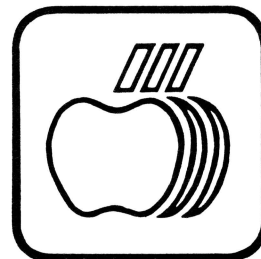


open apple gazette



Ninth Edition

Volume 2, Number 3

Feb./Mar. 1984

Report On COMDEX	2
Inside Mail List Manager - Part I	9
Interfacing Apples and Oranges	20
Creating Word Juggler Mailing Lists From Quick File ///	26

REVIEWS:

BPI General Accounting Package	18
Apple File ///	22
Apple Speller ///	30

STEMS AND SEEDS:

Apple Writer ///	Upgrade	8
Does Apple Have Your Warranty Card?		8
Do You Want To Network?		19
Reduction In Cost Of Upgrade From 128K to 256K		21

Business BASIC:

Printer Control Program	16
Beginning Business BASIC - Lesson 4	28

Report On

/// /// /// /// COMDEX /// /// /// ///

November 28 to December 2 1983
Las Vegas, Nevada

by
Don Morris

I spent Monday just trying to comprehend 1,250,000 square feet of exhibit space filled with computers, peripherals and software vendors. The Las Vegas Convention Center was filled to capacity. Those exhibitors unable to obtain exhibit space on the main floor had to use nearby hotel lobbies and exhibit rooms. As an example of how scarce space was, registration (which last year was in the Convention Center) was in a temporary structure between the Las Vegas Hilton and the Convention Center. With over 83,000 attendees, Comdex became the largest single event held in Las Vegas. A record number of long distance phone calls were placed in a single week out of Las Vegas. This was evidenced by the number of attempts it took me to get a long distance call completed. Needless to say the show was big, much bigger than last year.

A very popular item on the floor at COMDEX were replacement keyboards for the IBM PC and PC jr. These real typewriter keyboards place keys such as the "Return" and "Shift" keys in the standard locations, or rather, locations which those of us who are touch typists are used to.

With the notable exception of Apple, computer manufacturers at COMDEX introducing new computers seemed to be trying to load their keyboards with as many function keys as possible. While they were all detachable, they were assuming the dimensions of small football fields in order to make room for all of the function keys. Being a touch typist I have a personal dislike for any function, control, or command key that requires me to leave the home keys. I seriously doubt that these companies are going in the right direction.

If you consider that a good many of the new computer users (especially those in an office environment where the staff is used to a standard keyboard) will face learning a completely new keyboard, with lots of special keys that are hard to reach, this appears to present a significant barrier to learning to use the machine and at the

minimum lengthen the learning curve. The other limitation with dedicated function keys is that the software and hardware designer must anticipate what will be the most popular functions that users will want.

Frankly, I prefer Apple's approach. Adopt a keyboard layout and shape that is ergonomically designed, that is, universally accepted and designed for ease of use and user comfort, ergonomics being the industry "buzz word" for computer systems created in such a manner. Then, provide the software author with a great selection of special function keys via the Open Apple and Control Keys. Finally, let the end user define them him/herself.

One of the built-in features of the Apple /// is software definable keys, such as the Open Apple Key and the Control key. Software developers are beginning to take advantage of this capability. One example of this type of user definable function keys are the glossaries you can create with Apple Writer. After you define the glossary, by simply pressing Open Apple and another character you can create over 52 special functions. The latest version of Apple Writer, 2.0, lets you include Control characters into your glossary files.

For example, I use both Apple Writer and Word Juggler. I have defined the "."(dot) key on the 10-key numeric pad to print a document in Apple Writer so it is now standardized with Word Juggler. This way I can print in nearly the same way with either word processing program. The specific sequence I have developed for Apple Writer is as follows:

```
[G]?.[P]PR[N]][P]NP]
```

The square brackets around a character represent a CONTROL character, such as [G]. The right square bracket "]" is a carriage return. The use of this is explained on page 53 Volume Two of the new Apple Writer Manual.

The definition of this sequence is the following: [G]?. creates the glossary entry "." on the numeric keypad; [P]PR[N]] prints to the console the ASCII equivalent of [5] on the numeric keypad, which turns off the video output so that the document will be "printed" to the spooler faster; [P]NP] then prints the document to ".spool", which is the printer destination under my print/program menu of Apple Writer.

Another sequence runs the Apple Speller program by pressing one key on the numeric pad.

[K] sequences you can create with Advanced Version Visicalc are another example of user definable function keys. For each model I use with Advanced Version VisiCalc, I define a sequence so that when I press [K] S the model is automatically saved by its pathname and filename. Sure beats entering /SS(Pathname)/(filename) every time you want to save a model.

The Apple /// room at the Aladdin Hotel

The Apple /// Team invited dealers to visit software and peripheral vendors at a meeting room in the Aladdin Hotel. The two days were well attended by dealers. The dealers spent an average of an hour and a half visiting with the vendors.

Here is a look at some of the more interesting programs and peripherals being developed to take advantage of the Apple ///'s features:

Sun Data Products
95 West 100 South
Logan, Utah 84321

(801) 752-7631

Sun Data products include:

LIFESTAF, a program which enables life insurance agents to make sophisticated forecasts of an individual's life insurance requirements based upon the financial information supplied by the insurance buyer. The program enables the agent or other financial advisor to make changes in the information and determine the results with the programs "what if" capability.

PRINTER DRIVER ///. Allows most printers to be used with most printer interface cards designed for the Apple][. The driver can be customized to a particular interface card by changing the configuration block using the System Utilities Program. In addition, included are drivers already configured for the following cards:

Epson APL
Apple][Parallel
Tymac PPC-100
Microtek RV-611C
MPC
Orange Micro Grappler interface
Orange Micro Grappler Plus
Microtek Apple Dumpling - GX
Microtek Dumpling-64 Buffer/spooler
Practical Peripherals Microbuffer]]

This driver also enables you to take full advantage of features in Apple Writer such as underlining and superscripting with your Epson printers. In addition it will work even with obscure and discontinued printers such as the Epson TX 80.

EASYTERM ///. Communication program enabling you to send files to and receive files from a remote computer directly or via modem.

INFONET. An information network software program written for novice to expert computer users. This program was designed for organizations who need a method of electronic communications between themselves and other computer users. This program will allow your Apple /// to act as an electronic bulletin board/message center. The program can accommodate over 1,000 users depending on the amount of room on your disk drive.

PSCREEN. Allows printing of Apple /// text and graphics screens to an Epson or Apple's Silentyper printer. It also comes with a demo program which allows you to print Apple /// Business Graphics screen files.

SOSTRAN. Copys files from Apple][format to Apple /// (SOS) format. Conversions include Applesoft and Integer BASIC to Apple /// Business BASIC. Images of the High Resolution graphics screen (also known as "Foto" or "Picture" files) can be converted between Apple][and Apple /// format.

SOSTRAN can also list the catalog of any DOS 3.3, Pascal, or SOS disk, in case you forget the name of the files you want to transfer. The documentation includes some of the best information I have seen for converting Applesoft programs to Apple /// Business BASIC.

Brock Software
P.O. Box 799
Crystal Lake, IL 60014

(815) 459-4210

Brock Software's **Keystroke** Database is one of the most exciting pieces of software to be introduced for the ///. For those of you who have not seen it, I suggest you visit your dealer for a demonstration. It is a very powerful, easy-to-use database and report generator package.

Keystroke's features include:

- a logical menu structure; (menus appear as boxed windows over the current work screen)
- four user selectable key fields
- access to a second file
- interface ability to Quickfile & PFS
- read and write DIF compatible files
- up to 4096 characters per record
- 74 character maximum field width
- 90 fields per record
- copy data into and from all files

Interfacing data in and out of **Keystroke** to other Apple /// software is probably its most powerful feature. In the past, other software developers and products have required you to reenter your data when moving from one application or data format to another. My personal feeling is that I enter information into the computer ONCE, and then use software to manipulate it.

Keystroke also allows you to create your own function keys. Function keys, as I have previously mentioned, condense into one keystroke what would normally take several.

The Database retails for \$249, the Report Generator for \$149.

Quark Incorporated
2525 West Evans
Suite 220
Denver, CO 80219

(303) 934-2211

Quark is an example of a company concentrating its development efforts on improving their software to take full advantage of specific machines like the Apple /// and][e. This is done in contrast to adapting software to fit every new computer to come down the pike.

Examples of this philosophy are their recently revised **Word Juggler**, **Catalyst 2.0**, **Lexicheck** and **Discourse Spooler**.

Quark's product **Catalyst** is the program which enables you to have all of your application programs (Word Juggler, VisiCalc, Quickfile etc.) loaded onto your profile so you can easily switch between programs without rebooting. Several enhancements have been made to **Catalyst Version 2.0**. Probably the most important is that it is virtually self-installing. It automatically performs the operations to install most of the popular programs, merely prompting you to put various boot diskettes into the inboard drive when necessary.

With **Catalyst 2.0** you choose the program you want to run using the up and down arrow keys and then press RETURN. A few seconds later the application you want is loaded into memory. The program has also been modified to work with the Micro-Sci A143 floppy disk drive. Previously the program required a hard disk drive.

Several software companies, like Brock Software Products, are now including "Quark.Install" with their software to make installation easy. With "Quark.Install", you merely insert into the built-in disk drive the program disks you want to put onto the hard disk and follow the prompts on the screen.

Previously, **Catalyst** users had to add every driver they would be using to the SOS.DRIVER file on the **Catalyst** Boot Diskette. For example, suppose you have installed a graphics program such as Graph'N Calc and a communications program such as Access ///. The RS232 and the graphics driver would be using up some of the memory in your /// even though you have no need for the RS232 driver when you are using Graph 'N Calc.

Likewise you have no need for the graphics driver when you are using your modem with Access ///. Version 2.0 of **Catalyst** gets around this problem with dynamic drivers. These are drivers that are only loaded into memory as you need them for a particular program; the graphics driver for Graph'N Calc and the RS232 driver for Access ///.

Those of you currently using **Catalyst** with VisiCalc will appreciate getting the memory back for your large spreadsheets.

A new driver for profile is also included with **Catalyst 2.0**. It keeps the root directory of your profile in memory, which enables programs such as the Great Plains hard disk accounting system to store and access data faster on the profile.

The new version 1.4 of **Discourse** will allow you to change the order in which your documents are printed out from the spool status. Additionally, the spool status will show you from which program your document originated, i.e., VisiCalc, or Apple Writer. Word Juggler users will be able to spool documents four times as fast with the new version 1.4.

A very user-friendly feature of this new version of **Discourse** is that it is completely self-installing. You simply boot the **Discourse** diskette and follow the prompts to insert your boot diskettes such as **Catalyst**. The program then deletes your old spool and spool status drivers and replaces them with the new versions. Sure beats fumbling around with the System Configuration Program!

In the event of power failure, **Discourse 1.4** will recover unprinted documents and requeue them for printing. Incidentally, **Discourse** was the first piece of software to make extensive use of the interrupt capabilities built into the Apple ///.

Lexicheck has been updated to include a "GUESS" feature in which several words are presented in a window for selection as the correct word. An inverse bar you can move selects the proper spelling for the word you want to replace.

Microsoft
10700 Northrup Way
Bellevue, WA 98004

(206) 828-8080

Micro Soft recently introduced their spreadsheet **Multiplan**, and yes, it runs native on the ///. (A program runs "native" when it uses all of the power of the /// and does not run in emulation mode or require a CP/M card (Softcard ///)).

Haba Systems
15154 Stagg Street
Van Nuys, CA 91405
(213) 901-8828

/// **EZ Pieces**, Haba's new integrated piece of software and its Apple][version, Appleworks, are absolutely fantastic. Those of you familiar with Quick File will recognize the commands and ease of operation. The program was written by Rupert Lisner, who wrote Quick File.

Some of the features are as follows:

-/// **E-Z Pieces** is a fully integrated, easy to use Word Processing, Spreadsheet, and Data Base Management System designed exclusively for the Apple /// Computer. Once you've learned one part of /// **E-Z Pieces**, you've nearly learned them all. All modules use the same commands which are displayed on "HELP" screens when you need them. All commands are described in English, not in single character codes.

-With /// **E-Z Pieces**, word processing is truly easy, and there are no surprises. Text will be printed exactly as you've typed it. View imbedded print commands at the touch of a key. Insert, delete, and move letters, words, lines, sentences, and paragraphs by highlighting the appropriate text, and using simple commands. Find and replace words (or word strings) quickly and easily.

-/// **E-Z Pieces** contains all of the features of Quick File and then adds some more. Reports can be in either label or table format. Each record can contain up to 30 categories. Sort time is dramatically faster than other systems. (Approximately 15 seconds for 1500 records!) Plus, you can cut and paste information into your Word Processing files because /// **E-Z Pieces** is fully integrated!

-The spreadsheet capability of /// **E-Z Pieces**, up to 999 rows by 126 columns, handles the largest of spreadsheets. To enter larger headings or numbers, change column widths quickly with simple keystrokes. Automatically repeat formulas for relative columns or rows. It's EZ and fast too. This spreadsheet reads files up to 8 times faster than VisiCalc!

I have been using /// **E-Z Pieces** for

several weeks without a manual and find it to be a very powerful and easy-to-use program. The program, with its extensive use of help screens, enables you to start being productive without extensive reference to a manual. In my opinion it is even easier than VisiCalc or Quick File.

For ease of use the spreadsheet portion of the program is way ahead of VisiCalc. For example, when you start a spreadsheet, you are prompted for the disk drive, pathname and filename under which the spreadsheet will be stored. Then, when you want to save it, just press "OPEN APPLE S". This sure beats entering "/SS(DRIVE/PATHNAME/FILENAME)" every time you want to save a spreadsheet template.

Another plus for this program is that it can read the files created by the Apple][version and vice versa. This is because the Apple][version was written using ProDOS, the new operating system for the Apple][and //e. ProDOS will enable those of you with a /// in the office and a][at home to work on the same files without making any DOS to SOS conversions.

Speaking of **VisiCalc**, the following best describes the attitude currently held by VisiCorp towards the ///.

I stopped by the VisiCorp booth at the Fall Comdex show in Las Vegas.

Quick question:

Are you doing anything for the Apple ///?

Quick answer:

No!

Quick Exit.

Too bad. Their product **VisiCalc** was the reason I purchased my /// over three years ago. At that time, VisiCalc was THE spreadsheet for microcomputers PERIOD. Prior to Word Juggler and Apple Writer being available, I even used it as a text editor and developed a way to do form letters. Advanced Version VisiCalc was a definite improvement, yet it failed to cure one of VisiCalc's primary downfalls, namely such SLL00WW (how about EIGHT minutes to load a big template) loading and saving of files.

Hard Disk Drives

Apple is apparently getting out of the disk drive business. The Apple /// division is seriously considering hard disk drives from outside manufacturers that will work with the Apple ///. Some possible candidates:

Mountain Computer
300 El Pueblo Road
Scotts Valley, CA 95066

(408) 438-6650

Mountain has a line of hard disk drives compatible with the Apple ///. They are available in either 5, 10, 15, or 20 Megabyte capacities.

They also announced a 1/4" tape drive back-up for the profile. This is interesting in that it has an Apple part number, which is another indication that Apple is at least looking to supplement their line of disk drives with those of outside manufacturers.

Santa Clara Systems
1860 Hartog Drive
San Jose, CA 95131

(408) 374-6972

SCS manufactures hard disk systems with storage capacities ranging from 5 to 120 megabytes. They offer back-up utilizing either high density eight inch floppies or removable cartridges. They have some networking products, but nothing for Apple products at the present time.

I.C.E.
c/o Space Coast Systems
301 S. Washington Ave.
Titusville, FL 32796

(305) 268-0872

I.C.E. features a tape back-up for hard disk drives, their own as well as Apple's profile. Their literature indicates that the tape backup runs three times faster when backing up their drives (90 inches per second), than with profile (30 inches per second). Their data transfer rate is evidently faster. I will have one of these drives for evaluation, and will compare it with profile for speed of data transfer in loading programs such as System Utilities and , which are large pascal programs and

take several seconds to load.

They are currently working with Quark to make their drives compatible with **Catalyst**.

Davong Systems Inc.
121 Humboldt Court
Sunnyvale, CA 94089

(408) 734-4900

In addition to a line of hard disk drives which are compatible with the Apple ///, **Davong** is considering adapting their networking to work with Apples.

It is encouraging to see these manufacturers of hard disk drives taking an interest in the Apple ///. Their enthusiasm has been sparked by the Apple /// team which did not exist six months ago. It is a welcome change.

One of the disk drive manufacturers said they had benchmarked the /// with the IBM PC and found what most of us already know. The /// is faster in most operations.

When the Apple /// will address a 16 Megabyte file, to offer it with a 5 Megabyte hard disk greatly limits its capabilities. Generally, you can purchase a 10 Megabyte hard disk drive from one of these manufacturers for only a couple of hundred dollars more than the cost of a profile.

Printer Cards

Wesper Micro Systems
14321 New Myford Road
Tustin, CA 92680

(714) 730-6250

Wesper Micro Systems introduced their BPO /// serial parallel buffered printer card for the Apple ///. This card enables you to have a separate serial port for your printer, freeing your RS232 port for your modem. This eliminates switching cables whenever you want to change from printing to telecommunications.

The BPO /// comes with 8K (over 8,000 characters) of buffering, which you can increase to 128K.

Special software selectable features include: software selection of Serial or Parallel printer; data compression to enhance buffering capacity; text formatting; auto line feed generation, and multiple copies ability.

Also available is an external switch which enables you to make multiple copies, pause, and reset the buffer.

Computer Books

The Osborne/McGraw-Hill Apple /// book sells for \$17.95. It should be more correctly titled "Apple /// Business BASIC" because over half of the book is devoted to Business BASIC. If you are looking for a tutorial on Business BASIC, this book is it. It has very clear, concise explanations and examples.

Another recently released book for the Apple ///, Using Apple Business Computers, by Kenniston W. Lord, has an emphasis on Business BASIC from a beginners' point of view. The book evolves into a series of useful programs for the /// in a small business environment.

The program listings include:

CHART:	used to build the chart of accounts
CUSTOMER:	customer file for accounts receivable
VENDOR:	vendor file for accounts payable
PRODUCT:	rudimentary inventory control mechanism
RECEIVABLE:	Provides initial entry, invoice preparation, posting of account records statements and aging
INCOME:	produces an income statement using prestored totals from the other files
BALANCE:	produces the balance sheet

Last, but not least, **DOT RODENT**.

Well, that's what I call it. The /// Mouse, (which is the same as the //e Mouse), was demonstrated with a soon-to-be-released graphics package. This program is extremely user-friendly. Within minutes first time users were creating all kinds of drawings

and paintings on the RGB color monitor. This program will be released some time in the first quarter of this year.

With this program you can zoom in to an individual pixel and change its color.

BASIC programers will be interested to note that the program is written in Business BASIC. When BASIC booted up with this program the version number was 1.23. Obviously Apple is still making improvements and enhancements to the language.

Quark and several other software vendors are modifying their software to use DOT RODENT.

Why do I call it DOT RODENT? As all Apple /// owners know, you have to precede all driver names with the period or dot and somehow dot mouse just doesn't sound appropriate for a driver name.

In conclusion, while there were a lot of IBM clones (Monkey see:Monkey do) introduced at Comdex, lots of exciting things are happening for our favorite Personal Computer. If Comdex gets much bigger it will become almost impossible to really cover adequately.

///

Dear Don,

I tremendously enjoy your publication and look forward to attending your meetings as my schedule permits.

Please inform your readers that the **State Of The Art General Ledger Package** will not work with Quark's Catalyst. I spoke to someone at Quark who said that the State Of The Art people have been uncooperative in providing them with the information needed to make a fit.

W. Randall Revell
President, Context Trainings Corp.
San Francisco, CA.

///

Stems and Seeds:

Does Apple Have Your Warranty Card?

If you haven't received the mailings from Dave Fradin, manager of the newly formed Apple /// Personal Business Systems Group at Apple, chances are Apple doesn't have your warranty card. Here are some of the announcements you've been missing:

- **Catalyst, Version 2.0:** program selection software upgrade;
- Apple Serial Card ///: adds additional serial port to your ///;
- Apple Color Plotter: four-pen color plotter for pictures, charts or words;
- New phone card, **HabaDex**, from Haba Systems;
- Apple /// Plus: upgrade your present /// to include more features;
- **Keystroke** Data Base and Report Generator from Brock Software;
- **/// EZ Pieces** Data Base/Spreadsheet/Word Processor from Haba Systems.

To remedy the situation, simply send a quick note to Ms. Linda Maybrouk, Apple Computer, 20525 Mariani Avenue, Cupertino, CA 95014. Be sure to include the serial number of your Apple ///.

Those of you who have purchased Quark Engineering products should be receiving upgrade announcements as long as you mailed back the product warranty card. If you didn't, be sure to do so today since the majority of Quark's products have been upgraded. Mail your card to Quark Inc., 2525 West Evans, Suite 220, Denver, CO 80219. (303) 934-2211.

Quark is an excellent example of a software company which is continually upgrading their products.

Apple Writer /// Upgrade

Version 2.0 of Apple Writer /// has been announced by Apple Computer. In order to get your copy upgraded, send your Master disk along with the cover of your manual and \$100 to: Apple Writer Upgrade, P.O. Box 106, Half Moon Bay, CA 94019. Please make all checks payable to Apple Computer and allow 6-8 weeks for delivery. Along with the upgrade you will receive three revised manuals which include detailed information on WPL.

Inside Apple /// Mail List Manager - Part I

by
Allan M. Bloom, PhD, CDP.

The "Apple /// Mail List Manager" (MLM) is a fine product for those who wish to enter, maintain, and print mailing list information. Apple Computer is not far off the mark when touting MLM as a general purpose list processing system for all one's personal and business needs. MLM is a neat data entry/update system, but it falls off for those who want to do any more with the resulting data base than print labels. What's holding it back? One should certainly be able to do more with MLM than Apple provides in its program product.

To do anything useful with a data base, you must know its data structures and its encoding algorithms. It takes more than access to gain control of a data base. A data system is remarkably secure if it is wierd enough. Apple's "Mail List Manager" qualifies admirably on that score.

This article uncovers MLM's data structures and lays open its encoding algorithms. It represents an effort roughly equivalent to solving a quarter-acre New York Times Sunday crossword puzzle. Had it not been for an ill-advised promise to my wife to convert her mailing list to MLM format, I'd have given up quite early. MLM is a bear, folks.

The following sections describe MLM's file and data structures. I cannot and do not explain. There is a bunch of "cute" in what is described. I hate "cute." Nonetheless, the information is valuable. Whether done with "cute" programming or not, MLM is an outstanding product. I know very few data management systems of such power and ease of use. The descriptions that follow are summarized as a Pascal library unit that allow applications systems programmers to free MLM to be a management information tool, to be as powerful as it can be.

MLM File Structure

Apple's "Mail List Manager" Version 1.0 is a diskette-oriented data base. An MLM file takes up an entire standard diskette with its two files: MLMGR.INDX and MLMGR.DATA. The INDX file consists of 31 blocks (of 512 bytes each) of data. The first block is a "header" record, consisting of file-related information. The second block onward is the

actual "index" to the data records in the DATA file, with 32 sixteen-byte index entries per block. The more mathematically oriented will note that 30 blocks of 32 entries each totals the 960 that is the maximum number of records that an MLM file can hold.

The DATA file actually holds the MLM data records, with four 128-byte entries per block. The 960 maximum entries take up 240 blocks. Both files, then, consume 271 of a possible 280 blocks on a diskette. SOS overhead takes up the remaining blocks, and a catalog listing of an MLM diskette will show zero available blocks.

The MLM diskette is so crammed full, in fact, that it cannot be copied on a file-by-file basis. The file copy facilities of the "System Utilities" program or of PASCAL's "Filer" will copy the MLM files, but MLM will not recognize the result as valid. MLM will balk at a disk that has been file-by-file copied, with the message "Error -- a good mailing record diskette is not seated properly in that drive." Try it. You won't like it. A catalog listing of the copied diskette will show that the "Size" of each file is one block larger than on the original (32 and 241 blocks, respectively). MLM gets indigestion on those differences. Always device-copy an MLM diskette to keep everything properly scrunched. Or write it a single block at a time.

Accessing MLM Data Records

The individual records of the DATA file cannot be accessed meaningfully by themselves. They must be taken in the context of the INDX file entries. A datum in the INDX header record notes the number of active records ("n"). Only the first "n" INDX entries count. Those entries are physically sequenced by the most recent MLM sort request. Each entry points to a DATA file record, from 1 to 960. Only those DATA records pointed to by an INDX entry are active. When a mailing list record is "deleted," MLM does not actually delete the DATA record. The INDX file simply no longer points to it. The DATA space on the disk is noted as available for future additions.

If a mailing list file has had "m" deletions posted to it, without adding any new records, the "n" active records are distributed through the first "n + m" records of the DATA file. Without the INDX file pointers, there is no way to tell which "n" DATA records are active. Figure 1

Apple /// Mail List Manager 1.0 File Structure

<u>INDX File Records</u>	<u>DATA File Records</u>
Header Record	Data Record 001 Name = ZOMBIE
Record Count = 004	Data Record 002 Name = CLARK
Index Record 001 Points to 004	Data Record 003 Name = SMITH
Index Record 002 Points to 007	Data Record 004 Name = ABLE
Index Record 003 Points to 005	Data Record 005 Name = JONES
Index Record 004 Points to 001	Data Record 006 Name = SMART
	Data Record 007 Name = DUMB

1. The INDX file contains one 512-byte "header" record. Among its many data items is a count of the number of active data records in the file.
2. Following the header in the INDX file are 960 "index" records of 16 bytes each. In the above sample file, only four data records are active, so only the first four index records contain meaningful information.

The above sample MLM file has been sorted by "Name," so the four index records are sequenced to point to the active data in alphabetic order.

Data records 002 (CLARK), 003 (SMITH), and 006 (SMART) are not referenced.

3. The DATA file contains 960 records of 128 bytes each. In the above sample, seven data records contain information. Records 002, 003, and 005 are inactive. There is no way to know this without information from INDX.

Figure 1

diagrams the situation.

Accessing the three record types of a MLM file is a bit less than straightforward. The INDX file header record must be read as a single 512K block, and it should probably be kept in memory (if for no other reason than to know how many records the file contains). The least convoluted way of accessing the rest of the INDX file begins with closing it after having read the header block, then re-opening it to read the subsequent 16-byte records. A "seek" to record number 32 then sets you up to "get" the first index entry. (This is all PASCAL nomenclature. You Business BASIC types will have to cope).

Having gotten (GETted?) an index entry, you know to which of the DATA file records it refers. SEEK that record number minus one and GET it. The "minus one" crops up because the index entry pointers range from "1" to "960," whereas SEEK calls the first record of a file "0." To read the whole file, go

through the cycle for the number of times specified by the INDX header's record count. There is no "end of file" indication. The sequence of DATA records will be as advertised in the MLM "The file is sorted by _____" message.

If you're not fussy about sequence, you can speed up reading the DATA records rather considerably. The method is not intuitively obvious, but it isn't all that arcane, either. Set up an array of 960 booleans, each initialized to FALSE. Read in all index entries ("record count") sequentially. Set the array item corresponding to each entry's DATA record pointer TRUE. Then read the DATA file sequentially, keeping track of record number and records used. If the array entry corresponding to record number is FALSE, bypass the record, else increment "records used" and process the record. When "records used" equals "record count," you're done.

The MLMUTIL Intrinsic Unit

The following sections revolve about and continually reference the PASCAL intrinsic unit named MLMUTIL whose listing is attached. It contains the record descriptions for all three types of MLM records, plus a half dozen public procedures that handle the more onerous data encoding and decoding. Amid the public procedures are another half dozen private ones that could be useful in a broader context than MLM. Indeed, they were extracted from a library unit of PASCAL utilities that I've found to

be generally useful. They were placed here to make MLMUTIL a stand-alone unit.

It might be best to take a few minutes now and look over the source code for the unit for general familiarization. As I said, we'll be continually going back and forth between it and the following text.

The INDX Header Record

The INDX file's header is defined as a packed record type named "indx_hdr" in the beginning of the MLMUTIL listing. It holds by far the most unique data items of the three record types. And some of the most inexplicable. A bunch of the data names include "dunno," and I for sure dunno what they are. If anyone finds out, I'd appreciate being told. After six months of working with the record description, though, I haven't been burned by not knowing what the dunno's mean. With that modest encouragement, let's look at Figures 2, 3 and 4, the index header records, datum by datum.

Some of the "index header" record's entries seem just plain inexplicable. The Sx_LINE_NUM and Sx_LINE_ENTRY fields are redundant. Sx_NAME_NUM points to DATA_GROUP "x", which contains the line number and entry-within-line for sort field "x". One might also ask why the WRK array is in the record at all. A lot of great questions can come to mind.

The questions of whether "NAME" is the primary sort field and "ZIP CODE" is the secondary sort field are not necessarily flaky. They, and perhaps some of the integer "constants," are needed to handle Free Format records. I have no experience with -- and no interest in -- such things, and they will never be mentioned again herein.

The MLM Sort Field

MLM's "Sort Field" came up several times in the index header's record description, and it will arise again in the actual index entries. Now seems as good a time as any to describe it. The "sort_field" type in the MLMUTIL unit is rather ingenuously defined as a three-element array of integer. A dump of a sort field indeed shows three repeating fields, each an integer initialized to \$'8000', -32768. That's the only way I've found to encode/decode the field, but it may be inappropriate for actual sorting. As they say in the textbooks, actually sorting the

Bytes	Data Name	Datum Description
001-002	DUNNO_1	It seems to be an integer of no redeeming value.
003-008	MLM_ID	A length-5 string containing "MLMgr." It doesn't <u>seem</u> to matter if you call it "Fred ," so I don't know what purpose it serves.
009-010	DUNNO_1A	Integer constant, the number 16. This may or may not indicate that the length of each index entry is 16 bytes.
011-012	REC_COUNT	Integer denoting the number of active records in the file. Very important quantity.
013-014	DUNNO_1B	Integer constant, the number 0.
015-016	INT_960	Integer constant, the number 960. Indicates the maximum number of records permitted in the file.
017-048	LIST_NAME	A length-31 string containing the name that was given to this particular mailing list.
049-054	S2_HIGH_VAL	The highest value found the last time the file was sorted on its secondary sort field. See the "Sort Field" section.
055-060	S2_LOW_VAL	The lowest value found the last time the file was sorted on its secondary sort field. See the "Sort Field" section.
061-062	DUNNO_2A1	Integer constant, the number -1, \$'FFFF'.
063-064	DUNNO_2A2	Integer constant, the number +1, \$'0001'.
065-070	S1_HIGH_VAL	The highest value found the last time the file was sorted on its primary sort field. See the "sort field" section.
071-076	S1_LOW_VAL	The lowest value found the last time the file was sorted on its primary sort field. See the "Sort Field" section.
077-078	DUNNO_2B1	Integer constant, the number 0.
079-080	DUNNO_2B2	Integer indicating the file's sort status. If the file was most recently sorted on the primary sort field, the number becomes -1. If the file was most recently sorted on the secondary field, it is -2. Zero implies an unsorted file.

Figure 2

Bytes	Data Name	Datum Description
081-092	Print format fields from the "Print/Display" MLM screen:	
082	LBL_MARGIN	"Left Print Margin," 0-99, default 0.
084	LBL_ACROSS	"Number of Labels Across Page," 1-4, default 2.
086	LBL_SPACING	"Characters Between Labels," 0-99, default 4.
088	LBL_SKIP	"Lines Between Labels," 0-99, default 1.
090	LBL_WIDTH	"Width of Physical Labels in Characters," 1-99, default 30.
092	LBL_LINES	"Height of Physical Labels in Lines," 1-99, default 5.
093-096	Logical label data from the "Record Format" MLM screen:	
094	LINE_LENGTH	"maximum characters per line," default 30.
096	LINE_COUNT	"lines printed on physical label," default 4.
097-288	DATA_GROUP	Array (1..12) of label and comment field data from the "Record Format" MLM screen. Each of the twelve fields is 12 bytes long.
108	DATA_NAME	A length-11 string containing the name given the label or comment field. If this field isn't used, string length is zero (byte one = chr(0)), and the field is filled with "=" symbols.
110	LINE_ENTRY	Integer indicating this field's position within its label or comment line. Zero if this field isn't used.
112	LINE_NUMBER	Integer showing which label or comment line this field is on. If LINE_NUMBER exceeds LINE_COUNT (above), the line is a comment. Zero if this field isn't used.
289-290	S2_LINE_ENTRY	Integer indicating the secondary sort field's position within its label or comment line.
291-292	S1_LINE_ENTRY	Integer indicating the primary sort field's position within its label or comment line.
293-294	S2_LINE_NUM	Integer indicating the label or comment line that the secondary sort field is on.
295-296	S1_LINE_NUM	Integer indicating the label or comment line that the primary sort field is on.
297-298	S2_NAME_NUM	Integer indicating which of the up to twelve defined DATA_NAME's (see above) is the secondary sort field.

Figure 3

file based on its sort field values is intuitively obvious and left as an exercise for the student. In other words, I haven't the foggiest notion.

Before going into the data structure itself, some background is in order. MLM sorts no more than the first nine characters of a label or comment field (or no more than the first nine characters of "last name" if NAME is a sort field). A sort field contains only letters, numbers, left and right parentheses, hyphens, and spaces. The value that MLM displays for a sort field is created by moving the sort field's value into a nine-character string, truncated or padded to the right with blanks as needed. Then letters are converted as needed to upper case, and non-permissible characters are replaced with blanks. Then, finally, a sort field is constructed from that "sort string." A label/comment field value of "R & D, Inc." would become "R D INC" as a sort string.

The MLMUTIL procedure SORTSTRG creates an MLM sort string from an arbitrary character string. If the sort field represents the label/comment entry called NAME, SORTSTRG does its damndest to find a "last name" in the string and returns it as the sort string. The private procedure LNAME in SORTSTRG performs this function. It is of more general use than creating MLM sort strings, and you may wish to add it to your PASCAL programming toolkit. Procedure LNAME, further, uses another general-use procedure (BLANK1) to strip leading, trailing, and double blanks from a string.

The number of permissible characters in a sort string is limited to the forty mentioned in order to save space by encoding. They are encoded as ASCII 0 through 39 as follows:

- 0 space
- 1 left parenthesis
- 2 right parenthesis
- 3 dash
- 4-13 the digits "0" through "9"
- 14-39 the letters "A" through "Z"

The nine characters of a sort string are split into groups of three, one group for each integer in the sort field. For example, the sort string "LINDQUIST" would be converted and grouped as follows:

Group 1	Group 2	Group 3
-----	-----	-----
L I N	D Q U	I S T
25 22 27	17 30 34	22 32 33

A hex dump of the corresponding sort field shows that Group 1 is represented by the integer \$'1FCB', or decimal 8,139.

OK, readers, you've as many clues as you need at this point. Each three-character group, with 40 possible values per character, fits into a 16-byte integer field. You might wish to exercise your mind a bit with the following puzzle. What mechanism converts the string "LIN" to the integer \$'1FCB'? Fortunately, it is unique. You can check your answer by applying the same algorithm to the other two groups. "DQU" translates as \$'EF12', and "IST" is \$'OEAL'.

Short (?) Break for Puzzle Solving

Well, did you get it? Give up? Never start? The first group can skip to the next section. The second group should read on. If you've no interest or ability in the math realm, worry not. MLMUTIL's procedures SORTFLD and SORTVAL handle the encoding and decoding of MLM sort fields for you. The solution is keyed to the numbers 40 and \$'8000'. Powers of 40, actually. Let's encode Group 1 ("LIN") as the first integer in a MLM sort field:

"L"	25	x	40 ** 2	=	40,000
"I"	22	x	40 ** 1	=	880
"N"	27	x	40 ** 0	=	27

			Sum	=	40,907
				=	\$'9FCB'

Aha! We've seen that \$'FCB' before, the last three hex "digits" of the sort value. Adding \$'9FCB' and \$'8000' yields \$'11FCB'. Ignoring the high "digit" yields the sort field value. Try it on the other two groups of sort string "LINDQUIST." The MLMUTIL procedure SORTVAL implements this algorithm, taking a nine-character sort string and returning its three-integer sort value. SORTVAL uses the private procedure X TO Y to raise an integer "X" to the integer "Yth" power, perhaps another addition to your PASCAL tool kit.

You can also invert that encoding algorithm to decode an MLM sort field into its equivalent sort string. Those of you who gave up on encoding have a chance to redeem

yourselves here. Can you puzzle out the de-coding procedure?

Short (?) Break for Puzzle Solving

Welcome back. Did you get it? Something of a bear, yes? For those still stuck, the

procedure involves first converting the \$'1FCB' to decimal, the result being 8,139. Now remember that encoding left a five "digit" hex number, so the real value of the sort field is either \$'11FCB' or \$'01FCB'. The difference is 64K (65,536). Back to the ubiquitous \$'8000' (decimal 32,768).

Bytes	Data Name	Datum Description
299-300	S1_NAME_NUM	Integer indicating which of the up to twelve defined DATA_NAME's (see above) is the primary sort field.
301-302	S2_ZIP_SORT2	Integer indicating whether the file's secondary sort field is "ZIP CODE." If so, the value is 1, else it is 0.
303-304	S1_NAME_SORT1	Integer indicating whether the file's primary sort field is "NAME." If so, the value is 1, else it is zero.
305-512	WRK	Array (0..12) defined the same as the 16-byte index entries (see below). Apparently used by MLM as work space of some sort.

Figure 4

Bytes	Data Name	Datum Description
001-002	REC_NUM	Integer, range 1 to 960, indicating which DATA file record this index entry points to.
003-008	SORT_1	MLM sort field, the encoded value of the DATA file record's primary sort field.
009-010	SOUNDEX	Integer, the Soundex-encoded value of the DATA file record's primary sort field. Used in "similar name" searches of an MLM file.
011-016	SORT_2	MLM sort field, the encoded value of the DATA file record's secondary sort field.

Figure 5

Remember that \$'8000' was added to the calculated \$'9FCB' to get the five "digit" hex number whose high "digit" was lopped to yield the MLM sort field. Therefore, if the sort field value is less than \$'8000' (32,768), the high "digit" is \$'1', else it is \$'0'. Since 8,139 is less than 32,768, the real value of the sort field is \$'11FCB', or 73,765.

From there it's easy to go the the encoding algorithm backwards. Subtracting \$'8000' (32,768) from 73,765 yields 40,907. Back to the powers-of-forty bit to translate that into sort string characters:

```
40,907 / 40 ** 2=25 remainder 907 25="L"
 907 / 40 ** 1=22 remainder 27 22="I"
 27 / 40 ** 0= 0 remainder 27 27="N"
```

As an exercise, you might decode the remaining integers of the sort field, \$'EF12' and \$'OEAL', the rest of the "LINDQUIST" sort string. Procedure SORTFLD in the MLMUTIL library unit implements this algorithm, yielding the nine-character string represented by a three-integer MLM sort field.

The MLM Index Entry Record

As I said earlier, the MLM index entries hold the key to the MLM data record file. Each is a sixteen-byte packed record of the type called "indx_entry" in the MLMUTIL unit. The record's data elements are shown in Figure 5.

Mail List Manager uses the index entries to avoid having to deal with DATA file records as much as possible. Keeping all 960 index entries in memory takes little room (15,360 bytes) and speeds processing quite a bit. Sorting an MLM file involves no more than re-ordering the file's `INDX_HDR.REC_COUNT` active entries. Searching for a record is a snap, scanning the index entries for a matching sort field value (or a "similar name" match on the primary sort field), and only then actually seeking and decoding the DATA file record.

Of the four data elements in an index entry record, only `SOUNDEX` has yet to be introduced. The MLMUTIL procedure named `SOUNDEX` creates a Soundex-encoded integer from an MLM sort string. Those not interested in exactly how that is done may skip the next section.

(Ed. Note: Part II of Dr. Bloom's article will appear in Volume Two, Number Four of the Open Apple Gazette .)

Printer Control Program

by
Thomas A. Muller

One of the big advantages of having a high quality dot matrix printer is the ability to use several types of character spacing, thus permitting the optimum choice for each software/paper combination. On my Prism 132 I like to use 5 characters per inch for presentations, 10 for letters, 12 for forms, and 16 for spreadsheets and data base reports.

Unfortunately, most printers don't offer easy switch controlled type changes. On my Prism and on my MX80 at home, it is necessary to get into the microswitches and to reboot the printer, possibly having to reset the top of form as well. A cleaner method is to use program control, but most packaged software doesn't permit special control characters to be sent to the printer. Those that do rarely reset to the default condition when printing is complete. Besides, you may not always want to use the same typestyle with a given software package.

The program below is a standalone printer control program that runs automatically when added to a BASIC boot disk under the filename "Hello". It doesn't require powering down the printer and can be run whenever you are shifting software. I resort to the switches only when I find myself in the middle of a program with the wrong printer setting. I don't have a hard disk yet, but am planning to add this to the software menu when I do.

In the current version, the printer wastes a page testing itself. This feature can be eliminated by inserting: "3031 Goto 4010". However, I've found that the resetting process advances the printer one line, ruining printer registration. Seeing the test improves operator confidence, so I run it this way rather than writing a routine to re-register the printer on line 1 of the paper.

The program as written is not totally bug free. Lines 42 to 1740 are the character spacing loop, which works just fine. Lines 2000 to 2830 are the line spacing loop, which I cannot get to run properly and have bypassed. If one of the readers can catch the bug, we'd all benefit.

The concept of this program applies to almost any printer and can be rewritten by substituting the appropriate control statements in lines 500-510, 600-610, 840-850, 1000-1010, 1200-1210 and 1680-1690. I have a similar version for the MX 80 to run off an Atari 800, although the version here is strictly Apple /// to Prism 132. I've found that utility programs like this are most effective when they are placed on a boot disk as an autorun program or as a subroutine to an autorun menu.

```

10    REM PRINTER SETUP PROGRAM, NAME
      PRINTSET
20    PRINT"=====
30    PRINT  PRINTER SETUP ROUTINE
           by Thomas A. Muller
40    PRINT"=====
41    PRINT
42    PRINT
50    PRINT"Chose the number of char-
      acters per inch:"
60    PRINT" [1] 5.0"
70    PRINT" [2] 6.0"
80    PRINT" [3] 8.4"
90    PRINT" [4] 10.0"
100   PRINT" [5] 12.0"
110   PRINT" [6] 16.8"
120   INPUT CHOICE
122   OPEN#1, ".PRINTER"
124   OUTPUT#1
130   IF CHOICE=1 THEN GOSUB 500
140   IF CHOICE=2 THEN GOSUB 600
150   IF CHOICE=3 THEN GOSUB 840
160   IF CHOICE=4 THEN GOSUB 1000
170   IF CHOICE=5 THEN GOSUB 1200
180   IF CHOICE=6 THEN GOSUB 1680
185   CLOSE#1
190   GOTO 50
500   PRINT CHR$(29)
510   PRINT CHR$(01)
520   CLOSE#1
530   PRINT"The printer is now set
      for 5.0 characters per inch."
540   PRINT"It will print 43 char-
      acters on 8.5 inch paper and"
541   PRINT"70 characters on 14 inch
      paper."
550   GOTO 2000
600   PRINT CHR$(30)
610   PRINT CHR$(01)
620   CLOSE#1
630   PRINT"The printer is now set
      for 6.0 characters per inch."
640   PRINT"It will print 51 char-
      acters on 8.5 inch paper and"
641   PRINT"83 characters on 14 inch
      paper."
650   GOTO 2000
840   PRINT CHR$(31)
850   PRINT CHR$(01)
860   CLOSE#1
870   PRINT"The printer is now set
      for 8.4 characters per inch."
880   PRINT"It will print 71 char-
      acters on 8.5 inch paper and"
890   PRINT"117 characters on 14 inch
      paper."
900   GOTO 2000
1000  PRINT CHR$(29)
1010  PRINT CHR$(02)
1020  CLOSE#1
1030  PRINT"The printer is now set
      for 10.0 characters per inch."
1040  PRINT"It will print 83 char-
      acters on 8.5 inch paper and"
1050  PRINT"139 characters on 14 inch
      paper."
1060  GOTO 2000
1200  PRINT CHR$(30)
1210  PRINT CHR$(02)
1220  CLOSE#1
1230  PRINT"The printer is now set
      for 12.0 characters per inch."
1240  PRINT"It will print 99 char-
      acters on 8.5 inch paper and"
1250  PRINT"167 characters on 14 inch
      paper."
1260  GOTO 2000
1680  PRINT CHR$(31)
1690  PRINT CHR$(02)
1700  CLOSE#1
1710  PRINT"The printer is now set
      for 16.8 characters per inch."
1720  PRINT"It will print 139 char-
      acters on 8.5 inch paper and"
1730  PRINT"233 characters on 14 inch
      paper."
1740  GOTO 2000
2000  GOTO 3000
2010  PRINT"Chose the number of lines
      per inch:
2020  PRINT" [1] 6.0
2030  PRINT" [2] 8.0
2040  INPUT lines
2045  OPEN#1, ".PRINTER"
2050  IF lines=2 THEN GOSUB 2800
2600  CLOSE#1
2605  PRINT"The printer is now set
      for 6.0 lines per inch."
2610  PRINT"It will print 66 lines on
      11 inch paper."
2630  GOTO 3000
2800  PRINT CHR$(27);"B, 1, $"
2810  CLOSE#1
2820  PRINT"The printer is now set
      for 8.0 lines per inch."
2830  PRINT"It will print 88 lines on
      11 inch paper."
3000  OPEN#1, ".PRINTER"
3005  OUTPUT#1
3030  LET n=n+1

```

```
3040 PRINT"PRINTER TEST 123456789
      0123456"
4000 PRINT CHR$(12)
4010 CLOSE#1
4020 END
```

BPI General Accounting Package

by
Dennis R. Cohen

BPI is called a General Accounting package because it is not just a General Ledger system as are most packages in its price range. Payroll reports, payroll checks, vendor checks, customer statements and invoices, lists of employees, vendors, customers, and budgets are all available in addition to the reports one normally finds in an accounting package. It maintains a number of journals and ledgers while performing all the standard accounting functions associated with a General Ledger package, as well as many of the functions of packages that are usually add-ons in other systems.

Getting Started

Learning to use BPI General Accounting is very easy. (If I learned it, anyone can!) The program is driven entirely by a series of menus. At first, the top-level menus looked a little cluttered and intimidating, but I rapidly adapted and manipulation of the program became second-nature. Backing out from any menu level is fairly simple, generally "Open-Apple E". You're always aware of your current location since the menus are all numbered and the current numeric path is displayed in the upper right corner of the screen.

Ease of use is further aided by on-line tutorials included with the system. A large part of the manual is devoted to guiding you through the tutorial and includes numerous pictures and diagrams.

BPI is also very safe to use so you don't have to worry about losing data. Any operation of consequence requires reconfirmation and the system maintains a complete audit trail of your transactions. Each company you establish has its own company code and you can also tie a password to any company created.

BPI Program Features

Most accounting systems do not provide the Accounts Payable, Accounts Receivable, and Payroll capabilities except as an additional module. While BPI offers these packages as separate modules in their full capacity, the basic functions of these modules are all available in the General Accounting Package. The user has an Accounts Receivable Ledger, an Accounts Payable Ledger, and a Payroll Ledger.

Any or all of these three ledgers, plus the General Ledger, may be printed using a single keystroke. BPI General Accounting also provides you with a check-printing and statement-printing capability.

Income statements may be consolidated or departmental, and you can print them in any of five formats:

- Current and Year-to-Date
- Current Budgetary
- Year-to-Date Budgetary
- Current Comparative
- Year-to-Date Comparative

Trial balances are available for any of the four ledgers and balance sheets may be either summarized or detailed, comparative or not. A spread sheet capability is also included with five templates. You cannot, however, construct your own template.

End-of-period processing is also extremely easy with BPI and ties in very nicely with the disk management utilities capability. The only thing that I found uncomfortable about data-entry is that negative entries have the minus sign as a suffix rather than a prefix, but I am a programmer/mathematician and not an accountant.

BPI With profile And The A143

Enough of the hosannahs. Nothing is perfect, and that includes this package. Of the BPI package's few problems, the majority may be laid right at the doorstep of Apple Computer, Inc. Beginning in July of 1983, Apple had been advertising this package very heavily in two-page ads in virtually every major microcomputer publication. I have yet to find one comment in any of those ads that states that BPI General Accounting is designed specifically for use with the profile hard disk. Nor does it appear

anywhere on the BPI package other than in fine print on the back. There it states "or other compatible mass-storage device."

Since I run on a system with three external drives (two of them Micro-Sci A143's), I didn't look at the back of the package and probably wouldn't have been concerned if I had since I also have access to an Apple /// with a 20 megabyte Corvus hard disk. This lack of concern would have been a mistake as I found BPI easier to install on my A143 than the Corvus, and the floppy installation was not trivial.

Although the BPI accounting system was designed for use with the profile, using the System Configuration Program (SCP) on the Systems Utilities disk will enable the program to work satisfactorily with the Micro-Sci A143 floppy drives. Fortunately, I install systems as a business and am very familiar with the SCP option on the System Utilities diskette.

(Ed. Note: Due to space limitations, we are unable to provide the step-by-step procedure for reconfiguring your System Configuration Program to accommodate BPI on a Micro-Sci A143. We will publish this in the next issue. However, if any of you would like to have this information before then, drop us a line and we will forward the procedure to you.)

The fact that you can reconfigure the drivers in your system is a tribute to the versatility of the /// and the good design and implementation of BPI. Apple, however, has not advertised the package along these lines. I feel that their advertising and packaging is misleading and can only contribute to the feeling that Apple really doesn't care about the ///, all of their protestations to the contrary.

Some Flaws In The Package

BPI does not have enough customer service lines and busy signals are a frequent occurrence. They are adding more lines so this should not be a problem by the time you see this review. They are very helpful on the phone and when they came up with more information, they called me to add to what they had already told me.

The other flaw concerns a documentation practice that seems unprofessional. It is not unusual for a package to be released

with errors in the manual and for an "Errata Sheet" to be included. Where BPI deviates is that they include an extra cost for corrected manuals. If the corrected manuals are available, they should be shipped with the package.

The low-cost update option should be sent to those who purchased and registered their systems before the manuals were updated. I seldom object to paying a nominal fee for an update: it's just part of the price I pay to get something early. However, it irks me to be told at the time that I purchase that I already need to buy the upgrade as well.

The Bottom Line

My overall impression of the BPI General Accounting Package for the Apple /// is that it delivers more for the dollar than any other accounting package on the microcomputer market. If you are looking for a general-purpose accounting package and either have a profile or another large mass-storage device (and the expertise to do the configuration), you need look no further because I have yet to find a package to compete with BPI in its price-range, \$395 to \$495.

(Editor's Note : We have received numerous requests from /// owners as to which accounting package we recommend. Because it is difficult to ascertain your particular needs, and since various programs have features that may or may not suit your needs, we strongly encourage /// owners to investigate the many packages currently available for the ///. If you do not have the Guide To Apple /// Software , you can obtain them for \$4.75 through the group. Group member Ed Flora has promised to review another accounting package, **Hard Disk Accounting Series** from Great Plains Software.)

Stems and Seeds:

Do You Want To Network With Other /// Owners In Your Area?

We have received numerous requests from /// owners wishing to obtain the addresses and phone numbers of other /// owners in their area for networking purposes. If you wish to have us give this information out, please respond in the affirmative and provide us with your system's equipment as well as what software and hardware you have used.

Interfacing Apples And Oranges

by
Brian R. Chambers, MD

An interface between the Apple /// and other microcomputers may dramatically increase the Apple's potential applications. For example, the ability to transfer text files between home and office provides me with greater flexibility in preparing journal manuscripts and grant applications. This necessitated linking my Apple /// and an Intertec Compustar (or Superbrain) at the office.

Initially, I was not certain that an Apple /// would "talk" to a Z-80 based CP/M computer, but it was surprising how much could be achieved without communications software. The following is a synthesis of information gleaned from telephone conversations, technical notes and many hours of trial-and-error. It is recommended that you establish a direct hard-wire link before attempting telephone transfers, as "teething" problems are inevitable.

Like the Apple ///, the Compustar has a serial RS-232 port, in fact two. The auxillary port is generally used for a printer. The main port is enabled by running the CP/M transient program DDT.COM and using the S (set) command to change the hexadecimal value at memory address EF02 from 17 to 37.

At first glance, the RS-232 pin assignments appear to be a problem. Below is a comparison of the assignments for the two machines:

<u>Apple ///</u>	<u>Description</u>	<u>Compustar</u>	
1	SGND	Shield ground	1
2	TXD	Transmitted data	2
3	RCD	Received data	3
4	RTS	Request to send	4
5	CTS	Clear to send	5
6	DSR	Data set ready	6
7	GND	Ground	7
8	DCD	Data carrier detect	
	?	Transmit clock	15
	?	Receive clock	17
20	DTR	Data terminal ready	20
	?	Ring indicator	22
	?	Clock	24

The serial port on the Compustar has four extra pins wired, yet pin 8 is not connected. Fortunately, only pins 1,2,3,4,5,6,7 and 20 are required (maybe only 2,3,6,20). Purchase two "male" RS-232 connectors and a length of ribbon cable and solder the connections. Pins 2 and 3, 4 and 5, and 6 and 20 must be reversed at one end unless you have a "modem eliminator" (also known as a "null modem"). Before connecting the two computers, ensure they are plugged into the same power outlet to prevent damage from incorrect power wiring.

Assuming, for the time being, you have no communications software, the next step is to configure the Apple /// RS-232 driver (a component of the file SOS.DRIVER) using the System Configuration Program on the Utilities Disk. (Refer to Standard Device Drivers Manual.) If you prefer to work with Apple CP/M, this task is a little more complicated. SOS.DRIVER must be configured in the same way and then transferred to the CP/M disk as the file DRIVER.SOS. After booting the CP/M System Disk, use SOSXFER.COM to convert SOS.DRIVER (SOS format) to DRIVER.SOS (CP/M format). (Refer to Softcard /// Installation and Operation Manual.)

On the Compustar, the RS-232 driver is modified by menu-directed options using the transient program CONFIGUR.COM. Because I planned to use a 300 baud modem, I configured both machines for 300 baud, asynchronous, half-duplex transmission with 8 data bits and 1 stop bit. This set-up represents the simplest form of transmission, ie "no protocol". Transmissions at faster baud rates result in lost data if the file is to be written to disk, unless a communications protocol is adopted. (See Standard Device Drivers Manual.)

Text files may be transferred from Apple /// to Compustar using the System Utilities' "Copy a File" option. When prompted for the destination file, type ".RS232". Similarly, you can send files from Applewriter /// by changing the print destination on the PRINT/PROGRAM command menu to ".RS232". Unfortunately, the Apple /// cannot receive files in the same way. This is not a problem if you only intend to send files.

To receive files, you could write a BASIC program and take advantage of the GET# function, which reads characters one at a time from the .RS232 driver, but I opted for the communications software, Access ///. There is another alternative - if your Apple /// has a Z-80 card (eg Softcard ///), text files may be transferred in either direction using the transient program PIP.COM. Remember to convert them from SOS format to CP/M format first, using SOSXFER.COM.

Apple/// SOS to Compustar CP/M

Boot Access ///, select the "Transmit a File" option, but wait until the Compustar is ready before the final command to commence transmission. On the Compustar run the PIP.COM transient program as follows:

```
Type:          PIP <filename.ext>=
                INP:[B] <RETURN>
Compustar responds: ECHO Y/N
Type:          Y <RETURN>
Compustar responds: +
```

Start transmission from the Apple /// keyboard. The file will be echoed on both screens and written to the diskfile designated. On completion of the transfer, typing @Z (ie CONTROL-Z) closes the new Compustar diskfile.

Compustar CP/M to Apple/// SOS

On the Apple, choose terminal mode, turn on the recording filter and recording mode. On the Compustar use the following commands:

```
Type:          PIP OUT:=<filename
                .ext>[B] <RETURN>
Compustar responds: ECHO Y/N
Type:          Y <RETURN>
Compustar responds: +
Type:          ^B
```

Transmission then proceeds. On completion, the Apple /// diskfile must be closed by typing <OPEN APPLE>-R.

Apple/// CP/M to Compustar

The procedure for receiving a file on the Compustar is unchanged. OUT: is an invalid logical device in Apple CP/M. Use the following command:

```
PIP PUN:=<filename.ext>[BE] <RETURN>
```

Compustar to Apple /// CP/M

The procedure for sending a file on the Compustar is unchanged. INP: is an invalid logical device in Apple CP/M. Use the following command:

```
PIP <filename.ext>=RDR:[BE] <RETURN>
```

If the text file was created using Wordstar, you will notice that the transmitted text has lost characters and contains inappropriate line feeds. The explanation is a bit technical but the problem is remedied by adding the Z option to the PIP command line (ie PIP OUT:=<filename.ext>[BZ]).

If it was created with Applewriter, the transmitted version will contain inappropriate line feeds. You can avoid this by transmitting a file that was first "printed" to disk. Before leaving Applewriter, change the destination name on the PRINT/PROGRAM command menu to a disk-file name and PRINT the file.

Once skilled in direct hard-wire file transfers between computers, you are ready for file exchanges via modem. Before connecting the RS-232 cable from computer to modem, don't forget to disconnect the "null modem" (or restore the pin connections - see above). File transfers are straightforward using the keyboard sequences listed. Note that there are some limitations. Keyboard entry is required on both computers and only text files, including .HEX and .DOC files, may be transferred with these techniques.

Although I refer specifically to an interface with the Compustar, the same principles should apply for any "alien" microcomputer. Hopefully this article will stimulate some informative comment from other Apple /// users interested in telecommunications.

I would like to hear from anyone who has installed (or knows how to install) the CP/M public domain communications program, MODEM-7. on the Apple ///.

Stems and Seeds:

Upgrade For Less \$\$\$\$\$

Apple Computer has announced a major reduction in the cost to upgrade 128K machines to 256K. Suggested retail price is \$300. See your dealer for more details.

AppleFile ///: Data Base Management With a Plus!

by
Llona Cunningham

Since Data Base Management has been something of a passion for me during the past three years, I was anxious to get my hands on AppleFile /// as soon as possible. But I also approached it with a skeptical eye. This new product would not only have to work, it would have to work better than the "custom to my own specs" programs I have written for myself.

The choice between swapping disks and/or limiting my data base files to less than 300 records became intolerable at an early stage. Hence, the move to learning how to program (Business BASIC), and spending the last two years developing and improving my own DBM system to handle my files. But there are times when I need a new file layout and the prospect of revamping my base program (again!) is anything but appealing.

What I need is a good, commercial Data Base Management program that can handle a new file quickly, and handle growth and changes of the file over long periods of time.

After all, isn't that what DBM is all about?

What is DBM?

Primarily, a good, commercial Data Base Management program should do the following:

1. Allow you to design a form or "model" to your specifications as the way you want the information entered and stored.
2. Allow you to read, edit, delete, or print any piece of information in the file.
3. Sort the information and organize it in any manner you desire.
4. Print reports of organized information by your specifications of what you want printed, and where.
5. Have the ability to handle files of the size you require in your applications.

AppleFile /// handles the 5 basic requirements, plus several others that prove to make it easily the most user-oriented DBM program you're likely to find for the Apple ///.

Getting Started

Here's what you'll find in the box:

10 disks:

- 1 Installation Disk
- 4 AppleFile Program Disks
- 2 Start-up Disks
- 1 Sample Files Disk
- 2 Training Disks

2 Manuals (3 Books):

- Tutorial
- User's Manual (2-volume)

3 Cards:

- "Read Me First" card
- 4-Sided Reference card
- "Getting Started" card

If the contents seem overwhelming at first, you'll get comfortable with the package quickly by starting with the "Read Me First" card. Allow an hour or so to run the 2-Disk "Apple Presents AppleFile ///" training program. Then follow through with the tutorial manual.

Appendix A in the tutorial manual gives you step-by-step instructions to install AppleFile /// on your Profile hard disk unit. (No, you can't transfer the program to your A143. Maximum storage on the A143 is less than the 1051 blocks required by the multi-disk series.)

Clear, easy to understand instructions for starting up the program are also included in the tutorial's Appendix A. Such professionally prepared information (even a reminder to make back-ups of your configured disks!) is a refreshing improvement over the documentation some of us have fought our way through in the past.

Both the Tutorial and the 2-book Reference Manual are thoroughly indexed for quick reference. The Tutorial also includes a helpful glossary of terms common to DBM but which may not be so familiar to new users.

The 4-sided reference card is a handy addition to the manual set but most users will find that everything they need is prompted on the screen or contained in the help-screens available while the program is running.

Getting Started In A Hurry

If you consider yourself an experience user in the DBM environment, you may have trouble sitting still through the training program. You may well decide to delegate the tutorial to your spouse, and plunge ahead into the Reference Manual set.

A word of advice: If you skip the Tutorial, you should also skip (temporarily) the Reference Manual set, and do your "plunging" with the program itself (don't forget you must first configure the start-up and installation disks, and install AppleFile /// on your Profile).

The Reference Manual set is highly detailed, well organized, and definitely lives up to the standards we've been asking for in reference material. But the two-book set is obviously designed for REFERENCE. Reading them as you would a textbook is more likely to be more counter-productive than helpful due to the sheer volume of information.

Chances are, you'll find a trial run through the Menus and their accompanying help screens the fastest and most helpful method of getting down to business with your own file data. And here's the best part: you can print any screen (help screens and menus included) by pressing Control-P from any point in the program. Note: the Solid-Apple you see on the screen will be printed as an asterisk (*) on the hard copy.

It may not be as speedy as looking at the reference card, but printing a copy of the editing commands and a few of the help screens (and reading them, of course) can do a lot toward helping you learn to take control of the project.

What You Can Do With AppleFile ///

From the Main Menu you can:

1. Work With an Existing Data Base
2. Use the File Manager
3. Set Up/Change Model
4. Make an Empty Data File
5. Transfer/Repair Data Base Contents

6. Review/Change AppleFile Special Settings
7. ESCAPE to Leave AppleFile ///

Solid-Apple-? from the Main Menu will get you a help screen that explains each of the options. Entering Control-? from the help screen (or from any help screen throughout the program) will get you the key strokes used to edit your keyboard entries.

Right and Left arrows move the cursor. Using the arrows with the Control key erases characters in the direction of the arrow. Control-E erases the line. Control-U "undoes" the line and restores the standard entry. Control-K is used to "kill" all characters to the right of the cursor. And of course Control-P sends the screen display to the printer.

As with any other DBM program, the first thing you have to do is set up the model you want the file to use. Each field is designed as a separate item, making use of several prompts to guide you through establishing the name of the field and any specifications you want to apply.

You can select (T-Text, I-Int, R-Real, M-Money, S-Soc Sec, D-Date, C-Clock, P-Phone) as the field type; specify whether the field is required to have data, set up Standard Entry data to use as default entry, affix column titles to be used on reports, and select the maximum field length with justification of L-Left, R-Right, or C-Center.

In addition, you can establish a prefix to be automatically attached to each record (such as "Dr." for a file which will contain the names of physicians), and/or automatic suffixes (such as "M.D." to be added at the end of each name on the same file).

When the fields have been designed, the next thing to do is establish what your keys will be. Key fields are used to maintain alphabetical order. For instance, "Last Name" might be the key field used to organize your personal phone book. A nested key would be a sequence of "Last Name, First Name" so that any two identical last name records would be sub-organized by the first name. You can set up multiple key sequences, nesting as many as 8 separate fields for each key.

Be prudent with your choices. The more keys, and the more fields required to specify each key, the more space will be

consumed to maintain the index file. For instance, I experimented with a 10-field client data file using 5 separate key sequences and nested each one with 3 fields. The index file with only 8 records took up 42 blocks on the disk! Increasing the file to 14 records caused the index file to grow to 52.

You would do better to establish one sequence using the most needed fields as your key to maintain the file. When you need alternate key sequences for special reporting, you can create a temporary key that won't have to gobble up such a disproportionate amount of disk space.

Establishing entry rules and/or Password protection are both optional and should be studied carefully if they are to be used. The explanations on the help screens are adequate. But if you aren't familiar with the process you would save yourself a lot of grief if you further studied those sections in the Reference Manual before using either one.

After designing your model and saving it on the Profile, select number 4 from the Main Menu -- Make an Empty Data File. This establishes your Model and your Data Base as viable files which can be accessed for data entry and maintenance.

"Work With an Existing Data Base" (number 1 from the Main Menu) is Menu-driven so well that you probably won't need the help screens at all. The process of adding records, editing records, and deleting records is self explanatory as long as you understand that function in the DBM environment. If not, you'd better make another run through the tutorial.

Do make use of the help screen and Reference Manual if you don't feel comfortable with the "Print Report" option. The choices are pretty simple (and herein lies my biggest disappointment in the program); you can print only one line for each record in the file. The choice you have to make is to decide which fields to print (if what you have can't fit on one line), and in what order you want them to appear.

You can add, move, or remove column choices. You can select a different key (a temporary sort index). You can make Summary Breaks (Totals, Averages, Line Counts). And you can set up Selection Rules (print only those records whose 2nd field begins with a letter between "C" and "F", or some such designation).

Someday, some kind soul will add the ability to print ALL the fields, from selected records, in an organized report, even if it takes more than one line per record to do so. Until then, what this program offers is a great deal of professional reporting capabilities that should not be overlooked.

In spite of all the frills, the basic features listed so far are pretty standard stuff for Data Base Management in a commercial program. Now we come to the elite features, the ones you won't find in average programs.

From the Main Menu, item 2, Use the File Manager, is like having the familiar SOS.UTILITIES disk built into the program. You can Set Prefix, Make a new subdirectory; list, copy, rename, and delete files. You can set Date/Time, list devices, FORMAT A DISK, and calculate the number of records that will fit in a specified space.

If you have the A143 drive, you CAN transfer the A143 formatting drivers to your start-up disk and format from within the program. You cannot verify, however, so use this capability with caution. Better to format the A143 from your Utilities disk and verify before trying to store a data file. But it is nice to know the formatting can be done if it becomes necessary.

To be able to calculate the number of records (free record space remaining on the specified device) is a feature long awaited by DBM users that will soon become the one we didn't know how we ever got along without. Can you remember working with a file, knowing you needed to add 20 more records, and wondering if you should try it first or start over with a new file? This little helpmate will tell you, based on the size of your model and its specifications, how many more records will fit (give or take a few -- e.g. 115 to 153) in the free space remaining on the disk.

By the way, that 10-field client data file I experimented with was given a range of 1072 to 1719 records possible on the A143 with a blank, formatted disk. That's a wide range, but would probably tighten considerably if I actually had enough records stored to be worrying about whether I could afford to add more. Still, it's nice to have even that much of an estimate to work with when deciding if the file should reside on the hard disk or the library shelf.

The second elite feature, item 5 from the Main Menu, Transfer/Repair Data Base Contents, is a utility rarely included in such an already powerful program.

You can transfer data from one data base to another by simply specifying source, destination, and which fields you want transferred (including setting up new entry rules). This is not a simple act of transferring data from a full file to an empty one. This is actually merging data from one file into another EVEN WHEN THE MODELS FOR THE TWO FILES ARE DIFFERENT! Think hard about the last time you wished you could transfer the daily business entries into a master file but couldn't because the master file had an entirely different make-up.

You can repair a damaged file, to the extent of reading all undamaged information and storing it in a separate place. The damaged portion is stored in a "Discard" file, which can later be edited, corrected, or thrown out, whichever you choose. The point is, accidents do happen. And this program module allows you to save many hours of reconstruction by capturing most of the data and separating it from the damaged portion.

The last "elite" feature is item 6 from the Main Menu, Review/Change AppleFile Special Settings. You can set a Standard Prefix to be used each time AppleFile starts up. You can select the default printer for Control-P operation. You can define special Printer Codes. And you can reverse the display video. We've seen all these before.

But you can also set the frequency for saving records on the disk. This allows you to choose between frequent savings -- to the disk each time a record is entered if you want maximum security against power failure; or you can choose "whenever memory is full" to maximize the number of records you can enter before you have to wait for a disk access.

And last, but probably the most fun, is the feature which lets you play around with the audio signal. Even if you opt not to use the multiple tone setting, it's worth a few minutes just to play with it. You can select "no sound", "a beep" which is the same you get with most programs' error messages, or "distinct tones" which sound like mini sound-tracks from some old familiar education disks.

To hear each of the tones and/or adjust the

volume at which they're played, enter solid-Apple-H from the Select Tones Menu. There are five separate tones (or tone sequences) for Warning, Error, Confirmation, Completed Action, and System Failure. Better to hear them in this trial mode than to experiment with active data files.

Conclusion

The few points that I consider weak spots, i.e., the amount of space required for file storage, the inability to format reports of more than one line per record, and the sheer size of the program itself, are minor compared to the massive list of capabilities provided.

Users who are new to Data Base Management applications will find the menu-driven operation a considerable help in designing and building their files. Experienced users will appreciate the many features at their disposal which had previously been only wishful thinking.

In comparison to other commercial DBM programs, AppleFile is greatly superior to well-known programs in the same price range, and superior to many which cost a great deal more.

AppleFile /// is a product of Apple Computer, Inc. Retailing for \$325.00, it requires a 256K Apple /// plus a profile or other mass storage unit, like a Micro-Sci A143.

(Editors Note : Many of our readers have called requesting our recommendations for a data base. Suggesting one over another is difficult to do in that the data base you use will depends on the one which has the features you need. For example, if you need computed columns in your reports, or, if the ability to read other files such as DIF, Quick File, and PFS is important to you, look at **Keystroke** from Brock Software.

In future articles, Llona will compare the database features of **Versaform**, **/// Easy Pieces**, **Applefile ///**, and **Keystroke**.)

Creating Word Juggler Mailing Lists From A QuickFile /// Data File

by
G. Scott Warner

This article is written for those of you who have a basic familiarity with the Quickfile and Word Juggler application programs. A technique is described for creating a variety of different Word Juggler mailing lists from a single Quickfile file.

When I first purchased the Word Juggler word processing software for the Apple /// in our office, one of my primary objectives was to be able to create multiple mailing lists from a common data base. Since Quark offers a "Mail List Manager Interface" as an accessory for Word Juggler to allow it to access files created by Apple's Mail List Manager, I investigated the use of Apple's Mail List Manager for use in my application.

I found it to be inadequate in two fundamental respects. First, it did not provide the flexibility I needed in file configuration and report generation. Second, it did not allow enough flexibility in record selection. This second shortcoming was the most critical.

Perhaps a brief description of my application would help clarify this. Our firm assists individuals and corporations to invest capital in various ventures. In the normal course of our business, we regularly communicate with investors, each of whom may have a financial interest in only one venture up to several different ventures. I needed to create a mailing list for each venture since the information to be communicated to investors is "venture specific."

If a particular investor is involved in several different ventures, his name should appear on each of the mailing lists for those ventures. I did not want to create and maintain separate independent mailing lists for each venture. I wanted to maintain a common file with one record for each investor and then to create and easily update mailing lists for each venture based upon that common data file.

Since I had used Quickfile in some other applications, I knew I could create the file format I needed and generate the reports I needed. I also knew I could, by the use of

what I call a "SORT CODE" field in my file, arrange records in such a way that I could print out lists of the investors in each venture. (Since the number of investors is only a few hundred, I was not restricted by one of the most serious limitations of Quickfile: the entire data base must be able to fit in RAM. This generally limits the data base to less than one thousand records on a 256K machine and varies depending upon the number of categories in the file.)

What I needed was a way to print my lists of investors in each venture to disk and then convert those lists into a file that Word Juggler could use to merge with form letters and to print envelopes. I consulted with a software specialist at the computer store repeatedly recommended to me as having the most knowledgeable personnel of any Apple dealer in the Bay Area. In fact, they have provided me with valuable assistance in many areas. Unfortunately, they were unable to help me in this situation. My next course of action was to contact an application program specialist at Apple. After some searching, I found someone who knew Quickfile quite well and was somewhat familiar with Word Juggler, but in the end was not able to help.

The procedure for creating the mailing lists is really quite simple and is presented below:

FILE CREATION- This section briefly describes how to set up a file that can easily accommodate conversion.

1. Create a Quickfile file. Determine the category names you want for each record in the file and type them in. Each record in my file has the following category names:

LAST NAME - Listed separately in order to be able to arrange records alphabetically by last name.

FIRST NAME(S) - Listed separately for use by Word Juggler in the salutation of form letters.
(i.e. Dear <FIRST.NAME(S)>:)

MAIL ONE - The full name of the individual(s) in a four-line address. This line will be blank for three-line addresses (the usual case for home addresses).

MAIL TWO - Either the second line

of a four-line address, usually containing the name of a business (e.g. ABC Corporation), or the full name of the individual(s) in a three-line address (e.g. Mr. & Mrs. William Jones). (Using two general category names such as "MAIL ONE" and "MAIL TWO" allows the use of either a three line or a four-line address without leaving a blank line in three-line addresses.)

STREET ADDRESS - Street number, name of street and apartment or suite number.

CITY STATE ZIP - A separate category for zip code is usually not needed even though it would allow arranging records numerically by zip code for bulk mailings. If you do bulk mailings, Quickfile probably doesn't allow enough records.

HOME TELEPHONE - Home telephone number.

WORK TELEPHONE - Work telephone number and extension, if any.

SORT CODE - A field containing a single character code for each different venture in which the investor has a financial interest. We use digits 0 through 9, then alpha characters, then special characters if needed.

2.Establish Sort Codes. Choose the codes you require and make a written list of the codes along with an explanation for each code. Each record in your file may have from a single code up to all possible codes in the SORT CODE field.

3.Data Entry. Enter the data for each record into the file.

SET-UP - This section describes the one-time set-up procedure which must be followed to establish the basic report format for creating the mailing lists. These steps do not need to be repeated in order to create the second and subsequent mailing lists from the Quickfile file.

1.Create a Labels Report Format.
In Quickfile, you have the choice of creating either a "Tables Report Format" or a "Labels Report Format."

Create the format by deleting data field categories you don't need for your Word Juggler mailing lists and rearranging the categories you do need, if desired. Do not leave any blank lines in the format. The data in these categories ultimately will be plugged into "variable" locations in form letters or envelope printing formats by Word Juggler when merging data files. The report format for my application is as follows:

MAIL ONE
MAIL TWO
STREET ADDRESS
CITY STATE ZIP
FIRST NAME(S)

2.Change Printer Options. Use "Open Apple O" to change the following options:

- No. 12 - Do not stop at the end of each page.
- No. 13 - Do not print report header at top of each page.
- No. 15 - Do not omit line when all entries on line are blank.

3.Save File. Save the file onto the Quickfile data disk. This will save the report format for future use.

CONVERSION - These steps must be followed for each Word Juggler mailing list that is created from the Quickfile file.

1.Select Records. Return to the report format, and use "Open Apple R" to select the records you want in your first mailing list. Change the record selection rules from "all records" to "SORT CODE contains x" where x is the code corresponding to the desired list.

2.Print Report to Disk. Remove the Quickfile data disk from drive 2, and insert a disk for temporary file storage. Use "Open Apple P" to print the file to the disk in drive 2. When printing to the disk, you will be asked for a pathname. Let's say, for example you use ".D2/LIST.X".

3.Exchange Program Disks. Remove the Quickfile program disk from drive 1, and boot the Word Juggler program disk. Select No. 7 from the Word Juggler menu and change the prefix from ".D1/" to ".D2/".

4.Load the Data File. Select No. 3

from the Word Juggler menu and load the file which was printed to the temporary storage disk by the Quickfile program in Step No. 2 above. For the pathname, type "*LIST.X.ASCII".

5. Modify the Data File. Add the normal information needed at the top of any Word Juggler data file (mailing list) that is to be merged with form letters or used to print envelopes or mailing labels. This information consists of one line containing a digit representing the number of fields in each record. The following lines contain the variable names for each field. These variable names must be typed exactly as they appear in the form letter or envelope printing format. The information added at the top of each of the Word Juggler mailing lists in my application is as follows:

```
5
<MAIL.ONE>
<MAIL.TWO>
<STREET.ADDRESS>
<CITY.STATE.ZIP>
<FIRST.NAME(S)>
```

6. Store the Data File. Remove the temporary storage disk from drive 2, and insert your Word Juggler data disk. Store this newly created file on your Word Juggler data disk.

This completes the process. Once you become familiar with the conversion steps by performing them a few times, you should be able to create each new mailing list in about 5 minutes.

Editor's Note :

We encourage responses from our members to articles appearing in the Gazette. Please direct all correspondence to us and we will then forward it to the appropriate author for his/her review. We feel that this method will benefit our readers by giving us the chance to publish the answers to questions you might have had about a certain article. Many times these letters themselves become material for articles.

Beginning Business BASIC Lesson 4

by Stan Guidero

First, a correction to the last lesson's "MENU1" program. In lines 120 thru 160, there shouldn't be parentheses around the numbers. Line 120 should read:

```
)120 IF SELECT = 1 THEN 200
```

Do the same for lines 130 thru 160.

Continuing with the "IF-THEN-ELSE" statement for error checking. This is probably the most important step to master when learning the art of programming. It is also one of the most difficult to learn. When trying to make your program "bomb-proof", you may inadvertently create a disastrous bug. So, you must thoroughly check your program for every conceivable problem that may arise.

Let's take our Discount Table program which we saved under the name "DISCOUNT" and load it into memory.

```
)LOAD DISCOUNT
)LIST
```

What would happen if we entered a letter other than A, B, C or D? The results may be unpredictable. Go ahead and run the program and try to input other than the letter that's requested. In the first input, any input besides A, B, C or D will result in no discount being applied. That's because all variables are set to zero when you first run the program. Since "CREDIT" will equal "zero", then the formula [SALESPRICE = PRICE - (PRICE * CREDIT)] will not give you a discount. How do we fix this? Here's where our IF-THEN-ELSE comes back into play. List your program again. Now, add this line:

```
)35 IF DISCOUNT$<"A" OR DISCOUNT$>"D"
THEN GOTO 30
```

What we have done here is to insert the numerical values that are used by most computers to make our checking work. Computers generally do not understand letters, so the letters are converted to numbers by the computer itself. A standard code was established by some smart-thinking engineers and they called the code "ASCII". Basically, each letter is assigned a number in ascending order. Here is a short list of

the letters:

Decimal number -----		Character -----
65	=	A
66	=	B
67	=	C
68	=	D
69	=	E
70	=	F
71	=	G
72	=	H
73	=	I
74	=	J
75	=	K
76	=	L
77	=	M
78	=	N
79	=	O
80	=	P
81	=	Q
82	=	R
83	=	S
84	=	T
85	=	U
86	=	V
87	=	W
88	=	X
89	=	Y
90	=	Z

The lower case letters are numbered from 97 to 122 (a = 97 and z = 122). A complete list of ASCII code is on page 220 of your Apple Business BASIC Reference Manual - Volume 2. For those of you who have always wondered, ASCII is an acronym for American Standard Code for Information Interchange. (Ed. Note: The term ASCII has been adopted by every computer manufacturer except IBM.)

At this point, all line 35 does is to send you back to the INPUT line at 30 where it re-displays the prompt. The main problem with doing this is it doesn't tell the user what he or she is doing wrong. Let's re-type line 35 so we can add an "oops" message:

```
)35 IF DISCOUNT$<"A" OR DISCOUNT$>"D"  
    THEN GOTO 500
```

Here's a short cut built into Business BASIC for those of you who are basically lazy like me. Rather than re-typing line 35, first list line 35. Then, press the "ESCAPE" key and a plus sign will appear in the cursor. This will allow you to use the up, down, right and left arrow keys to move the cursor

anywhere on the screen. Using the arrows, move the cursor over the 3 in line 35 and press the "ESCAPE" key again. The plus sign will disappear. Now, using the right arrow key, scan across the line until you reach the number 30. Then type 500 over the 30 and press "RETURN". List the line again to make sure it worked.

Now add these lines to your program:

```
)200 GOTO 999  
)500 INVERSE  
)510 PRINT "Type only upper case  
        letters A,B,C OR D"  
)520 NORMAL  
)530 PRINT:PRINT "PRESS RETURN"  
)540 GET HOLDIT$  
)550 GOTO 30
```

Line 200 simply skips the error routine if nothing is wrong. Line 500 starts the error routine. "INVERSE" simply reverses the characters to black on white (or green if you have a Monitor ///). 510 prints the actual message and line 520 returns the print back to normal. Line 540 will hold the message on the screen until you press a key. GET is similar to an INPUT, but doesn't wait for a "RETURN". In this case, we hold the program execution until a key is pressed. The next line used is 550 which returns us back to line 30. Now hopefully the user knows what went wrong and how to correct it.

As you know, the Apple /// has a lower case as well as an upper case. This adds one more problem to the programmer: how to handle lower case letters with INPUT statements. The method we used here was simply to tell the user to use only upper case letters. Another method would be to convert the input from lower case to upper case. We do this with a look-up table similar to the one in lines 60 to 90.

```
)31 IF DISCOUNT$="a" THEN DISCOUNT  
    $="A"  
)32 IF DISCOUNT$="b" THEN DISCOUNT  
    $="B"  
)33 IF DISCOUNT$="c" THEN DISCOUNT  
    $="C"  
)34 IF DISCOUNT$="d" THEN DISCOUNT  
    $="D"
```

Now any lower case a, b, c, or d will be converted to upper case. This leaves us with the numerical input at line 40. What if we accidentally type a letter instead of a

number? Run the program and see what you get.

```
)REENTER?
```

Reenter what? This is a built-in error message but it doesn't tell you what your mistake was. We can use this facility to our own advantage. Since this is the most likely error that will occur at this point in the program, we will write a message to inform the user of his or her error. We have a command at our disposal to use here. It's the "ON ERR" command. Type the following into our program:

```
)38 ON ERR GOSUB 600
)190 OFF ERR
)600 INVERSE
)610 PRINT "*** Numbers only please
      ***"
)620 NORMAL
)630 PRINT:PRINT "PRESS RETURN"
)640 GET HOLDIT$
)650 RETURN
```

Line 38 turns on the error routine. It will continually search for an error until we turn it off at line 190. When an error is encountered, such as typing a letter where a number is expected (we used a numerical variable "PRICE" for our input), the program will jump immediately to line 600 for handling. Lines 600 thru 650 are somewhat the same as 500 to 550 except we use a "RETURN" statement for our GOSUB.

One note of caution. ON ERR will look for any of the built-in error messages so only use it when the chances of an unexpected error occurring are low. Incidentally, CONTROL-C is considered an error so while ON ERR is on, try not to stop the program with CONTROL-C.

For more detailed information on error messages, see the Business BASIC Reference Manual, pages 121 to 124. A list of errors covered by this command is in the appendix, pages 226 thru 235. A list of codes is on page 234. These codes may be used in your error handling routine to handle specific problems. The error code is automatically placed in a special variable called ERR. Here again we can use our old stand-by, the IF-THEN-ELSE statement.

```
)600 IF ERR=254 THEN 605 : ELSE
      IF ERR=255 THEN END
)605 INVERSE
```

Line 600 checks on two error codes. 245, which is the REENTER code, sends you to line 605 that tells you that it wants a number. 255 checks if you tried stopping the program with CONTROL-C. This allows you to stop if you wish. A word of warning. Be careful in using ERROR handling. You could cause more bugs than cure them.

Before closing this lesson I have one more change to make in our program. You probably noticed that sometimes the discount price doesn't come out in a dollar and cents form. Here is a correction:

```
)120 PRINT "The discount price is ";
)130 PRINT USING "$$#.2#";SALESPRICE
```

Well, that should whet your whistle until next time when we get up to our necks in print formatting. Maybe we'll take a look at the newly improved Business BASIC from Apple. Till then, Happy Programming!

Apple Speller ///

by
John Eric Holms

I am a free-lance writer and novelist living out in the wilds of New Mexico. Having purchased my Apple /// to act as a word processor, I would like to comment on a topic of interest to people who use the Apple for writing, namely, the Apple Speller /// program.

The Speller comes with three discs: one is the program and the utilities, and two discs are the Random House Concise Dictionary. The instruction manual is one of the best I have seen. The program works with Applewriter ///, Pascal Text Editor, and any ASCII text file.

How It Works

Apple Speller /// reads the document you want checked off the disc the document is on. If, like me, you have only one disc drive, you'll have to exchange document and dictionary discs frequently throughout the spelling check. The computer then makes an alphabetical list of all the words in the document, "a, an, and,..." along with the number of times the word appears. You can view this list, or print it out, if you desire.

The program compares all the words in the text document with the first version of the Random House Dictionary and tells you how many words are left over that it can't find in the dictionary. You then have the option of running another dictionary search with another disc (the second disc provides a list of less common words from the R. H. Dictionary, called "Random.House.2") or going to a line-by-line check.

Here the screen shows three lines of text with the offending word highlighted, a bit of blank screen for responses and a command menu. You can tell the computer to ignore the word and go on to the next (somebody's name, for instance), or mark the word for later attack by Applewriter (the computer inserts "¶" in place of the last letter of the word, making it easily available to your "Find" command).

You can set the word aside to be entered into another dictionary so you won't have to bother with it in subsequent document checks (the names of all the characters in a novel, for instance). You can also replace the word with the correct one, if you know it, and, most fun of all, you can ask the computer to guess what word you wanted!

Apple Speller /// then proceeds as follows. First, it assumes that the first letter is correct. (It loses its bet with me on that one occasionally!) Then it transposes every pair of letters after that. It interposes every letter of the alphabet between each pair of letters in the word and then removes each letter of the misspelled word one at a time. The program then compares the results of each manipulation with the dictionary list and any matches are shown to you.

You can then replace the word in the text with one of the listed words. If your computer is 256K, you can put in any of the words. Otherwise you can only replace with a word of the same length or shorter length, since the lesser memory will not support the text, program and dictionary at the same time. On my poor 128K ///, I have to "mark" each word with a "¶" and "Find" it later.

Special Features

The program allows you to save words from your document and put them on your own dictionary, so that further documents can be compared to it. This allows you to list commonly used names and terms from your office. You don't have to look at them each time you correct a letter or a memo. The

time required to do each of these operations is only a few minutes (word list, dictionary #1) or a few seconds (guess what word I want).

For each of the novels I'm working on, I have a little dictionary with the character's names, the place names, and any jargon words. I write science fiction, so I have a dictionary that contains "hyperspace," "laser rifle," and "chitinous." The Concise Dictionary word list is limited. A legal and a medical dictionary are available from Apple.

Some Program Flaws

You can only save a word to be added to a dictionary if it is correctly spelled in the document. I wish there was some easy way to correct a word ("replace") and, at the same time, "add" it to my personal dictionary.

If you use a real word in the wrong place, "wit" for "with," the program won't catch it, since both words are in the dictionary. You must still proof read the text for meaning. The annoying little typos that only show up on the finished copy, however, get caught by the Speller program.

I've had one bug show up in the program. If the document is too long for complete wordlisting, the program processes the words in its memory and then goes back to the document and picks up where it left off, fills the memory again, and compares that list to the dictionary.

On three occasions I've had the program working on its third memory-full of words suddenly shout "I/O Error!" It then goes back to the main menu and insists on processing the document all over again from the beginning. This has resulted in going through three sweeps of the first part of the manuscript before I could get the whole thing checked. I haven't figured out how to get out of that one.

In Conclusion

Wow, have I been waiting for this program! If you don't have 256K and two disc drives, Apple Speller will make you wish you did. And if you write a review of Apple Speller, you discover that the word "speller" isn't in its dictionary!

open apple gazette



1850 Union Street #494
San Francisco CA 94123

Business Apple Group, Inc.

CLUB INFORMATION

MEETINGS

Meetings are held at 7:30 PM on the third Wednesday of each month. Call Don Norris for current meeting location.

MEMBERSHIP

Annual membership dues are \$30 from the date application received. Your check payable to the Business Apple Group, Inc. may be mailed to the address below.

OPEN APPLE GAZETTE POLICY

All manuscripts, photographs, and other materials are submitted free and released for publication. They become the property of the Business Apple Group, Inc. and the Open Apple Gazette. Authors should clearly mark all material submitted for publication so that credit may be given.

The publishers/editors do not necessarily agree with, nor stand responsible for, opinions expressed or implied by other than themselves in this publication.

The Business Apple Group, Inc. is a non-profit organization comprised of, and supported by, Apple /// owners and users. We endeavor to aid other Apple /// users through this educational publication - "OPEN APPLE GAZETTE." Address all inquiries to: Business Apple Group, Inc., 1850 Union Street #494, San Francisco, CA 94123.

REPRINT POLICY

All articles appearing in the Open Apple Gazette not

copyrighted by the author may be reprinted by another non-profit Apple user group so long as proper credit is given to both the Open Apple Gazette and the author. Proper credit is defined as article title, author, and the words "Printed from VOL X, NO Y of the Open Apple Gazette." Permission to reprint a copyrighted article may be obtained by writing to the author c/o the Business Apple Group, Inc..

ARTICLE SUBMISSION POLICY

The Open Apple Gazette welcomes any and all articles dealing with the Apple /// Computer and its associated hardware and software. Articles should be submitted on diskette as an ACSII text file, such as those produced by either Word Juggler, Apple Writer /// or the Pascal Editor. Typewritten double spaced articles are also acceptable.

OFFICERS

PRESIDENT Don Norris (415) 921-3774
SENIOR EDITOR Stuart Henderson

Back Issues

Volume 1, Numbers 2-6 \$4.00 each
Volume 2, Numbers 1-2 \$4.00 each

Mail Requests for Back Issues to:

Open Apple Gazette
1850 Union Street, #494
San Francisco, CA 94123

-///-