                          *
30    REM * A Demonstration Program For The Save/Restore Invokable   *
40    REM * ------------------------------------                     *
50    REM * (c) 1988 by ON THREE, Inc.  All Rights Reserved          *
60    REM *                                                          *
70    REM * Written by Bob Consorti                                  *
80    REM *                                                          *
90    REM ************************************************************
100    ON ERR GOTO 120
110    GOTO 130
120    OFF ERR:IF ERR=20 THEN INVOKE"SAVE.REST.INV"
130    PERFORM ScreenOn:REM Check to see if the invokable is present
140    OFF ERR
200    PERFORM ScreenOff:REM Turn the screen off while we draw our boxes
210    HOME
220    PRINT"                                        "
230    PRINT"|                                      |"
240    PRINT"|     This is a test dialog box:       |"
250    PRINT"|                                      |"
260    PRINT"|  Use the arrow keys to move it       |"
270    PRINT"|    around the screen or press        |"
280    PRINT"|          ESCAPE to exit.             |"
290    PRINT"|______________________________________|"
295    PERFORM SaveScreen(%1,%1,%35,%8,%0):REM 1st (0) screen to save
300    HOME:CATALOG
310    x1%=1:y1%=1:x2%=35:y2%=8:REM Starting coordinates for the box
320    PERFORM SaveScreen(%x1%,%y1%,%x2%,%y2%,%1):REM 2nd (1) screen
325    PERFORM RestoreScreen(%x1%,%y1%,%x2%,%y2%,%0)
330    GET a$
335    PERFORM restorescreen(%x1%,%y1%,%x2%,%y2%,%1)
340    IF a$=CHR$(11) THEN 400:REM Move up
350    IF a$=CHR$(10) THEN 410:REM Move down
360    IF a$=CHR$(21) THEN 420:REM Move right
370    IF a$=CHR$(8) THEN 430:REM Move left
380    IF a$=CHR$(27) THEN 999:REM exit on an ESCAPE
390    GOTO 330
400    IF y1%=1 THEN 320
405    y1%=y1%-1:y2%=y2%-1:GOTO 320
410    IF y2%=24 THEN 320
415    y1%=y1%+1:y2%=y2%+1:GOTO 320
420    IF x2%=80 THEN 320
425    x1%=x1%+1:x2%=x2%+1:GOTO 320
430    IF x1%=1 THEN 320
435    x1%=x1%-1:x2%=x2%-1:GOTO 320
999    HOME
9999    FOR a%=0 TO 255:PERFORM ReturnMemory(%a%):NEXT a%
10000    REM ************************************************************
10010    REM *                                                          *
10020    REM * Always remember to return the memory used by each of     *
10030    REM * the screens that you use with the ReturnMemory call.     *
10040    REM *                                                          *
10050    REM * If you don't, Business Basic will not be able to use     *
10060    REM * that memory and it will be 'lost' until you restart.     *
10070    REM *                                                          *
10080    REM ************************************************************
```

Program Listing #2: Invokable Module Source

```
;
; Apple /// Business Basic Screen Management Routines
;
; by Bob Consorti (c) 1988 by ON THREE, Inc.  With all rights reserved.
;
; This file may not be reproduced without the express written permission of
; ON THREE, Inc.
;
; Version 1.00          First Started: 07/30/88    Last Revision: 08/01/88
;
;
; A Business Basic invokable module to enhance screen i/o.  These routines
; were written to facilitate the quick display of information and the fast
; saving/restoring of information.
;
;
; These routines are intended to speed up the display of information by a
; technique known as caching.  This allows the information to be written once
```

```
; via the regular slow method and then cached (saved) in a section of memory
; by calling the save screen routine.
;
; When the information is to be displayed again, the user calls the restore
; routine which then writes the information to the display very quickly.
;
; There are five routines that this invokable contains.  They are:
;
; Save Screen: Saves any region on the screen.
; ---------------------------
; PERFORM SaveScreen(%x1,%y1,%x2,%y2,%screen.num)
;
; Restore Screen: Restores the saved image to the screen.
; ---------------------------
; PERFORM RestoreScreen(%x1,%y1,%x2,%y2,%screen.num)
;
; ReturnMemory: Release the memory used by a screen back to Basic.
; ---------------------------
; PERFORM ReturnMemory(%screen.num)
;
; ScreenOff: Turns off the sceen so the screen data can be written
;            without the user seeing it.
; ---------------------------
; PERFORM ScreenOff
;
; ScreenOn: Turns the screen back on after it's been turned off.
; ---------------------------
; PERFORM ScreenOn
;
; These routines use a number of the internal Business Basic calls that
; are available to an invokable module.  The calls used are primarily to
; provide memory chunks for the saved screen data.
;
; The memory management is fairly complex and it works nicely.  Please do
; not change anything within and expect it to work as many of the routines
; call each other and are very much interwoven.
;
; Note that all items given to these routines must be integers (%).  Also,
; this invokable will save/restore up to 256 screens of information, depending
; only on the amount of memory available in your system.
;
; The item %screen.num must be an integer between 0 and 255.  Any number
; greater than these permissible values will generate an 'Range Error'.
;
; If there isn't enough memory available to save the requested screen, the
; routines will generate an 'Out Of Memory' error.
;
; Please note that when restoring a screen it may be in a different location
; than where the screen was originally saved.  The restored screen may also
; consist of different window dimensions.  It must, however, contain the same
; number of characters or SOS will not restore it.
;
; These routines will automatically take and restore memory from and to the
; Business Basic memory pool as needed.  No preallocation of memory is
; required as the routines take care of all memory management.
;
; ------------------------------------------------------------------------

; ------------------------------------------------------------------------
; Macro Defintions
; ------------------------------------------------------------------------

        .MACRO  SOS                     ;To make SOS calls
        BRK
        .BYTE   %1
        .WORD   %2
        .ENDM

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

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

        .MACRO  ResetView               ;For .CONSOLE i/o
        .BYTE   01
        .ENDM

        .MACRO  ViewTop
        .BYTE   02
        .ENDM

        .MACRO  ViewBottom
        .BYTE   03
        .ENDM

        .MACRO  Turn_Off
        .BYTE   14.
        .ENDM

        .MACRO  Turn_On
        .BYTE   15.
        .ENDM

; ------------------------------------------------------------------------
; Equates
; ------------------------------------------------------------------------

Open            .EQU    0C8             ;For SOS
Read            .EQU    0CA
Write           .EQU    0CB
Close           .EQU    0CC

D_Status        .EQU    82
D_Control       .EQU    83
Get_Dev_Num     .EQU    84

Request_Seg     .EQU    40
Find_Seg        .EQU    41
Change_Seg      .EQU    42
Get_Seg_Info    .EQU    43
Release_Seg     .EQU    45

Scrunch         .EQU    16.             ;For Business Basic
Expand          .EQU    17.

Dispatch        .EQU    0E4
Subrnumb        .EQU    0E7

Buffer          .EQU    0E8             ;Indirect pointer
X_Byte          .EQU    0E8+1601        ;    "       "    x-byte

Indir           .EQU    0EA             ;And another one
In_XB           .EQU    0EA+1601

; ------------------------------------------------------------------------
; Generic routines used throughout the various invokable modules.
; ------------------------------------------------------------------------

                .PROC   Generic_X

                .DEF    Open_Console, Close_Console
                .DEF    X_1, X_2, Y_1, Y_2, Bounds_Check, Mult_Temp
                .DEF    Con_Name
                .DEF    Basic_Error
                .DEF    S_Num, Num_Active
                .DEF    Buf
                .DEF    SR_Pages, SR_High, SR_XB
                .DEF    Fix_It, Bump
                .DEF    Get_Memory, Got_Memory
                .DEF    Mult_Temp
                .DEF    GM_Info
                .REF    Rel_Num, Decr_Num

                RTS                     ;In case anyone calls 'Generic_X'

        .ASCII  "(c) 1988 by ON THREE, Inc.  All rights reserved.  "
        .ASCII  "Written by Bob Consorti."

Open_Console    SOS     Open, Con_List  ;Get a reference number to write to

                LDA     Con_Num         ;And setup for the close call
                STA     Close_C_Num

                RTS                     ;And return it in 'A'

Close_Console   SOS     Close, Close_C_List     ;Make sure to close .CONSOLE

                RTS

Basic_Error     LDA     #14.            ;We want to raise a basic error
                STA     Subrnumb        ;condition

                JSR     Dispatch        ;Does not return

Bounds_Check    SEC                     ;See how many rows high it is
                LDA     Y_2
                SBC     Y_1
                TAX
                INX
                STX     Y_Size
                STX     Mult_Of

                SEC                     ;And how many columns wide
                LDA     X_2
                SBC     X_1
                TAX
                INX
                STX     X_Size
                STX     Mult_By

                LDA     Y_Size          ;If either is 0 we can't save it so
                BEQ     Bad_Bounds      ;we generate an error condition
                BMI     Bad_Bounds

                LDA     X_Size
                BEQ     Bad_Bounds
                BMI     Bad_Bounds

                JSR     Mult_Nums       ;Figure out how big this window is

                CLC                     ;And add in the three bytes of
                LDA     Mult_Temp       ;viewport information needed by the
                ADC     #03             ;.CONSOLE save viewport routine
                STA     Mult_Temp
                BCC     $00
                INC     Mult_Temp+1

$00             RTS

Bad_Bounds      LDX     #10.            ;We have a range error
                JMP     Basic_Error     ; This routine never returns

; ------------------------------

Mult_Nums       LDA     #00             ;Multiply the two numbers in
                STA     Mult_Temp       ;Mult_Of and Mult_By and leave the
                LDX     #08             ;Result in Mult_Temp (Low, High)
Mult_Loop       LSR     Mult_By
                BCC     No_Add
                CLC
                ADC     Mult_Of
No_Add          ROR     A
                ROR     Mult_Temp
                DEX
                BNE     Mult_Loop
                STA     Mult_Temp+1
                RTS

; ------------------------------

Get_Memory      STX     Find_Pages      ;Save the number of pages to get
                STX     Chg_Pages
```

Continued On Page 16

Great Products For Your Apple ///

Productivity Tools	Price	S/H
Apple File ///	$99.00	$7.00
BPI General Accounting	$99.00	$7.00
BPI Accounts Payable	$99.00	$7.00
BPI Accounts Receivable	$99.00	$7.00
BPI Job Costing	$99.00	$7.00
General, Payable, Receivable Bundle	$199.00	$10.00
Draw On ///	$99.00	$5.00
Draw On /// Graphics Tablet Version	$99.00	$5.00
Mail List Manager	$99.00	$7.00
Quick File ///	$50.00	$7.00
Side Print ///	$29.95	$3.00
Lets you print wide /// E-Z Pieces & VisiCalc files sideways		
StemSpeller ///	$59.95	$5.00
Lets you spell check /// E-Z Pieces word processor files		
StemWriter ///	$79.95	$7.00
The Communications Manager	$39.95	$3.00
The Graphics Manager	$29.95	$3.00
/// E-Z Pieces	$135.00	$3.50
/// E-Z Pieces & StemSpeller Bundle	$169.95	$7.50

Utilities

Backup ///	$49.95	$3.00
Lazarus /// File Recover Utility	$39.95	$2.00
RAM Diagnostic Disk		
Specify 128K, 256K or 512K version	$15.00	$2.00
Selector /// Program Switching Utility	$69.95	$7.00
UnCopyProtect Driver	$29.95	$3.00

Make backup copies of AppleWriter, VisiCalc & Advanced VisiCalc For stand-alone use or for with Selector ///

Games

BattleFleet	$19.95	$2.00
Card Machine (Black Jack)	$24.95	$2.00
Crossword Scrambler	$9.95	$3.00
Fruit machine (The Slots!!!)	$19.95	$2.00
Sandman (xxx-man Arcade game)	$19.95	$2.00
Game Combo (All of the above games)	$49.95	$3.00

The Desktop Manager Desk Accessories that can be used within all Apple /// programs

Main Accessories	$69.00	$6.00
Includes Note Pad, Appointment Calendar, Calculator, the text pickup & paste utility and an automatic installation program		
Disk Manager	$29.95	$3.00
Provides System Utilities commands at any time		
Macro Manager	$29.95	$3.00
Provides powerful macros within all programs		
Communications Manager	$39.95	$3.00
Gives powerful communications within all programs		
ASCII Conversion Table	$4.95	$3.00
Super Accessories No. 1	$29.95	$2.00
Includes Printer Setup, TypeWriter, Envelope Labeler, New Desktop Setup, Make Subdirectory and Menu Manager that allows an unlimited # of accessories		
Super Accessories No. 2	$19.95	$3.00
Includes File View, File Print, Change Font, Show Time, Change File Type, Print Screen, Reload & Exit, PKASO/U and Warning		
Super Accessories No. 3	$24.95	$3.00
Includes Accessory Manager, Character Rulers, Hex View & Print, Info Help, SOS Errors, Block View/Edit		
The Graphics Manager	$29.95	$3.00
Print *any* graphic image in any size on most printers		
Sandman (xxx-man Arcade game)	$19.95	$2.00
TDM /// Tool Kit for programmers	$19.95	$2.00

Hardware Products

Apple /// with 256K and monitor	$499	$35
Apple /// with 512K and monitor	$899	$35
512K Memory Upgrade	$399	$10
256K Memory Upgrade	$169	$10
128K Memory Board	$15	$10
512K with 0K of memory	$159.95	$10
512K with 256K of memory	$299.95	$10
Memory Board Insurance (1 Year)	$50	$10
512K or 256K installation	$25	$35
ON THREE O'Clock/Calendar Kit	$39.95	$3.00
Interlace Kits	$75.00	$3.00
CP/M Card With Complete Docs.	$99.00	$7.00
Parallel Interface Card and driver	$79.95	$6.00
Seikosha Color Printer with interface card and the Color Graphics Manager	$259.95	$20
Replacement color ribbons	$25.00	$4.00
Ribbon inker for color ribbons	$20.00	$3.50
Silentype thermal printer	$49.95	$10.00
Silentype paper (1 roll)	$5.99	$2.00
Sider 20 Hard Disk drive with interface card and the driver & documentation	$899	$35
If you have an Apple II Sider, send us the interface card for modification and we'll send you back an Apple /// card with driver and documentation	$150	$7.50
Apple UniDisk ///.5 800K 3.5 " drive with interface card and driver software	$499	$10
UniDisk ///.5 Driver & Documentation	$50	$3.00
CPS 800K 3.5 " drive with interface card and driver software	$399	$10
CPS interface card & driver only	$149	$5.00
Connects any 2 Mac 800K, 400K or Apple II 140K drives		
CPS 800K 3.5 " drive only	$249	$6.00
CPS 400K 3.5 " drive only	$145	$6.00
CPS 140K drive only	$139	$6.00
CPS Driver & Documentation only	$50	$3.00
Apple //e Mouse & Interface Card Use with Draw ON ///, TDM & Games	$160.00	$5.00
TrackBall /// - Use with Draw ON ///, TDM, Games and all other /// programs	$89.95	$6.00
1200 BAUD Modem with cable	$129.95	$10.00
2400 BAUD Modem with cable	$229.95	$10.00
The above modems include the Communications Manager		
Titan ///+//e (Apple //e Emulation)	$349	$10.00

Miscellaneous Products And Services

Apple /// Beginner Pack	$99.00	$7.00
Includes McGraw Hill Guide to the Apple ///, Lazarus, ThreeWorks, System Utilities Product Training Pak, Will Someone Tell Me What An Apple // Can Do? guide, Cross-Word Scrambler, Mr. Sandman, Card and Fruit Machine games. Normally over $200!!!		
McGraw Hill Guide To The Apple ///	$29.95	$3.00
System Utilities Product Training Pak	$14.95	$3.00
ThreeWorks (requires /// E-Z Pieces)	$39.95	$3.00
Bibliography of everything ever done for the Apple ///		
Disks Of the Month	$9.95	$2.00
Taken from the pages of ON THREE Magazine		
#1 Disk Pak 1, Key Things Graphics Demos & More.		
#2 DOS File Lister, CharDownLoad, Graphic Demos +.		
#3 Keyboard Change Utility, WPL utilities, Graphic Sketcher ///, Disk Formatting Utility.		
#4 Emulation Patch, Access /// Startup, Pascalculator, Radiate Graphic Demos, Musical medolies, graphics.		
#5 Access /// Startup, Ben's Super Slot Machine, PFS-WPL Merge, Decision Support Spreadsheet, Draw ON demos.		
#6 Basic Program Lister, NoiseMaker, Spirograph, Surface, Golfball graphic demos and Draw ON pictures & fonts.		
#7 Imagehelper, Sorting and Printer Control.		
#8 Directory Sorter, Drive Cleaner and File Reader.		
#9 Graphics, Music and more.		
#10 Graphics Football Pool, Useful Pascal Prompt		
#11 Apple DMP, ImageWriter, Epson graphics printing utility. Graphics and audio demos.		
#12 Lottery number picker, Two-way directory catalog, Find Text Files on disk, Print Files To Screen or disk.		
Standard Device Drivers Manual	$39.95	$3.00
Dust Cover for Apple ///	$11.95	$2.00
Dust Cover for Apple /// with profile	$12.95	$2.00
I Love My Apple /// T-Shirts sm., med., lg., yellow, blue, white, tan	$11.95	$3.00
I Love My Apple /// Sweatshirts small only yellow, blue, white, silver	$18.95	$3.00
ON THREE back issues	$5.00 each	
ON THREE Binders	$9.95	$3.00
ON THREE subscription	$30.00 per year	
Resource Guide for the Apple ///	$15.00	$3.00
"Will Someone Please Tell Me What An Apple /// Can Do?" Program Guide	$3.00	$2.00

Programming & Languages

Apple /// Pascal	$99.00	$5.00
Pascal Debugger version 1.2 (Update)	$49.95	$7.00
Business Basic	$99.00	$5.00
Cobol ///	$99.00	$5.00
Apple /// Fortran (req. Pascal)	$99.00	$5.00
Device Driver Writers Guide	$29.00	$3.00
SOS Reference Manuals	$59.95	$3.00
Service Reference Manual & Schematics	$49.00	$5.00

Program Updates

With your order you must send in your original disk to verify ownership for the updates listed below:

Apple Writer 4.0	$15.00	$2.00
Lets you use the full memory, wide columns and more!		
Business Basic version 1.23	$15.00	$2.00
Lets your Basic programs be larger than 64K		
Draw ON /// Printer Update	$15.00	$2.00
Lets you use Draw ON with different printers		
Graphics Manager (Color or B/W)	$15.00	$2.00
PFS: File, PFS: Report	$15.00	$2.00
Copyable and installable under Catalyst or Selector ///. **PFS: Report can't be installed under Catalyst.**		
Word Juggler	$15.00	$2.00
Copyable and installable under Selector /// (256K or 512K)		

Order Form

ON THREE, Inc.
ATTN: Order Dept.
P.O. Box 3452
Kirkland, Washington 98033 U.S.A.

Order by calling TOLL FREE: (800) 443-8877
In Washington: (206) 823-0516
Customer Service: (206) 823-0516
Technical Help: (818) 701-1355

Phone in an
ORDER TODAY!
Ask About Our Quantity Discounts!

VISA, Master Card, American Express, money orders and personal checks accepted. Add $3 for C.O.D. Add 3% surcharge to total on orders charged to American Express. CA residents must add 6.5% sales tax, WA residents must add 8.1% sales tax. Overseas customers can pay by credit card, international money order, a check drawn on a U.S. bank or bank transfer. Call first to ascertain foreign shipping costs and mode of transport. Returns must have prior approval and are subject to a restocking fee. We do not copy-protect any of our software products so there are NO returns allowed on software. Not responsible for typographical errors. Please allow up to four weeks for delivery as some items may be backordered.

QUANTITY	PRODUCT DESCRIPTION	EACH	TOTAL	SHIPPING

Name _____
Company _____
Address _____
City _____ **State** _____ **Zip** _____
Credit Card _____ **Card No.** _____
Exp. Date _____ **Phone** _____

PRODUCT SUB TOTAL	
SHIPPING SUBTOTAL	
C.O.D. ($3.00)	
CA 6.5%, WA 8.1% TAX	
3% AM/EX SURCHARGE	
GRAND TOTAL	

Continued From Page 13

```
                LDA     #Scrunch        ;We want to give memory from Basic
                STA     Subrnumb        ;back to the SOS available pool

                TXA                     ;Get this many pages
                JSR     Dispatch

                LDA     Got_Memory      ;If we already have taken memory, just
                BNE     GM_Extend       ;extend the segment downward in memory

                SOS     Find_Seg, Seg_List      ;And lets see if we can get
                                                ;that many pages from SOS
                BEQ     Aft_GM_Err      ;No errors here please...

GM_Err          LDX     #07             ;Give them an 'Out Of Memory' error
                JMP     Basic_Error

                JSR     Dispatch        ;Does not return

Aft_GM_Err      LDA     #80             ;Setup our flag that indicates we
                STA     Got_Memory      ;have some memory

                LDA     Mem_Num         ;Get the segment number and ...
                STA     Rel_Num         ;Store it for the release call
                STA     Chg_Num         ;and change segment call
                STA     Get_Num         ;and the get segment info call
                STA     Decr_Num        ;and for the 2nd change segment call

GM_Setup        CLC                     ;Setup the X-Byte
                LDA     Mem_Base
                ADC     #80
                STA     X_Byte
                STA     Buf+2

                LDA     #00             ;And the starting address of memory
                STA     Buffer
                STA     Buf

                SEC
                LDA     Mem_Base+1      ;Convert to extended address notation
                SBC     #20

                STA     Buffer+1
                STA     Buf+1

                RTS

GM_Extend       SOS     Change_Seg, Chg_List    ;Reclaim some more memory
                BNE     GM_Err

GM_Info         SOS     Get_Seg_Info, Get_List  ;Get the new starting address

                LDA     Get_Base
                STA     Mem_Base
                LDA     Get_Base+1
                STA     Mem_Base+1

                JMP     GM_Setup

; ---------------------------------------------------------------

BUMP            INC     BUFFER+1                ;Step to the next page

FIXIT           LDA     BUFFER+1                ;If pointer = 0 or if = 80
                BEQ     $2                      ;then it must be adjusted
                CMP     #0FF                    ;IF POINTER <> FF THEN POINTER
                BNE     $1                      ;IS OK THEREFORE EXIT
                LSR     BUFFER+1                ;FFXX -> 7FXX
                INC     X_Byte                  ;BANK N -> BANK N+1
$1              RTS

$2              LDA     #080                    ;00XX -> 80XX
                LDY     X_Byte                  ;IF NOT (80)
                CPY     #080
                BNE     $3                      ;THEN SKIP AHEAD
                LDA     #020                    ;ELSE MUST DO ADJUST
                LDY     #090                    ;OF MSB AND BANK NUMBER

$3              DEY                             ;DEC BANK
                STY     X_Byte                  ;AND UPDATE BANK
                STA     BUFFER+1                ;UPDATE MSB
                RTS

; ---------------------------------------------------------------

Seg_List        .BYTE   06              ;$06 parms
                .BYTE   02              ;Data may be in any bank
                .BYTE   33              ;ID = user program and data
Find_Pages      .WORD   0000            ;Number of pages to get is here
Mem_Base        .WORD   0000
Mem_Limit       .WORD   0000
Mem_Num         .BYTE   00

Chg_List        .BYTE   03
Chg_Num         .BYTE   00
                .BYTE   01              ;Add before the 'base' address
Chg_Pages       .WORD   0000            ;Number of pages wanted

Get_List        .BYTE   05
Get_Num         .BYTE   00
Get_Base        .WORD   0000
                .WORD   0000
                .WORD   0000
                .BYTE   00

Got_Memory      .BYTE   00              ;Flag indicating memory already taken

; ---------------------------------------------------------------

Con_List        .BYTE   04
                .WORD   Con_Name
Con_Num         .BYTE   00
                .WORD   COP_List
                .BYTE   01

COP_List        .BYTE   03              ;Open for reading and writing

Con_Name        .BYTE   08
                .ASCII  ".CONSOLE"

Close_C_List    .BYTE   01
Close_C_Num     .BYTE   00

X_1             .WORD   0000            ;Window coordinates of the region
X_2             .WORD   0001            ;on the screen to save
Y_1             .WORD   0000
Y_2             .WORD   0000

X_Size          .BYTE   00
Y_Size          .BYTE   00

Mult_Temp       .WORD   0000            ;Used by the multiply routines
Mult_Of         .BYTE   00
Mult_By         .BYTE   00

S_Num           .WORD   0000            ;Screen number to use

Num_Active      .BYTE   00              ;Number of segments currently used

Buf             .WORD   0000            ;Starting memory address
                .BYTE   00

; ---------------------------------------------------------------

;Table of pointers to screen data

SR_High         .BLOCK  128.,128.       ;High addresses

SR_XB           .BLOCK  128.,128.       ;X-Bytes

SR_Pages        .BLOCK  128.,128.       ;Number of pages in this segment

; ===============================================================

                .PROC   Screen_Off

                .REF    Open_Console, Close_Console

                JSR     Open_Console

                STA     Off_Num

                SOS     Write, Off_W_List       ;Write the screen off char.

                JSR     Close_Console

                RTS                     ;Exit back to basic like this

Off_W_List      .BYTE   03
Off_Num         .BYTE   00
                .WORD   Off_Data
                .WORD   0001

Off_Data        Turn_Off                ;The off data list

; ===============================================================

                .PROC   Screen_On

                .REF    Open_Console, Close_Console

                JSR     Open_Console

                STA     On_Num

                SOS     Write, On_W_List        ;Write the screen on char.

                JSR     Close_Console

                RTS                     ;Exit back to basic like this

On_W_List       .BYTE   03
On_Num          .BYTE   00
                .WORD   On_Data
                .WORD   0001

On_Data         Turn_On                 ;The on data list

; ===============================================================

                .PROC   Return_Memory,1

                .DEF    Rel_Num, Decr_Num, Reclaim_Mem
                .REF    S_Num, Num_Active
                .REF    Buf, Basic_Error
                .REF    SR_Pages, SR_High, SR_XB
                .REF    Fix_It, Bump
                .REF    Got_Memory
                .REF    GM_Info

Reclaim_Mem     POP     Our_Addr        ;Save the return address

                POP     S_Num           ;Get the screen they want to return

                LDA     S_Num+1         ;Make sure it's not out of range
                BEQ     $00

                LDX     #05             ;Give them an 'Illegal Quantity' error
                JMP     Basic_Error     ;Does not return

$00             LDY     S_Num           ;If no segment is there just exit
                LDA     SR_Pages,Y
                STA     Decr_Pages
                BNE     $010

                JMP     Re_Exit

$010            LDA     Num_Active      ;If there's only one active screen
                CMP     #01             ;we can release our entire segment
                BNE     $10
```

```
                JSR     Clear_Entry     ;Remove all traces of the entry
                SOS     Release_Seg, Rel_List   ;Release the memory to SOS
                LDA     #00
                STA     Got_Memory      ;We don't have it anymore

                LDA     #Expand         ;We want to give memory from SOS
                STA     Subrnumb        ;back to Basic

                LDA     #00             ;All memory available
                JSR     Dispatch

                JMP     Re_Exit         ;Go to the exit routine

$10             LDA     SR_High,Y       ;Check to see if the screen they
                CMP     Buf+1           ;want to remove is at the beginning
                BNE     Rem_Mid         ;of our memory segment
                LDA     SR_XB,Y
                CMP     Buf+2
                BNE     Rem_Mid

Rem_Clear       JSR     Clear_Entry     ;Remove all traces of the entry

                SOS     Change_Seg, Decr_List   ;Release the memory to SOS

                JSR     GM_Info         ;Reset the pointers

                LDA     #Expand         ;We want to give memory from SOS
                STA     Subrnumb        ;back to Basic

                LDA     Decr_Pages      ;Just the amount we just released
                JSR     Dispatch

                JMP     Re_Exit         ;And exit

Rem_Mid         LDA     #00             ;Setup the low byte
                STA     Buffer

                CLC                     ;And the high byte
                LDA     SR_High,Y
                STA     Comp_High
                ADC     Decr_Pages
                TAX
                DEX
                STX     Buffer+1

                LDA     SR_XB,Y                 ;And the X-byte
                STA     Comp_XB
                STA     X_Byte

                JSR     Fix_It          ;Adjust for any addressing problems

                LDA     Buffer          ;Setup the buffer to move into
                STA     Indir
                LDA     Buffer+1
                STA     Indir+1
                LDA     X_Byte
                STA     In_XB

                LDY     S_Num           ;Setup the buffer to move from
                LDA     #00
                STA     Buffer
                LDA     SR_High,Y
                TAX
                DEX
                STX     Buffer+1
                LDA     SR_XB,Y
                STA     X_Byte

                JSR     Fix_It          ;Adjust for any addressing problems

$10             LDY     #00             ;Move a page
$20             LDA     @Buffer,Y
                STA     @Indir,Y
                DEY
                BNE     $20

                JSR     Bump_Back       ;Adjust the pointers

                LDA     Buffer+1        ;Keep looping through moving the
                CMP     Buf+1           ;pages until we've finished with
                BNE     $10             ;all of them
                LDA     X_Byte
                CMP     Buf+2
                BNE     $10

                LDY     #00             ;Move the last page
$30             LDA     @Buffer,Y
                STA     @Indir,Y
                DEY
                BNE     $30

; Now adjust the addresses in the table of any screen that moved.  The only
; ones that have moved are the ones which were lower in memory than the one
; we just removed.

Adjust_Entries  LDX     #00                     ;Start with entry #00

$00             CPX     S_Num                   ;Don't match it against itself
                BEQ     $50

                LDA     SR_Pages,X              ;Also, we don't need to do
                BEQ     $50                     ;anything to empty entries

                LDA     SR_XB,X                 ;Compare the x-bytes
                CMP     Comp_XB
                BCC     $10
                BNE     $50                     ;If equal we must ...

                LDA     SR_High,X               ;Compare the high byte
                CMP     Comp_High
                BCC     $10
                BCS     $50

$10             LDA     #00                     ;Adjust to the new address
                STA     Buffer
                LDA     SR_High,X
                STA     Buffer+1
                LDA     SR_XB,X
                STA     X_Byte

                LDA     Decr_Pages              ;Setup how many pages to
                STA     Temp_Pages              ;adjust the address

$20             JSR     Bump                    ;Adjust to the new address

                DEC     Temp_Pages
                BNE     $20                     ;See if we've reached the end

                LDA     Buffer+1                ;If buffer >= 8000 then it
                CMP     #80                     ;must be adjusted
                BCC     $30

                SEC
                SBC     #80
                STA     Buffer+1

                INC     X_Byte

$30             LDA     Buffer+1                ;When we're finished adjusting
                STA     SR_High,X               ;the address, get the new
                LDA     X_Byte                  ;address and save it in the
                STA     SR_XB,X                 ;correct spots.

$50             INX                             ;Move to the next entry
                BNE     $00                     ;If we're done, fall through

                LDY     S_Num                   ;Setup Y for the clearing

                JMP     Rem_Clear               ;Remove all traces of the entry
;-------------------------------------------------------------------------------
Re_Exit         PUSH    Our_Addr

                RTS                     ;Return like this
;-------------------------------------------------------------------------------
Bump_Back       DEC     BUFFER+1                ;Go back a page

                LDA     BUFFER+1                ;IF POINTER = 0 THEN POINTER
                BEQ     $2                      ;MUST BE ADJUSTED
                CMP     #0FF                    ;IF POINTER <> FF THEN POINTER
                BNE     $1                      ;IS OK THEREFORE EXIT
                LSR     BUFFER+1                ;FFXX -> 7FXX
                INC     X_Byte                  ;BANK N -> BANK N+1
$1              JMP     In_Back

$2              LDA     #080                    ;00XX -> 80XX
                LDY     X_Byte                  ;IF NOT (80)
                CPY     #080
                BNE     $3                      ;THEN SKIP AHEAD
                LDA     #020                    ;ELSE MUST DO ADJUST
                LDY     #090                    ;OF MSB AND BANK NUMBER
$3              DEY                             ;DEC BANK
                STY     X_Byte                  ;AND UPDATE BANK
                STA     BUFFER+1                ;UPDATE MSB

In_Back         DEC     Indir+1                 ;Go back a page

                LDA     Indir+1                 ;IF POINTER = 0 THEN POINTER
                BEQ     $2                      ;MUST BE ADJUSTED
                CMP     #0FF                    ;IF POINTER <> FF THEN POINTER
                BNE     $1                      ;IS OK THEREFORE EXIT
                LSR     Indir+1                 ;FFXX -> 7FXX
                INC     In_XB                   ;BANK N -> BANK N+1
$1              RTS

$2              LDA     #080                    ;00XX -> 80XX
                LDY     In_XB                   ;IF NOT (80)
                CPY     #080
                BNE     $3                      ;THEN SKIP AHEAD
                LDA     #020                    ;ELSE MUST DO ADJUST
                LDY     #090                    ;OF MSB AND BANK NUMBER
$3              DEY                             ;DEC BANK
                STY     In_XB                   ;AND UPDATE BANK
                STA     Indir+1                 ;UPDATE MSB

                RTS
;-------------------------------------------------------------------------------
Clear_Entry     LDA     #00             ;Clear the screen from our table
                STA     SR_Pages,Y
                STA     SR_High,Y
                STA     SR_XB,Y

                DEC     Num_Active      ;And reduce our count by one

                RTS
;-------------------------------------------------------------------------------
Rel_List        .BYTE   01
Rel_Num         .BYTE   00

Decr_List       .BYTE   03
Decr_Num        .BYTE   00              ;Release memory from the base of the
                .BYTE   00              ;memory segment
Decr_Pages      .WORD   0000

decr_error      .byte   00

Temp_Pages      .BYTE   00

Comp_High       .BYTE   00
```

```
Comp_XB         .BYTE   00
Our_Addr        .WORD   0000            ;Return address is saved here
;-------------------------------------------------------------

                .PROC   SaveScreen,5    ;Save the screen region

                .REF    X_1, X_2, Y_1, Y_2, Bounds_Check
                .REF    Open_Console, Close_Console, Con_Name
                .REF    Reclaim_Mem
                .REF    Basic_Error
                .REF    Num_Active
                .REF    SR_Pages, SR_High, SR_XB
                .REF    Get_Memory, S_Num
                .REF    Mult_Temp
                .DEF    Pre_Set_View, Rest_Close, Cont_2_List
                .DEF    Return_Addr

                POP     Return_Addr     ;Save the return address

                POP     S_Num           ;Get the screen they want to save

                POP     Y_2             ;Get the lower right corner
                POP     X_2
                POP     Y_1
                POP     X_1

                LDA     Num_Active      ;The first time through we must init
                BNE     $00             ;our tables or they will contain junk

                LDA     #00             ;Clear the Save/Restore table
                TAY
$000            STA     SR_Pages,Y
                STA     SR_High,Y
                STA     SR_XB,Y
                DEY
                BNE     $000

$00             LDA     S_Num+1         ;The number must be from 0-255
                BEQ     $10             ;or ...

                LDX     #05             ;Give them an 'Illegal Quantity' error
                JMP     Basic_Error     ;Does not return

$10             LDY     S_Num           ;Check to see that the entry they want
                LDA     SR_Pages,Y      ;is inactive
                BEQ     $20

                PUSH    S_Num           ;Otherwise, first reclaim the memory

                JSR     Reclaim_Mem     ;that's currently being used

$20             JSR     Bounds_Check    ;And calculate the size of the region

                LDX     Mult_Temp+1     ;Get this many pages from basic
                INX                     ;Inc by one to add the low byte

                TXA                     ;Save the number of pages used
                LDY     S_Num
                STA     SR_Pages,Y

                INC     Num_Active      ;And inc the seg counter

                JSR     Get_Memory      ;Go get it

                JSR     Pre_Set_View            ;Preserve the console settings
                                                ;and open up the viewport
                LDY     S_Num                   ;Index into the right location

                LDA     Buffer+1                ;Save the starting address
                STA     SR_High,Y               ;High byte ...

                LDA     X_Byte
                STA     SR_XB,Y                 ;And X_Byte ...

                SOS     D_Status, Stat_2_List   ;Preserve the viewport

                JSR     Rest_Close              ;Restore the console options
                                                ;and close the console driver
                PUSH    Return_Addr     ;Set the return address up

                RTS                     ;Exit back to basic like this

; ---------------------------------------------

Pre_Set_View    SOS     Get_Dev_Num, C_List     ;Get the .CONSOLE device num

                LDA     C_Dev_Num
                STA     S_Stat_1
                STA     S_Stat_2
                STA     Cont_1_Num
                STA     Cont_2_Num

                SOS     D_Status, Stat_1_List   ;Preserve the .CONSOLE
                                                ;status table
                JSR     Open_Console            ;Open the .CONSOLE for a moment

                STA     R_Ref

                LDX     X_1                     ;Adjust the viewport
                DEX                             ;settings to account for
                STX     R_X_1                   ;fact that the basic viewport
                LDX     Y_1                     ;starts at 1, not 0 like the
                DEX                             ;.CONSOLE likes it
                STX     R_Y_1

                LDX     X_2
                DEX
                STX     R_X_2
                LDX     Y_2
                DEX
                STX     R_Y_2

                SOS     Write, R_List           ;Set the viewport to here

                RTS

Rest_Close      SOS     D_Control, Cont_1_List  ;Restore the cursor options

                JSR     Close_Console

                RTS

; ---------------------------------------------

C_List          .BYTE   02
                .WORD   Con_Name
C_Dev_Num       .BYTE   00

Stat_1_List     .BYTE   03
S_Stat_1        .BYTE   00
                .BYTE   01                      ;Preserve status table
                .WORD   S_1_List

S_1_List        .BYTE   90.                     ;Leave at least 90 bytes for
                .BLOCK  90.                     ;the console status table

Stat_2_List     .BYTE   03
S_Stat_2        .BYTE   00
                .BYTE   18.                     ;Preserve viewport
                .WORD   Buffer

Cont_1_List     .BYTE   03
Cont_1_Num      .BYTE   00
                .BYTE   01                      ;Restore status table
                .WORD   S_1_List

Cont_2_List     .BYTE   03
Cont_2_Num      .BYTE   00
                .BYTE   18.                     ;Restore viewport
                .WORD   Buffer

; ---------------------------------------------

R_Data          ResetView               ;Reset to full screen

                .BYTE   26.             ;Go to the lower right corner
R_X_2           .BYTE   00
R_Y_2           .BYTE   00
                ViewBottom              ;And set the viewport bottom

                .BYTE   26.             ;Go to the top left corner
R_X_1           .BYTE   00
R_Y_1           .BYTE   00
                ViewTop                 ;And set the viewport top

R_Len           .EQU    *-R_Data

R_List          .BYTE   03
R_Ref           .BYTE   00
                .WORD   R_Data
                .WORD   R_Len

Return_Addr     .WORD   0000            ;Our return address is stored here

; ---------------------------------------------------------------

                .PROC   Restore_Screen,5        ;Restore the screen region

                .REF    X_1, X_2, Y_1, Y_2
                .REF    Basic_Error
                .REF    SR_High, SR_XB, SR_Pages
                .REF    S_Num
                .REF    Pre_Set_View, Rest_Close, Cont_2_List
                .REF    Return_Addr

                POP     Return_Addr     ;Save the return address

                POP     S_Num           ;Get the screen they want to restore

                POP     Y_2             ;And where the want to restore it
                POP     X_2
                POP     Y_1
                POP     X_1

                LDA     S_Num+1         ;The number must be from 0-255
                BEQ     $00             ;or ...

                LDX     #05             ;Give them an 'Illegal Quantity' error
                JMP     Basic_Error     ;Does not return

$00             LDY     S_Num           ;Check to see that the entry they want
                LDA     SR_Pages,Y      ;is not inactive
                BNE     $10

                LDX     #10.            ;If it is we exit with a range error
                JMP     Basic_Error     ;as we can't restore nothing

$10             JSR     Pre_Set_View            ;Preserve the console settings
                                                ;and open up the viewport

                LDY     S_Num                   ;Index into the right location

                LDA     #00                     ;Always a zero...
                STA     Buffer

                LDA     SR_High,Y               ;Setup the starting address
                STA     Buffer+1                ;High byte ...

                LDA     SR_XB,Y
                STA     X_Byte                  ;And X_Byte ...

                SOS     D_Control, Cont_2_List  ;Restore the viewport

                JSR     Rest_Close              ;Restore the console options
                                                ;and close the console driver

                PUSH    Return_Addr     ;Set the return address up

                RTS                     ;Exit back to basic like this

; ---------------------------------------------------------------

                .END
```

Ranntings

by Richard and Lavona Rann

How Long Will Your Floppy Disks Last?

Many of us have owned an Apple /// for six or seven years now. We were early owners, quick to see the ///'s greatness. Others of us have come to the Apple /// in the past couple years or less and do not have that long term relationship. Whether we have owned our /// from that very first day or have only gotten the /// recently, we all have one thing in common.

We use magnetic media, diskettes or hard disks, to store data that is important to us. Beyond the data, if you are a long time owner like we are, it is likely that you have a collection of programs tucked away on the shelf. Many of these old pieces of software were purchased for a special purpose, and again, if you are like us, they have not been looked at for a long time.

This situation of having stored quite a bit of data and software is something encountered by most /// users. In fact, it is common to nearly all who use computers.

What brought this particular subject up at this time, was a conversation we had recently with an Apple Systems Engineer. This particular individual is a long time Apple employee who's job it is not only to sell new Apples, but to make sure that they are installed and running at the accounts to which they are sold. Therefore, much of his responsibility is service.

When a customer has a problem, he is usually the first person who is called. In that capacity, he has run into a persistent problem many times over the past few months.

While it is true that much of this problem is related to Apple IIs, those Apple IIs are about the same age as many our Apple ///s. The problem is also seriously affecting many Apple /// owners.

He has been running into cases where important data is no longer readable. At first, everyone would assume that the disk had been mistreated in some way, and that is why the data could not be retrieved. But these were large Apple customers. People with offices, computer furniture, and storage cabinets for storing diskettes. In short, places that were set up to correctly handle diskettes.

After several of these incidents over the past year, our acquaintance went back to the technical people at Apple. He gave them several samples of "damaged" diskettes. What the technical staff found was a situation of which, Apple and every other maker of computers is fundamentally aware. The problem is involved in the nature of the magnetic media itself. We would all like to believe that once something has been backed-up to a floppy disk that it is permanent, but that is not the case.

The coating on a diskette is made of a ferrous substance that allows us to change the status its magnetic charge. We purposely manipulate the magnetic charge when we format the disk and then write on it, erase data and rewrite data. Each time we change the contents of the disk, we change the status of the magnetic charge on some part of that disk. In fact, the changeability is the very characteristic that makes magnetic media so useful. In the longer run, it also becomes a drawback.

What happens to the data stored on magnetic media, over time, is the culmination of a number of things that when combined, destroy the integrity of the data. The first factor is inherent in the environment we live in. Our environment is a constant threat to magnetically stored data. Air pollution, for example, is comprised of many things, one of which is tiny charged particles.

The particles flow unseen around us without our even noticing. These particles are capable of affecting the charge of magnetic media. Since that charge represents the "code" we use to store data, it affects our data. Another factor impacting stored data is the Earth's magnetic field.

The Earth's magnetic field is what makes the needle of a compass point north. It has the ability, over time, to adjust the magnetic substance coating a disk to a single polarity (charge status). In other words erase the data on a disk.

The Earth is constantly being bombarded from space by particles and energy that can effect data stored on magnetic media of any sort. For the most part this type of radiation goes unnoticed; still it can and does cause data failures. We all have experienced a change in a bit that changed a single character in a data field; this could easily be caused by a charged particle of solar radiation that struck that one place.

Scientists call this random breakdown of order "entropy." That is otherwise defined as the pressure on any organized system to return to a state of randomness.

Our own actions add to the resistance of the natural world to store data. Our Apple ///s can be partially to blame. Most of us know that any computer produces a magnetic field around it while it is operating. The Apple /// produces quite a bit of radio frequency, RF, energy as it runs. This type of electromagnetic energy can affect magnetic media over a period of time.

The electric motors in any appliance, including things like disk drives, televisions and monitors, produce a magnetic field. Even the electrical wires in the wall have a magnetic field. Twentieth century technology has bathed us in low level electromagnetic energy; it comes in numerous types and from every direction. Over time this can cause the slow but inevitable degeneration of stored data.

Finally, there is the limited life of diskettes themselves. Five years ago the science of development of diskettes was in a much cruder state than it is today. The particles that made up the magnetic coating were larger with less hold on their charge. It is much easier for the data on those old disks to be changed by random factors in our environment, than for data on more recent disk media to be randomly changed.

As any diskette ages, its charge slips slowly away. It becomes more vulnerable to the low level influences around it. No one knows just how long a disk will ultimately be able to hold a charge. We can often still read data written by the very first floppy disks drives, but that was done not much more than ten years ago. Do we have another five years, or fifteen, or possibly twenty-five years before there are so many errors on those early diskettes are totally unreadable?

What can we do? We can do very little to change the world to be more friendly to stored data. We are used to electric lights and we need electricity to run out computers. We can, however, do a few common sense things to protect our data. We shouldn't store important data or programs next to a known source of magnetic energy like a florescent light or an Apple ///.

It would be best to store important things in another room, especially a room with few electronic devices. Few of us can install below ground vaults to store our diskettes, but most of us have places better than next to our computers.

Most important is what scientists say about any organized system, "Energy must be expended to fight entropy." It is the same for us and our store of data and programs. The best thing we can do to keep or disks in good condition is to use them. To keep old data we should maybe once a year read it into our computers and write it back on disk. This refreshes the charge.

The error correcting routines in our computers will restore a limited number of questionable bits as they are read in, and this will often repair problems before they have gone too far to be saved. Every few years, important data should be transferred to new, freshly formatted disks. This will fight the degeneration of the media itself

As we write this, we are conscious of the disk boxes on the shelves around our ///. Just a rough count tells us that there are over 400 disks in storage there. Are we going to take our own advice? I'd like to think that we will. There are many important things there, tax records, old articles from four or five years ago; things that may be important to us again.

To keep them we must fight the entropy of our own system by expending the energy to refresh those records. It's not a task we like to think about, but it is an act of faith in our ///s, and a way of helping it keep its value. ///

The 512K Upgrade Kits

by Bob Consorti

A Look At An Inexpensive Way To Upgrade To 512K

The purpose of this article is to show everyone how easy it is to install and use one of the new ON THREE 512K memory board kits. The instructions below are from the new 512K User's Guide which now contains instructions for installing the 512K kit in your Apple ///.

The 512K memory board comes in a few different configurations. The first is a 512K fully populated (filled) with memory chips. The second is a 512K with 256K of memory filled. The third is a 512K with no memory or 0 (zero) K.

If you ordered a 512K memory expansion board kit that is empty or partially filled with memory, the following instructions will show you how to bring it up to a full 512K. The instructions assume that you have already opened up your Apple /// and removed your current 256K memory board.

Please note that all versions of the 512K come complete with instructions for installation and use, diagnostic and update programs - Everything you need to turn your Apple /// into a 512K powerhouse.

256K 512K Kit

The design of the ON THREE 512K memory board splits up the 512K of memory into two halves. The first 256K is made up of 64K memory chips like those on a regular 256K memory board. The second 256K is made up of eight 256K memory chips.

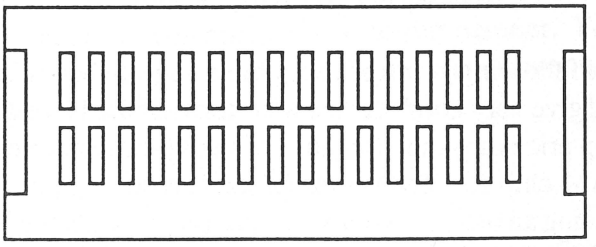
To fill a 512K memory board you need 32 64K memory chips and 8 256K memory chips. If you purchased your 512K kit with 256K of memory, it came filled with the 8 256K ram chips. You need to fill the 32 empty sockets with 32 64K ram chips. Simply use the Extractor tool to gently pull up each of the 32 ram chips on your 256K memory board and place them in position in the empty 32 sockets on the 512K memory board.

Please note that *some* 256K memory boards come with the memory chips soldered directly into the board. If the ram chips are *not* in sockets which are in turn soldered into the memory board, DO NOT attempt to remove the ram chips!! In this instance you must purchase 32 64K ram chips and install those in your 512K.

Make sure each is seated firmly by pressing down on the chips with your thumb. Also, be very careful not to have one or more of the pins on each memory chip bend under as it is being inserted into the sockets on the 512K memory board.

In the diagram below, the chips you need to transfer are displayed.

256K Memory Board With 32 64K Ram Chips



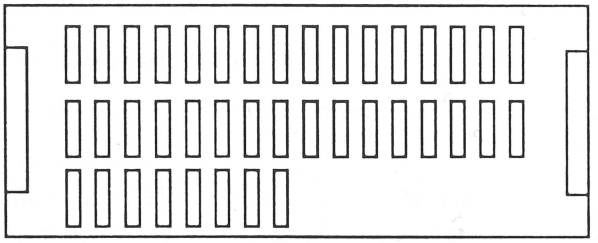
After moving the 64K ram chips to the 512K memory board your 512K is now fully populated with memory and you can continue with installing the 512K.

Zero K 512K Kit

If you purchased a zero K 512K memory expansion board, you should follow the above steps for filling the first 256K of memory. When completed you must then add 8 256K ram chips to the 512K memory board. This is a rather simple process. Simply purchase 8 256K ram chips and insert them into the second 256K ram sockets on the 512K memory board.

The positions of the 8 256K ram chips are displayed below.

512K Memory Board With 32 64K Ram Chips And 8 256K Ram Chips



When you attempt to insert the 8 256K ram chips, note that the pins on brand new memory chips may not fit directly into the sockets. In this instance bend the pins slightly until the fit correctly. You will then be able to insert the 256K ram chips into the sockets on the 512K memory board.

After adding the 256K ram chips to the 512K memory board your 512K is now fully populated with memory and you can continue with the installation.

At this point you install the fully populated 512K into your Apple /// and can start enjoying the benefits of a 512K Apple ///.

List Of Ram Chip Vendors

The following group of companies have all been checked and give very good service with some of the lowest RAM chip prices around. Because of the volatile nature of the RAM chip market, please price each company before placing an order. This will assure that you get the lowest possible price for the needed RAM chips.

If you ordered a zero K 512K kit and have a 256K Apple /// you will need to order 8 256K ram chips. If you have a zero K 512K kit and a 128K Apple /// or a 256K memory board with chips soldered into the board you will need to order 8 256K ram chips and 32 64K ram chips.

Remember that when ordering RAM chips, specify a speed of 150 or 200 nanoseconds. Any manufacturer will do (Hitachi, NEC, TI, OKI etc.) but make sure that you order the correct chip!

As of the time this article was written, 64K ram chips cost approximately $3.50 each and 256K ram chips were running around $12.50. This a set of 8 256K chips will cost about $100 and 32 64K chips will cost about $112.

If you ordered a zero K 512K kit you can get a fully functional 512K Apple /// for around $260! A great deal on any computer. As you can see, turning the 512K zero K or 256K kit into a fully functional 512K memory board is *very* simple. It adds no more than ten or fifteen minutes to the normal installation process and with it you can upgrade your Apple /// to 512K - **the inexpensive way!**

	Company Name	Phone Number
1)	JDR Microdevices	(800) 538-5000
2)	IC Express	(800) 892-8889 or (800) 882-8181 in CA.
3)	Microprocessors Unlimited	(918) 267-4961
4)	Jameco	(415) 592-8097

Continued From Page 24

In most services you exit by typing **BYE** and pressing **RETURN**. You should make sure that you know how to exit - **before** you call the service up.

Sometimes the service you are connected to will lock up and you will not be able to type anything - in other words you won't be able to exit. As the minutes (and dollars) tick off there is a last resort way of exiting and hanging up the modem manually. This should only be done in those rare circumstances that you can't exit by normal methods.

To force your modem to hang up and disconnect from the service, type three plus's (+++) in a row quickly. Wait a second or two and the modem will respond with an **OK**. Then type **ATH**. This is the modem command to hang up. Remember that this should only be used in an emergency when you can't normally disconnect or exit from a service. ///

Using A Modem To Dial Up A Communications Service

by Bob Consorti

The Wonderful World Of Communications

The following instructions are packed with each modem that we sell. Since a number of you may decide to purchase a modem elsewhere you can use these instructions presented below to get started in the wonderful world of communications.

To dial up and connect to a remote communications service such as CompuServe™, the Source™ or a local bulletin board system with The Communications Manager perform the following steps:

1) Make sure the baud and parity settings in your program are correct for the particular service you are calling up. A common one is 1200 baud, 8 bits with no parity. Information regarding setting up your communications program is found in the guide that came with your communications package.

2) Next, in your communications program type **ATDT{number}** and press **RETURN**. In the example below we're using the communications package to issue a command to the modem to dial the number 902-0488. Note that this number is my local CompuServe access number.

   ATDT9020488

   If your telephone only supports pulse dialing you will have to type **ATDP9020488**. In any case, the modem will dial the number for you and in a moment you will hear the other telephone ringing through the modems internal speaker.

   As the phone rings you should see a message on your screen that says "RINGING". Note that some Hayes compatible modems do not have this feature which is called 'Call progress detection'. Most of the newer modems do have this feature. It's a nice feature to ask about, *before* purchasing a modem.

   After one or two rings you will hear a sort or whistling noise and a message saying "CONNECT" will appear on the screen. This is the indication that your modem and the other service are talking together correctly.

3) At this point what you need to do is dependent on the particular service you are talking to. Information regarding each services connect procedure is described in the material that came when you signed up for that service.

Typically you will press **CONTROL C** or just **RETURN**. Some services will have you then enter your account number and password. Then you're in!

That's it. Please refer to the manual that came with the service you will be connecting to for instructions regarding use of that service. All of the communications services that I've seen operate in a similar manner. You enter a command and the service displays a new menu or shows you the information that you requested.

Recording Information

As you use information services such as CompuServe you will at one point or another want to save some of the information that is flashing by on your screen to a disk file for later viewing. Each communications package handles it differently but it generally goes like this.

You tell the comm. package the name of the file you want to send the information to. Then while viewing the information on the screen you can press ⌘ **R** to begin recording. As the information is sent to your screen it is also sent to the file you specified on a disk. Later when you have disconnected from the service you can run a word processing program such as AppleWriter and load up the file where you recorded the information.

One thing that you don't want to do is record the incoming information to a printer. If you do the time you are connected to the service will be increased and thus your costs. When recording to the printer the printer will hold up the modem while it prints resulting in long delays.

When viewing a large amount of information while you are using an information service you may want to temporarily pause the listing. Many services allow you to do this be pressing **CONTROL S** to pause and **CONTROL Q** to resume the listing.

Transmitting Information

If you want to send textual information from a program such as AppleWriter to a communications service you can use the transmit command of your communications package. Each communications package handles it differently, but it generally goes like this.

You tell the communications package the name of the file you want to send. In the Communications Manager you go to the File Setup Menu (Ó T) and setup the name of the transmit file. Then while viewing the information on the screen you can press Ó T to begin transmitting. As the information is sent to your screen it is also sent to the communications service. When the transfer is complete a message will appear on the screen telling you so.

Note that the command described above only allows the transmission of regular ASCII (text) files. If you want to transfer a /// E-Z Pieces word processing or Word Juggler file you must first convert it to an ASCII file. Information regarding this can be found in the programs respective manuals. The AppleWriter program saves its files as regular ASCII files so you can transmit them directly.

Sending / Receiving Programs Or Data Files

The Communications Manager has the ability to transmit and receive programs and data files with the error free XMODEM and Binary II transfer methods. Many of the files on the various information services are stored in this format. To send or receive files using this error free transmission method, please consult the guide that came with the service that you are communicating with for instructions.

Note that while the manuals will talk about using an Apple II, those instructions also apply to the Apple ///. The Binary II transfer protocol was originally designed for the Apple II ProDOS operating system. Because the formats of Apple /// SOS and Apple II ProDOS files are identical, the Binary II transfer protocol also applies to the Apple /// as implemented in the Communications Manager.

Exiting (Hanging Up) From Your Service

When you're finished viewing information or doing whatever else you do on a computer service you will want to exit. Whatever you do, **"DON'T HANG-UP OR SIMPLY TURN OFF THE MODEM"**. You should always exit from a service the way the service wants you to. Many services such as CompuServe will charge you for a full five minutes or so after you hang up the phone if you don't exit properly.

Continued On Page 22

AUTHORS WANTED!

We are accepting article submissions for review and possible publication from readers who are willing to share their Apple /// knowledge and experiences with others. If you would like to contribute to the pages of ON THREE Magazine, here's your chance! We're looking for articles of general interest, short programs, tutorials, hints on getting things done easier & more efficiently, games and anecdotes. Anything goes!

Just send your article or program on diskette to ON THREE. If it's chosen for publication we will check it for accuracy and even correct those silly spellling errors. Before you know it, you will become a rich (at least in spirit) and famous author. Your work will be read worldwide, by a captivated & spellbound audience.

For a copy of our author guidelines, please send a self-addressed, stamped envelope to **ON THREE Magazine, 8920 Yolanda Avenue, Northridge, CA 91324**, or give us a call at (818) 701-1355.

Letters To The Editor

by Bob Consorti — Questions and Answers From Your Apple /// Experts

New TrackBall?

Dear Bob,

I noticed in May's ON THREE that StemWriter has been updated to work under Catalyst.

Enclosed are my StemWriter /// disks, Serial #500366 for you to perform your magic with. I appreciate your good work.

Also saw your notice for the hard disk backup project. Add my name to the list of definite purchasers. Anything that will beat Backup ///.

I'm still trying to figure out why the glos file on the AppleWriter /// (i) which came with my 512K upgrade has a bug. When I press the - key on the keypad, the "Print/rogram Commands" screen overprints the console screen at about the 6th line down with the "Left Margin (LM) = 10" line and does not finish the entire screen. A second press of the same - key does the job correctly. Any ideas? Or is there a patch I haven't heard about.

Enjoyed Steve Brineaux's article on TrackBall ///. Somewhere along the line, I picked up a McBall Model 4000 by Data Input Group of Mountain View, California. It seems to work identical to the AppleMouse, and I wonder if it is the same as TrackBall ///. If so, is it possible to obtain the "Trackconsole" driver and will it allow me to use McBall (or AppleMouse) with any other application under Catalyst/DesktopManager?

Also, please send instructions for installing StemWriter under Catalyst. Thanks again for your help.

Sincerely yours,

Roger F. Suchyta, M.D.
Salt Lake City, Utah

Dear Dr. Suchyta,

By now you should have your StemWriter disks updated and back in your hands. This letter was separated from your disks so if you still don't have them by the time you get this letter give me a call as I imagine they might have been lost on this end.

We will add you onto our list of people wanting the Go Back program and will notify you as soon as it is available.

The problem you are having with the AppleWriter should not be happening. Try deleting the printer definition file on the AppleWriter disk and replacing it with one from a different disk. The next time you startup AppleWriter the problem should go away.

The TrackBall driver works only with the specific hardware found in the TrackBall interface card. If that McBall works just like an Apple //e mouse and conforms to the specifications of the Apple //e mouse card than you could use that piece of hardware in your ///. Programs like Draw ON /// and others that accept a standard Apple //e mouse would be able to recognize it.

My guess is that it's a trackball like device that works on the Mac or Apple IIGS connected to the ADB(Apple Desktop Bus) port. Those devices won't work when attached to the Apple //e mouse card and thus won't work on the Apple ///.

To install StemWriter under Catalyst, follow the instructions for installing a standard interpreter program. Basically you just create a subdirectory on your hard disk (in the CATALYST subdirectory is okay) and copy all of the files from the StemWriter disk into this subdirectory except the SOS.DRIVER and SOS.KERNEL.

Additionally, if you want to print with StemWriter one of your printer drivers must be named .PRINTER or .PRINTER1 - 4.

Draw ON Printer Support

Dear Bob,

I am very disappointed at how few have ordered the /// E-Z Pieces spelling checker. I think it was only a few days after you announced availability I sent in my order.

I am ready to order "GoBack" and if you want an up front deposit, I'll gladly send that at your request.

Do you have any plans to do a HP LaserJet Series II version of Draw ON ///?

Thanks for your support,

John C. Laughlin
Honolulu, Hawaii

Dear Mr. Laughlin,

It is very frustrating that more people have not ordered StemSpeller. Perhaps the numbers will increase over the next six months but at this point we aren't counting on it.

Even though the initial numbers look very bad, we're probably going to do an Apple /// version of Go Back. However small, it's something and at this point in time any sales will help!

To be perfectly honest there won't be any further improvements/changes to the Draw ON /// program. Mel is very busy with other projects and it would take to long to go back to add direct support from the program for a different printer. Your best bet is to send the specs. for that printer in and I think he can add printer support in the Graphics Manager program much easier than in Draw ON ///.

A Quicker Way...

Dear ON THREE:

I read with interest the article by Dan Martin in the June, 1988 issue. I also use the combination of Business Graphics, /// E-Z Pieces and CIS (CompuServe Information System) to periodically evaluate and track stocks and mutual funds. Mr. Martin appears to take an unnecessary step in transferring DIF files to Business Graphics.

My version of Business Graphics will directly read the DIF files without the need for making a Pascal Text file. This eliminates the step in which Mr. Martin uses the "TEXTMAKER" program. After making the appropriate DIF file using /// E-Z Pieces, the INTERCHANGE command is then used.

When asked the file to read from, simply add a period (".") after the name of the DIF file. It is necessary to have the CHANGE.CODE program on the same disk as the DIF file. The end result is a point file which can be used directly by Business Graphics.

Mr. Martin also indicates that Business Graphics mandates the use of the Silentype printer. However, one can have the disk modified ("PIK'd") for other printers. I believe that Sun Remarketing still offers the Business Graphics Printer Interface Kits. I use it with an Epson MX-100. However, if I wish to scale or "polish-up" a graph before printing, I use either the Graphics Manager or Draw ON /// and print it directly from those programs.

I find Business Graphics to be an extremely useful and flexible program but one that requires a significant investment in training to utilize all of its capabilities.

Sincerely,

Edward B. Ash
Marina del Rey, California

Dear Mr. Ash,

Thank you for your letter regarding the Business Graphics / DIF transfers. I'm sure our readers will enjoy your time saving thoughts.

AppleWriter 4.0 Info

Dear ON THREE:

I have a comment and a question.

I recently purchased your Graphics Manager for my Apple /// and the updated version (4.0) of AppleWriter. At first I had a little difficulty installing the Graphics Manager in my Desktop Manager, but after a few mistakes I finally got it right. It would help greatly if the installation instructions were written more like the previous ones for the earlier Desktop modules with a step by step format.

Now for the question. Is there some kind of documentation that goes with the updated version of AppleWriter ///? I at least need something to point out the updated features and explain how to use them. In addition, can I make a new Selector Start disk with the AppleWriter /// version 4.0 disk? If so, how? I have tried to do so but when I boot it I get a System Failure.

I would appreciate any help you can give me. I would also like to take this opportunity to say that if you do develop a simpler, more reliable backup program for the Apple /// I would purchase one immediately.

Sincerely,

John M. Duncan
Salem, Illinois

Dear Mr. Duncan,

I'm sorry over the confusion with the installation instructions. That guide was written quite some time ago and when we get some free time we will revise it.

There is a file called **IMPORTANT.NOTE** on the AppleWriter disk. It contains a list of the differences. The two main ones are:

1) The ability to select the column width to use. Go to the SOS Commands menu (**CONTROL O**) and you will find a couple of additional items. One is the display column width. You can change it from the regular 80 column screen, up to 255 columns across. As you type information and reach the right edge of the screen the screen will scroll over to the new column positions.

2) The other new feature is the ability to select the amount of memory that AppleWriter uses. This is normally 64K (or two 32K banks). You can select the number of 32K banks that the program uses. On a 256K machine you can set up to around 200K of work space. On a 512K Apple /// you will have about 450K or so of memory available.

Please note that if you want to use the Spelling Checker option of AppleWriter, the Super AppleWriter has a distinct problem. The speller was written to work out of a specific portion of the Apple ///'s memory. When using the new AppleWriter it takes up all of the available memory and there isn't any left over for the speller program.

If you are using a 512K Apple /// you will be able to select a 128K (4 32K bank) work space for AppleWriter and still be able to invoke the Speller. However, on a 256K machine or one with The Desktop Manager installed, you will have to limit AppleWriter to 64K or so of memory in order to use the spelling checker.

The instructions in the Selector manual are for the regular copy-protected version of AppleWriter. The new AppleWriter 4.0 is *not* copy-protected and can be installed per the instructions in the Selector manual for installing a standard interpreter program.

Thanks for the support with regard to the new backup program. As soon as GoBack is available we will notify everyone.

Micro Who?

Dear ON THREE,

Enclosed please find a disk with a game on it. I have some interest in programming my Apple /// and hope you will enjoy the game.

I am interested in getting the Titan ///+//e emulation board and a game card. I'd also like to get a Micro-Sci A143 560K drive, if you still have them. It's just a bigger driver, right? Acts like the external /// drive, but uses the compact disks and holds more blocks??? Can I hook it up as drive 2 or drive 3?

What is the 65C802 chip? Does it replace the 6502 chip? What happens if I buy it and put it in my computer? Please send me some information on this if possible.

Sincerely,

Jerise Newton
Pollock Pines, California

Dear Jerise,

I like your Alphazen game. Would you like us to publish it? If so, please send a copy of the program source code on disk for me to review. If it's a reasonable size we could probably put it in the magazine.

The Titan ///+//e boards are a really nice addition to your Apple ///. The Game Card has not been manufactured for some time now. However, with the recent upsurge in Apple II interest by /// owners, we're looking at bringing the product back. If you get the Titan boards you *don't* need a Game Card as the ///+//e has an Apple II joystick port built-in.

The Micro-Sci A143 drive has also been out of production for some years now. It was a very nice drive in that you could directly attach it to the back of the Apple /// without an interface card.

If you want expanded storage for regular use or for backups, I'd suggest an 800K 3.5 inch drive. Either the UniDisk ///.5 or the new CPS 3.5 drives will work nicely on the Apple ///. The nice part about it is that all existing Apple /// programs can use those new drives.

The 65C802 chip does simply replace the ///'s built-in 6502 microprocessor. All existing Apple /// programs will never notice the difference as it looks exactly like the regular 6502 to those older programs. However, since

Continued On Page 28

Continued From Page 27

there in all likelihood *isn't* going to be any new programs written that take advantage of the new capabilities in the 65C802 chip, I recommend against getting one and putting it in your Apple ///.

We've had a severe rejection of the 65C802 chips in Apple ///'s, it's been as bad as 75% of the ///'s won't work with the new microprocessor. It's some unfortunate timing problem that we can't track down. In any case, I look forward to your submission.

MOVING?

If you are in the process of moving, please give us a ***minimum*** of one months notice. Because the magazine must be printed and mailed prior to the date on the cover and because it takes us a short while to update our records, tell us before you are moving. Doing so will assure that your copy of ON THREE won't be lost in the mail or get to you late.

DeClassifieds

FOR SALE: Apple /// accounting software. State of The Art. $300 or best offer. M-F (818) 767-9590.

FOR SALE: Apple /// 256K, monitor, wide carriage daisy wheel printer, lots of extra software with manuals, very low usage. $950 plus shipping. (817) 261-2804.

FOR SALE: Business Graphics, Softcard ///, FX-80 printer with parallel card, PFS, ProFile Owner's Manual, Backup ///, VisiCalc ///. Call Robert Lamison at (904) 878-4688

FOR SALE: 2 Apple ///'s (1-256K, 1-512K), 4 external drives (1-Apple ///, 3 MicroSci A143), 2 monitors, 3 ProFile 5 MG hard disks, 2 monitors, 2 Silentypes, 1 Apple Dot Matrix Printer, 1 Apple Plotter. Lots of software and manuals. No offer for all or any part too ridiculous to be considered. Moving to Macintosh. Call Mert Corn (718) 979-4000.

FOR SALE: Apple ///plus with 256K, XComp 16 megabyte hard disk, Macintosh XL's, Mac SE, LaserWriter, and a variety of Apple /// software. Call for details. Can also get new Mac's & LaserWriter's for low prices. Call Bob at (818) 701-1355.

DeClassifieds Work!

Been searching for hard to find hardware or software? Need to sell some excess equipment? Try an inexpensive DeClassified ad! Our readers tell us that they really get results! Place your ad by phone or mail. All copy must reach us 45 days prior to publication date, e.g. October 1 for the November/December issue.

Rates $2 per word $25 minimum.
Subscriber discount $1 per word, $25 minimum.

Mail your copy with payment to:

ON THREE DeClassifieds
8920 Yolanda Avenue
Northridge, CA 91324
Or Phone (818) 701-1355

How Would You Like A Fast, Reliable And Easy To Use Hard Disk Backup Program?

After years of complaints about Backup /// being too slow and in many cases losing people's important information we've decided to do something about it. We know that many people *never* backup their hard disk because Backup /// is soooo slow or because they've had a problem with it losing information.

A while back we started work on a product we call Go Back. If you're sick and tired of the time it takes to backup your hard disk with Backup /// or the System Utilities, Go Back is what you need.

Go Back is written entirely in assembly language for speed, is four to five times faster than Backup /// and much more reliable. We guarantee it!

We were about half-way finished with the project when we stepped back, took a look at the potential market and asked ourselves, "Will there be enough sales to warrant finishing Go Back?"

If there's enough interest in Go Back we will complete it. Now we **don't** want you to send in your money, or give us a credit card number. All we want is your assurance that you will buy Go Back if we spend the effort to finish it. Actually, we want the assurance of at least two hundred people.

If you want a reliable, very quick and inexpensive (around $50) program for backing up your hard disk, drop us a note or give us a call at (818) 701-1355. Just leave your name and address so we can notify you if and when the project is finished.

Next Time In ON THREE:

Desktop Publishing With The Apple ///

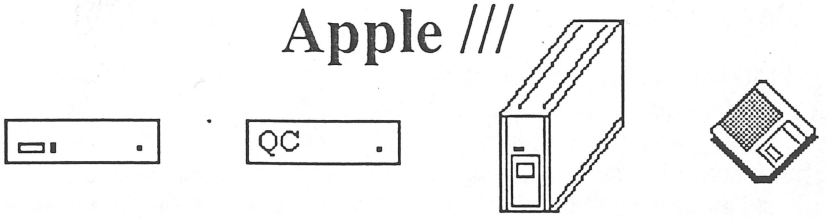
Brightening Up Basic

Three Attack

110 <-> 220 Apple /// Conversions (Really, I'll put it in this time!)

And our regular mix of letters and other information to help you get the most out of your Apple ///.

Selector /// - The Best Program Switcher For The Apple ///



Do you use a ProFile, Quark, Sider, UniDisk ///.5 or CPS disk drive?

Are you sick of the time it takes loading each of your different programs? Tired of searching though stacks and stacks of floppy disks for a particular program? Do you have a hard disk or a large capacity floppy disk drive? If so you can cure all of these headaches and more with the purchase of Selector ///!

Selector allows you to place all of the programs listed below (and many more) on your hard disk or large capacity floppy disk drive. Instead of waiting and waiting for a particular program to boot up, you simply boot your Apple /// with the Selector disk. When you want to run a program, just select it from the easy to use Selector menu by highlighting it and pressing RETURN. In a flash the program is loaded from the hard disk and you can be using it in seconds, not the minutes it usually takes.

Selector /// works perfectly with the Apple 5 and 10 megabyte ProFile, the Quark QC 10 and 20 drives, the Sider's, the UniDisk ///.5 and the new CPS 800K drive. It even has an automatic installation program that places Selector on your hard disk or other floppy drive.

Selector can speed up using your Apple /// and make your use of the Apple /// more productive and beneficial. It even works with programs that *you* write in Basic, Pascal or assembly language! Selector can hold hundreds of programs with it's hierarchical menus while other selectors only allow you 12 or 24 programs.

Selector is the only switching utility for the Apple /// that is completely compatible with virtually every disk drive and program for the Apple ///. Order Selector /// today and become a real power user. Special offer! From now until October 31st you can get Selector /// for only $69.95 plus $7 s/h. That's **30% off!**

Access ///	BPI Accounting	Fruit Machine	PFS: File*
Access 3270	Business Basic	Go Back	PFS: Report*
Advanced VisiCalc	Business Graphics	Graphics Manager	Quick File ///
Apple II Emulation	Cobol	Graph'n'Calc	Script ///
Apple File ///	Crossword Scrambler	Haba Merge	Senior Analyst ///
Apple Speller ///	Draw ON ///	Keystroke Data Base*	Side Print ///
Apple /// Pascal	Comm. Manager	Keystroke Report Generator*	StemSpeller
Apple Writer ///	Desktop Manager	Lazarus ///	StemWriter
Backup ///	EasyTerm	Mr. Sandman	VisiCalc ///
BattleFleet	Fortran ///	Multiplan	/// E-Z Pieces

The programs listed above with an asterisk after them are copy protected and require the 'key disk' to be inserted into the built-in drive when you select it from the Selector menu. After the program comes up you can remove that disk. Uncopyproteced versions of PFS: File and PFS: Report are available that work under Selector without the need for a key disk. Likewise an unproteced version of Word Juggler is available for only $15 plus $2 for shipping and handling.

Fantastic New Products For Your Apple ///

Seikosha Color Printer

A few years ago I spent over $1500 on a color printer that I could hook up to my Apple /// and produce great looking color printouts of my Draw ON /// and other graphic images. It was really nice but it was *verrrry* expensive. Today a good color printer costs around $600 but that's still very expensive for most people.

After much research we've found a color printer that works nicely on the Apple ///, produces great looking color printouts and doesn't cost an arm and a leg. The Seikosha color printer comes complete with printer, color ribbon, parallel interface card, driver and a special version of the Color Graphics Manager. This program lets you print your images to the Seikosha anywhere on the page, at any size and in a variety of rotations.

It's a perfect companion to Draw ON /// and the only way to get great looking yet low cost color printouts on the Apple ///. It's ***only*** $259 + $20 s/h.

Note that while the Seikosha is a parallel printer it must be connected to the supplied parallel interface card for it to work properly on the Apple ///. The parallel card included with the printer works with all other parallel printers in addition to the Seikosha. If you don't have a free slot inside your Apple /// you can take out your UPIC or other card and use the Seikosha one instead to drive both your existing & the Seikosha color printer.

Side Print ///™

For years people have been asking if there was a way to print very wide spreadsheets sideways on the Apple ///. Until now there wasn't. But with our new Side Print /// you can print your /// E-Z Pieces, VisiCalc and Advanced VisiCalc spreadsheets sideways on your Apple DMP, ImageWriter, Epson or OkiData printers.

Side Print /// is available today for only $29.95 + $3 s/h.

The Desktop Manager/Combo Pack

The Desktop Manager (TDM) is a utility that allows you to use desk accessories like those on the Macintosh™. They are utility programs that you use within other programs. You can be using /// E-Z Pieces or any other Apple /// program and a simple keystroke will freeze your program and display the TDM menu. Simply highlight the accessory you want, press RETURN and in a moment it will be up and running. When you're finished, press ESCAPE and you'll be right back in your program.

With TDM you can throw away your paper, pen, calculator, appointment book and more! We've included a number of standard features to enhance your productivity. TDM has a built in: Note Pad, Appointment Calendar, Calculator, Pickup & Paste, Help screens and more.

TDM is also expandable so you can design the system that best suits your own needs. Add new accessories such as macros, disk utilities, communications, games & more without using any additional memory inside your ///.

The Desktop Manager requires an external disk drive of any capacity and a 256K or 512K Apple ///. TDM uses about 32K of memory. A hard disk or large capacity floppy disk drive is highly recommended. Works with Selector, Catalyst and ALL other Apple /// programs. Regularly $129.95, for a limited time now only $69+$6!

But wait!!! It gets even better. Until the end of this sale (October 31, 1988) buy The Desktop Manager main accessories described above and get any two add-on accessories for only $30 extra! That's right for only $99 you can get TDM with the Disk Manager and Macro Manager, or the Communications Manager and a Super Accessories Disk. Any combination for only $99+$9 s/h.

Order yours today as this sale may never be repeated!

ON THREE, Inc.
Your Apple /// Experts
P.O. Box 3452
Kirkland, Washington 98033

Fantastic Fall Clearance Sale!
Check Out The Lowest Prices Ever Inside.