

A journal and exchange of Apple II discoveries

InCider wins, Beatles sue

A+ magazine has been sold by Ziff-Davis Publishing Company to IDG Communications/Peterborough, which will merge it into its *inCider* magazine after A+'s May 1989 issue, the companies announced March 8.

Editorial offices for the merged publication will be in Peterborough, N.H. and will be built around the current *inCider* staff. *InCider* will take over the subscription liabilities of A+, will pay Ziff-Davis an undisclosed amount of cash, and may also pay Ziff-Davis a bonus based on the future performance of the combined publication.

The first merged issue will be ready in time for AppleFest in early May. Subscribers to both publications (of which there were fewer than the companies expected) will have the option of having their expiration date moved back to reflect both subscriptions.

Ziff-Davis said it decided to leave the Apple II market because its publications primarily focus on business users of personal computers. In its view the Apple II is a home and education machine. In fact, Apple IIs are successfully used in many small businesses and their presence in the market would be much greater if Apple hadn't discouraged their use and ignored their potential for the last five years.

Paul Statt, Senior Editor of *inCider*, said the feeling at IDG Communications/Peterborough is that *inCider* prevailed because over the years it has emphasized both subscription and advertising income, whereas A+ was more dependent on advertising alone, and because *inCider* has always been an Apple II-only publication, whereas A+ lived part of its life as a combined Apple II/Macintosh magazine. *InCider*'s new circulation is expected to exceed 200,000.

Beatles claim trademark infringement. The Beatles initiated legal proceedings against Apple in London on February 21, claiming that a 1981 trademark agreement between Apple Computer and Apple Corps, the holding company formed by the Beatles to produce its records, has been violated.

Apple Corps says the agreement barred Apple from using its Apple trademark on musical products and that Apple has broken the accord with the introduction of various equipment that has music-related features, including the Apple IIs and the Apple MIDI interface.

According to the *Wall Street Journal* (February 22, page B11), the dispute arose after former Beatle George Harrison noticed an advertisement touting the musical side of Apple's computers. Harrison, who was aware of the contract, called in the lawyers.

According to information in the suit, it appears Apple was also aware of the problem, as it has filed petitions in at least four countries trying to convince authorities that Apple Corps's trademarks are invalid.

Heaven's Gate. Personal computers have unlimited potential for saving us time. But they never do. This paradox has now been explained by an amazing scientific discovery.

According to Dr. Kathleen O'Connell, a disorder called Computer Perseveration Syndrome is responsible for the problem. Dr. O'Connell says "perseveration" means the uncontrollable repetition of some activity or action.

Perfection seems like a real possibility to personal computer users, according to Dr. O'Connell. Before computers were widely available, people would devote as much time to a project as it deserved, she says.

Now they do a project in no time, but then spend hours repetitively refining it. Dr. O'Connell reports her husband once ran twelve drafts of a ten-word letter through his word processor.

Manifestations of the syndrome are also seen in spread-sheet users who fine tune a template for hours so it will be easy to use next time—and never use it again; in folks who spend days creating databases on who sent them Christmas cards, but who never have time to send any themselves; and in programmers who only write programs that make it easier to write programs.

Some researchers think the Computer Perseveration Syndrome is a serious problem that must be given utmost consideration by the scientific community. Others think it is a sign of approaching nirvana.

What's the DIF?

Besides the Computer Perseveration Syndrome, another major reason that personal computers often don't deliver on their potential to save us time is the widely discussed Computer Incompatibility Situation. In part, this problem is the price we pay for progress. New engineering feats naturally result in wares that are incompatible with their predecessors. Strict compliance to a Computer Compatibility Standard would stop innovation.

Mention of the Computer Incompatibility Situation usually brings to mind the many unique dialects of Basic; the several popular but disparate operating systems used just on the Apple II; and the many dissimilar command codes printers use.

Less noticeable, but just as big a time waster, are programs that save data—your work—in unique file formats. Programs that use only non-standardized, undocumented file formats diminish the value of your data.



If you want to use the data you have painstakingly entered or created with another program, your choices are:

- retype it all,
- deduce how the file is formatted and write a program to reformat it, or
- deduce how the file is formatted and modify the second program so it can access the non-standard file (or modify the original program so it uses a standard file format).

Each of these choices involves a massive waste of your time.

Programs that use standard file formats, on the other hand, make it easy to exchange data among programs. Text files are often used as a common denominator for sharing data among programs such as word processors. When the task at hand is to store data organized in a special way, however, simply using a text file usually isn't enough.

For example, imagine the following table is stored, just as it appears here, in a simple text file. It would be a fairly complex programming task to write a routine that could read a table like this and store it in an array—particularly if the program had to be able to adapt to tables with various numbers of rows, columns, headlines, and so on; as is often the case.

Monthly payment per \$10,000 of home mortgage

	-----interest rate-----				
years to pay	8	10	12	14	16
15	95.57	107.46	120.02	133.17	146.87
30	73.38	87.76	102.86	118.49	134.48

Better File Diffusion. The company that originally developed the first spreadsheet program, *VisiCalc*, also developed and documented a special type of sequential text file for storing these kinds of data. This type of file is called a DIF (Data Interchange Format) file.

VisiCalc and many other spreadsheet programs are able to read and write DIF files. In addition to spreadsheets, DIF file compatibility has been built into many accounting, graphics, and statistics programs, as well as some word processors and data base managers. The AppleWorks spreadsheet and data base modules can both read and write DIF files.

Programs that use the DIF format can easily exchange data with each other. Information can be stored online in DIF files, then downloaded and used by a variety of users. This not only saves time, but also avoids the inevitable mistakes that would corrupt the data if it had to be retyped by hand.

Equally important, it is quite easy for you, as a programmer, to get at the data yourself. DIF was created to be easy for both beginners and advanced programmers to use and understand. Programs using DIF files can be written in any language—Basic, assembly language, even Pascal. And DIF files are not dependent on the features of any particular computer or operating system.

Reading and writing DIF files with Applesoft programs is so easy all the necessary subroutines are included in this issue.

A very important aspect of DIF files is that it is much easier to read a DIF file than it is to write data input routines as good as the ones found in a spreadsheet. This means that for programs you write for yourself you can save tons of time by using a spreadsheet program to enter your raw data, then transferring it to your program with a DIF file.

All your own program has to be able to do is read the DIF file. You save time because you don't have to write complex data input routines. And you save time because you will be able to enter, check, and edit your data much faster with a spreadsheet program than you could possibly do with your own data input routines.

Two Difficult words. The only thing that's hard about DIF files is that two very scary words were chosen to describe how the numbers in a table are organized. Normally the words people use for this are *row* and *column*. The people who invented DIF files didn't want to use these, however, because they figured DIF files should reflect the fact that a table organized like the one above and one organized like the one at the top of the next column are identical from a data standpoint:

Monthly payment per \$10,000 of home mortgage

interest rate	years to pay	
	15	30
8	95.57	73.38
10	107.46	87.76
12	120.02	102.86
14	133.17	118.49
16	146.87	134.48

Rather than use easy terms such as row and column, then, the DIF developers chose...are you ready?...*vector* and *tuple*. In the above table, for example, the columns can be thought of as vectors and the rows as tuples. Or, if you prefer, think of the rows as vectors and the columns as tuples.

In this example, if you decided to let the columns, *years to pay*, be vectors, then you would have two vectors and five tuples.

On the other hand, if you decided the rows, *interest rate*, would be the vectors, then you would have five vectors and two tuples. *It doesn't matter which is which, just don't let those words scare you.*

Anatomy of a DIF file. A DIF file consists of two major parts. The first of these is the *header*. It contains information you can use to confirm that the file really is a DIF file; it tells how many vectors and how many tuples are in the file; and it can tell some other stuff we'll look at later.

The second major part of a DIF file is the *data*. The data section consists of a number of *data values*. Each data value includes three pieces of information written in two lines. The three pieces are the *type indicator*, the *number value*, and the *string value*. Here are some sample data values:

```

1,0      1,0      1,0
NO       ""       "April Fool's Day"

0,3      0,-913   0,95.57      0,1.0026E20
V        V        V        V

0,0      0,0      0,1          0,0
NA       ERROR   TRUE        FALSE

-1,0     -1,0
BOT      EOD

```

The *first number on the first line* of each data value is the type indicator. This number can be *1*, which indicates the value is a character string; a *0*, which indicates the value is numeric; or a *-1*, which indicates a special data value. (There are only two of these special values. The first one shown, *BOT*, marks the "beginning of a tuple"; the second, *EOD* marks the "end of data," which is also the end of the file.)

The *second number of the first line* of each data value, the number value, is *0* if the data value is special or a character string. If the data value is numeric, on the other hand, this is the actual data (inside the file the "actual data" is stored in as ASCII characters). The number value may be signed (+ or -) and may contain a decimal point. It may also contain an exponent of a power of ten. In this case the value is followed by the letter "E" and the signed or unsigned exponent, as shown above.

The *second line* of each data value always contains the string value. If the data value's type is "special" (-1), then this string can be only *BOT* or *EOD*, as mentioned earlier.

If the data value's type is "numeric" (0), then this string can be any one of the five following upper-case, unquoted words:

string value	meaning	appropriate numeric value
V	normal numeric value	actual data
NA	value not available	0
ERROR	value result of invalid calculation	0
TRUE	logical value	1
FALSE	logical value	0

Finally, if the data value's type is "character string" (1), the second line contains the actual string value. DIF file character strings are not allowed to contain control characters or quotation marks. If the string

is a single word, it may be entered directly on the line. If the string contains blanks or special characters, then it must be enclosed in quotation marks. If the string is null or empty, then the string-value field should contain two quotation marks with no space between them.

Now that we know everything there is to know about DIF file data values, let's look at an entire DIF file data section. This one is based on our home mortgage tables--"years to pay" are the vectors:

```

-1,0      -----
BOT
0,95.57
V          tuple 1 (interest rate 8%)
0,73.38
V
-1,0      -----
BOT
0,107.46
V          tuple 2 (interest rate 10%)
0,87.76
V
-1,0      -----
BOT
0,120.02
V          tuple 3 (interest rate 12%)
0,102.86
V
-1,0      -----
BOT
0,133.17
V          tuple 4 (interest rate 14%)
0,118.49
V
-1,0      -----
BOT
0,146.87
V          tuple 5 (interest rate 16%)
0,134.48
V
-1,0      -----
EOD
    
```

The important things to notice are that each tuple begins with the BOT special data value and contains a data value for each vector. To keep the format simple, all tuples in a DIF file must have the same number of vectors (and all vectors must have the same number of tuples). Also notice the EOD special data value at the bottom of the listing. This marks both the end of the data section and the end of the file itself.

As mentioned earlier, every DIF file has a header section in addition to a data section. The header section consists of a number of header items. Each header item includes four pieces of information written in three lines. The four pieces are the item's topic, the vector number the item applies to, a numeric value, and a string value. Here are some typical header items:

TABLE	VECTORS	TUPLES	DATA
0,1	0,14	0,3	0,0
""	""	""	""

LABEL	COMMENT	UNITS
1,0	3,1	0,0
"April Showers"	"SOME DATA ESTIMATED"	"metric tonnes"

The first line of each header item is the item's topic. The first number on the second line of each header item is the number of the vector to which the item applies. For example, the 'label' item shown above applies only to vector one, while the 'comment' item applies only to vector three. When the vector number is zero, as in the 'units' item, then the item applies to all vectors; that is, to the entire table.

The second number on the second line of the header items is a numeric value. For example, in the "vectors" and "tuples" items, this number indicates how many vectors and tuples are in the file. In the "label," "comment," and "units" items, it indicates a line number, which we'll look at in more detail later.

The third line of the header items always contains a string value. If the item has no associated string, then a null string is used, as in the "vectors" and "tuples" items.

Every DIF file header must contain the header items called "table", "vectors", "tuples", and "data". The "table" item must always be the first item in the file. The numeric value in the table item is the DIF version number. To date there is only one version, so this number must be "1". The table item may also contain the title of the file in its string area, if desired. Here's a typical DIF file header section:

```

TABLE
0,1
""
VECTORS
0,2
""
TUPLES
0,5
""
DATA
0,0
""
    
```

The 'data' item must always be the last item in the header section of the file. It indicates that the data section of the file follows.

Otherwise, the order of the header items isn't supposed to matter, except that the "vectors" item must come before the use of vector numbers in other items.

In addition to the four required header items, DIF files may contain optional items such as "labels" and "comments". Any program reading a DIF file, however, should be able to operate without any optional items. If a reading program requires the information provided by an optional item, it should prompt the user to supply the missing information rather than require the item itself to appear within the file.

Likewise, any program that reads DIF files should be able to ignore optional header items the program doesn't use. Unfortunately, AppleWorks can't do this. It chokes on DIF files that include optional header items (see "AppleWorks Difficulties," August 1986, page 2.56 and "Difficulties revisited," September 1986, page 2.59).

In the "label" and "comment" items, the numeric value indicates a line number. Thus, a multi-line comment is possible:

```

COMMENT
0,1
"Data from sources"
COMMENT
0,2
"thought to be reliable"
COMMENT
0,3
"but not guaranteed."
    
```

Optional header items for DIF files that have been defined include "periodicity", "majorstart", and "minorstart", which provide additional information about time series data; "size", which provides programs such as data base managers the option to give vectors fixed field lengths or sizes; and several more.

If necessary, it is also possible to develop new optional header items to fulfill the needs of specific programs. At one time Software Arts had a "DIF Clearinghouse" to keep track of such enhancements to DIF files, but I don't think it's around anymore.

Basic Differentials. The following Applesoft subroutines, which we've also included in separate files on this month's disk, show how easy it is to write and read simple DIF files. These routines can save the numeric data in an array into a DIF file and read a DIF file's data into an array. For simplicity's sake, neither routine supports string or logical variables (true/false). Also beware that the read-routine will stop execution if it encounters an NA, ERROR, or null value (blank spreadsheet cell) in a DIF file. Enhancements to the routines in these

areas are, as they say, left to the reader.

```

10000 REM          DIF writer
10005 REM
10010 REM  Before calling this subroutine the following variables
10020 REM      must be set by the main program:
10030 REM      D$ = control-D
10040 REM      F$ = filename to be used
10050 REM      V = number of data "vectors"
10060 REM      T = number of data "tuples"
10070 REM      A(V,T) = actual data to be stored in DIF file

10080 NULL$=CHR$(34)+CHR$(34) : REM "" (null string)

10090 PRINT
10100 PRINT D$;"OPEN";F$
10110 PRINT D$;"DELETE";F$
10120 PRINT D$;"OPEN";F$
10130 PRINT D$;"WRITE";F$

10140 PRINT "TABLE"
10150 PRINT "0,1"
10160 PRINT NULL$

10170 PRINT "VECTORS"
10180 PRINT "0,";V
10190 PRINT NULL$

10200 PRINT "TUPLES"
10210 PRINT "0,";T
10220 PRINT NULL$

10230 PRINT "DATA"
10240 PRINT "0,0"
10250 PRINT NULL$

10260 FOR I=1 TO T
10270 : PRINT "-1,0"
10280 : PRINT "BOT"
10290 : FOR J=1 TO V
10300 : : PRINT "0,";A(J,I)
10310 : : PRINT "V"
10320 : NEXT
10330 NEXT

10340 PRINT "-1,0"
10350 PRINT "EOD"

10360 PRINT D$;"CLOSE";F$
10370 RETURN

11000 REM          DIF reader
11005 REM
11010 REM  Before calling this subroutine the following variables
11020 REM      must be set by the main program:
11030 REM      D$ = control-D
11040 REM      F$ = filename to be used
11050 REM      A(x,y) = array must be DIMed large enough to hold DIF data
11060 REM
11070 REM  After calling, the following variables will be set:
11080 REM      V = number of data "vectors"
11090 REM      T = number of data "tuples"
11100 REM      A(V,T) = actual data found in DIF file
11110 REM
11120 REM  NOTE: This routine requires the DIF file to contain only
11130 REM      numeric data. String, logical, or null values, as
11140 REM      well as ERROR and NA, will stop execution of the routine.

11150 PRINT
11160 PRINT D$;"OPEN";F$
11170 PRINT D$;"READ";F$

11180 INPUT X$ : INPUT X,Y : INPUT Y$
11190 IF X$ <> "TABLE" OR Y <> 1 THEN PRINT "Not a DIF file." : GOTO 11390

```

```

11200 INPUT X$ : INPUT X,Y : INPUT Y$
11210 IF X$ = "VECTORS" THEN V=Y
11220 IF X$ = "TUPLES" THEN T=Y
11230 IF X$ <> "DATA" THEN GOTO 11200

11240 FOR I=1 TO T
11250 : INPUT X,Y
11260 : INPUT X$
11270 : IF X <> -1 OR X$ <> "BOT" THEN 11370
11280 : FOR J=1 TO V
11290 : : INPUT X,A(J,I)
11300 : : INPUT X$
11310 : : IF X <> 0 OR X$ <> "V" THEN 11370
11320 : NEXT
11330 NEXT

11340 INPUT X,Y
11350 INPUT X$
11360 IF X = -1 AND X$ = "EOD" THEN 11390

11370 PRINT "Unexpected data encountered in file."
11380 PRINT "File input stopped at Tuple ";I;," Vector ";J; "."

11390 PRINT D$;"CLOSE";F$
11400 RETURN

```

Here's some cheap advice for those of you who take the suggestion we made earlier to use a spreadsheet program to enter data rather than writing input routines yourself. When saving your data in a DIF file with the spreadsheet program, save only the portion of the worksheet that contains the columns and rows of numbers. Make sure there are no blank cells, NAs, or ERRORS in the saved portion. Alternatively, you can enhance the DIF reader subroutine so it doesn't trip on these sorts of things.

(This article originally appeared in my DOSTalk column in *Softalk* magazine in February 1984, page 65. Most of the stuff I wrote for DOSTalk has been reprinted in either the *DOSTalk Scrapbook* or *ProDOS Inside and Out*, but this article missed both those books. We've had a few questions about DIF files lately, so it seemed an opportune time and place to run the article again. Likewise, the Computer Perseveration Syndrome piece originally appeared in the very fat December 1983 issue of *Softalk*. And now, for posterity's sake, it's in a very thin A2-Central, too.)

Miscellanea

Two new products written by Glen Bredon prove the text screen is alive and well under GS/OS. One is a GS/OS version of the *Merlin* assembler, published by Roger Wagner. The other is a GS/OS version of *ProSEL*, published by Bredon himself. Both programs are knock-outs.

The GS/OS version of Bredon's assembler is called *Merlin 16-Plus*. It will be added to the assembler package Roger Wagner sells, which will continue to be called *Merlin 8/16*. The package will retail for \$125 and will include four versions of *Merlin-6502* DOS 3.3 (*Merlin*) and ProDOS 8 (*Merlin Pro*) versions and 65802/65816 ProDOS 8 (*Merlin 16*) and GS/OS (*Merlin 16-Plus*) versions.

Merlin 16-Plus includes many features new to the *Merlin* package, including compatibility with Apple Programmer's Workshop 'object' or OMF files. Routines assembled by *Merlin* can be combined with routines assembled or compiled by APW languages using either the *Merlin* or the APW linker. Tests with Apple's demonstration desktop program *HodgePodge* show *Merlin 16-Plus* to be about five times faster than the APW assembler.

Owners of the previous version of *Merlin 8/16* can get a *Merlin 16-Plus* update for \$45.95 plus \$4 shipping. The update includes two disks—one holding *Merlin 16-Plus*, the other holding sample programs, including a *Merlin*-compatible version of Apple's *HodgePodge*; *Nifty List*, David Lyon's 65816 disassembler; *Internals*, a program by Ken Kashmarek that displays interesting data about what's where in your ligs; and *Skeleton*, a thousand lines of code that just about all GS/OS assembly language programs need just to get started and stopped successfully.

Owners of any other commercial Apple II assembler (including ear-

lier versions of *Merlin* can send Roger Wagner Publishing the title page or original disk from that product along with \$60 plus \$4 shipping and get the complete *Merlin 8/16* package. (Roger Wagner Publishing, 1050 Pioneer Way, Suite 'P', El Cajon, CA 92020 619-442-0522.)

Bredon's ProSEL-16 is a GS/OS program selector that does all the same things the *Finder* does only much faster and much easier. It does take longer to learn how to use, but the time is well-spent. Many of the utilities that were separate under the ProDOS 8 version *ProSEL*, such as *Cat Doctor*, are now part of the main program. *ProSEL-16* also includes a new incremental hard disk backup program, as well as the current version of *ProSEL-8*. Updates for *ProSEL-8* owners are \$20. New copies of *ProSEL-8* (for classic Apple users) are \$40. New copies of *ProSEL-16* (for IIgs users; includes *ProSEL-8*) are \$60. (Glen Bredon, 521 State Rd, Princeton NJ 08540; outside US and Canada, add \$5 shipping and handling fee. No credit cards, purchase orders without checks, or checks from banks outside the U.S.)

And CP/M hasn't died either. A new company called Slot 8 Associates has just introduced a new CP/M co-processor board for Apple IIs, called CardZ180 (say cards-one-eighty). The card uses a 9.22 Mhz Hitachi 64180 microprocessor, which makes CardZ180 about 5.5 times faster than traditional CP/M cards for the Apple II, and comes with 256K of RAM. The company says the card is compatible with all common storage devices and accelerators and will work in slot 3 on a Iie or IIgs (\$349, Box 10098, Rochester, NY 14610).

Meanwhile, Borland has assigned the CP/M version of its Turbo Pas-

cal to a company called Alpha System Corporation. The company provides v3.01a of the program, with a revised 232-page manual, on Apple II disks for \$60 plus \$3 shipping (711 Chatsworth Place, San Jose, CA 95128 408-297-5594).

The Arizona Apple User Group will hold its third annual Apple Fiesta at the Safari Hotel in Scottsdale June 2, 3, and 4. The Fiesta consists of seminars, classes, listening to and talking with guest speakers, and exhibits. Entrance fees are \$7 at the door, and hotel rooms are \$35 a night. A2-Central will be an exhibitor and both Dennis Doms and I will be guest speakers. For more information call 602-498-4357. For hotel reservations call 602-945-0721 or 800-834-4356.

Microdot is a new ampersand-command-based replacement for Basic.system from Kitchen Sink Software. It's smaller than Basic.system so there's more room for your program. It includes commands for program overlays and for disk formatting that aren't in Basic.system. (\$29.95 from Kitchen Sink Software, 903 Knebworth Ct., Westerville, OH 43081 614-891-2111).

The Association for Computers in Mathematics and Science Teaching publishes three refereed, non-commercial quarterly journals some of you may be interested in. They are the *Journal of Computers in Mathematics and Science Teaching* (\$28/yr), the *Journal of Computing in Childhood Education* (\$36/yr), and the *Journal of Artificial Intelligence in Education* (\$26/yr). (ACMST, P.O. Box 60730, Phoenix, AZ 85082 602-952-2712).



Ask (or tell) Uncle DOS

The gang at Redmond Cable ("IIgs and 5.25 drives", March 1989, page 5.14) called to say that some users of their adapter for plugging old-style 5.25 drives into the connector on the back of 3.5 drives have fried their equipment by not turning their computer off when making the connections. You can usually get away with plugging and unplugging some kinds of cables while the computer's on (serial cables, keyboard cable), but it may be habit forming and isn't recommended. That's because plugging or unplugging a disk drive cable or a card in a slot while the computer is on almost always damages something. Don't do it!

IIgs battery problems

I purchased one of the first Apple IIgs computers made and now I have a problem. The battery is dead. The volume control doesn't work and the control panel always reverts to the original system configuration when I turn the system off.

The *Apple IIgs Owner's Guide*, in Appendix A (page 108), says "the battery should last between five and ten years. If it ever runs low the control panel program restores the original system configuration. If this happens, have your authorized Apple dealer replace the battery."

I called my authorized Apple dealer and spoke to the service department. I was told they would have to send my motherboard back to

Apple and get a replacement. This, they said, would take two to three weeks or more as Apple is slow. My problem is that it's tax season and I can't be without my computer that long. What can I do to get my system fixed faster?

In addition, my dealer said the cost would be a little over \$200 for the motherboard exchange. This seems like a fairly high price to pay for a dead battery.

Leon Munn
Inglewood, Calif.

Incredible. We have confirmed that Apple dealers who go by the technical service book have no option other than doing a \$200 motherboard exchange to replace a dead IIgs battery. The guys who designed the machine say the old battery can be clipped off the motherboard and replaced by any technician worth his soldering iron. That message hasn't gotten through Apple's channels and out to dealers, however. Try a radio or tv repair shop—they can probably fix it for you quickly for not much more than the price of the battery. (Apple doesn't provide batteries as service parts, so you'll have to find a replacement locally.)

I don't know how we can talk Apple into changing its policy on this, but taking a few dealers to small claims court for charging over \$200 to replace a dead battery might wake somebody up. It smells like fraud and it looks like fraud, even if it's only a failure to communicate.

AppleWorks snapshots, cont.

In last month's issue a reader asked how to incorporate AppleWorks' own menus into a word processor document (page 5.15). One solution is to insert Pause Here at the point in your document where you want the menu. After the printer stops, hit ESCape and bring the menu you want in the document onto the screen. Then press open-apple-H(ardcopy) to print it. After the menu is printed, return to the word processor and Print from the Cursor.

Les Dyer
San Carlos, Calif.

I've successfully incorporated screen snapshots into documents by using New Page and Pause Each Page options to stop printing at the point at which I want the snapshot, then using open-apple-H(ardcopy) to print the menu I want. To resume my printout, I go back to the word processor and take up where I left off by using Print from This Page. It takes a little planning but it is much quicker and neater than physically cutting and pasting screen dumps into the document.

Michele Olson
Grand Rapids, Minn.

Regarding the request by Richard Marchiafava for a way to capture a snapshot of any AppleWorks menu or screen, my company publishes two disks of macros, each of which contains a different macro which accomplishes this task. The disks are called *Ultimate Macros I* and *Ultimate Macros II* and are available for \$9 each.

Beverly Cadieux
AppleWood Micro Software
3103 Lake Stream Drive
Kingwood, Texas 77339

I've also discovered that MacroTools II from Beagle Bros includes a Screen Shot macro, by Mark Munz.

A sparse argument

Last month you suggested that programmers avoid large sections of embedded zeros in a file in order to prevent GS/OS from making the file sparse. Wouldn't it be better to let GS/OS do its thing to conserve disk space and ask programmers to do the following? First, avoid using the number of blocks in a file for identification purposes and second, if direct block-level access to files is absolutely necessary, treat \$0000 entries in index blocks as blocks full of zeroes. If all access is through ProDOS or GS/OS calls, no other special handling for sparse files would be necessary.

David Lyons
North Liberty, Iowa

Back to Basics, cont.

I have mixed feelings about the issue expressed by Hannelore Maddox in the March issue, and about your reply.

Granted, it would be much harder to make hardware mods to the GS. Can you imagine trying to get in there with a soldering iron, the way we did on the II-Plus for the shift-key mod?

But programming on the IIgs is as simple as it was on the II-Plus, if that's the level at which you'd like to approach it. Applesoft is still in ROM. And the programs you write will run three times as fast if you choose, or at the same speed.

Let's look at the larger issue of programming this beast to the "Human Interface" standards which Apple would like us to use. Complicated? Yes. But as an intermediate level programmer, I have gotten much farther with *Merlin 16* and the 65816 than I ever did years ago with the 6502. Several reasons: the IIgs toolbox gives the user an incredible amount of power. Macros make modern assemblers like *Merlin* and *APW* fantastically more friendly. Apple and third parties provide much better documentation for 65816 programming than ever was available for the 6502. Periodicals like *A2-Central*, *Call-A.P.P.L.E.*, and *The Sourcerer's Apprentice* disseminate the knowledge that programmers need.

GENie's A2PRO forum is visited regularly by a small group of talented and dedicated programmers who seem to delight in helping others with their programming troubles. Users helping users, sharing the power. That's what the Apple II has always been about.

I agree, the IIgs does not have enough power to run its own interface. We all still have our fingers crossed that this problem will be addressed in 1989.

Lorne Walton
Vancouver, B.C.

This month I'll let Dennis respond:

I agree with most of what you said. The exception I'd like to take is that there is an intermediate ground between "8-bit" programs and the toolbox/desktop arena and that is programming in a shell-like environment. I got spoiled by this in Unix; there are times when productivity is better when you can run a file through a set of small, simple processes rather than having to layer the desktop over the program (plus, you get to tell the program what to do, rather than having to depend on the desktop programmer's priorities). I know Apple doesn't discourage this approach, but they certainly don't encourage it, either, which I think is a shame. I'd rather have a choice, and I don't think I'm alone.

I do think there is a rather great hurdle in learning the toolbox, but TaskMaster makes this "lower" than the one on the Macintosh. Whether the time spent in designing a program around the desktop interface is well-invested or not is sometimes debatable; I can see a major need for a dBase III style of database for the II, for example, and I think trying to make everything in such a program menu-driven would defeat part of the strength of the program (the scripting functions).

Apple memory card problems

Our school purchased 20 IIgs computers with two extra sets of memory chips for each. But AppleWorks 2.0 won't run with three banks of memory chips on a card. It just hangs or crash-

es after loading to memory. If I remove one bank, leaving 512K on the card, AppleWorks boots normally. Any idea why?

Robert Hall
Nantucket, Mass

Apple's IIgs Memory Expansion Cards, which have been included as part of the IIgs since mid-March a year ago, don't support three banks of chips. In addition to the one-bank, 256K configuration they are shipped with, they can be configured for 512K or 1 megabyte. For 512K, the 16 sockets in the top row should be full of chips and the 16 sockets in the bottom row should be empty. For 1 megabyte, all 32 sockets need a chip.

In addition, you must put plastic jumper blocks on some pins on the card to complete 512K or 1 megabyte configurations. If you draw an imaginary line parallel to, but about an inch above, the gold edge connector that plugs into the IIgs memory expansion slot, you'll find the pins in the middle of the imaginary line. For a 512K card you must put a jumper block on the lower pair of pins. For a 1 meg card you must put jumper blocks on both pairs of pins.

User-defined characters

I can't get the "user-defined characters" function of my Epson FX-80 printer to work. My system is an Apple IIe, Grappler-Plus, and FX-80. I want to be able to define a few Greek letters and math symbols for use in scientific word processing. Do you know of any software available to do this defining for me? My local Epson dealer has no experience with advanced features.

Larry Mehne
Emory, Va.

*Two programs that provide the features you're looking for are **Print-Quick** from Third Wave Technology (\$49.95 11934 Lorain Ave, Cleveland, OH 44111 216-671-8991) and **Power Print** from Beagle Bros (\$39.95 6215 Ferris Square, Suite 100, San Diego, CA 92121 619-452-5500).*

Clean advice

What do you use to clean the heads of your 3.5 drives?

Richard Cichanski
Des Plaines, Ill.

I don't clean the heads of my 5.25 drives, much less my 3.5s. Only one drive, a 5.25, has ever needed to be repaired and that problem wasn't caused by dirt (or by using the back of "single-sided" disks). Most Apple II programs don't use the disk drives enough to get the heads dirty.

Bad mail-merge dates

While trying to print with AppleWorks using mail-merge, date fields that I print are coming out something like "@ 89 A O", instead of the actual date. The dates look fine in the database, and they print fine from the database. This even worked when I did it last month. But now it doesn't. I'm baffled.

I also run a business that re-inks printer ribbons, small or large. We charge \$2.50 for re-inking most ribbons. Re-inking larger ribbons depends on the particular ribbon and length, so we re-ink the first one free to establish a price and so the customer can compare quality.

William R. Hanna
376 Bethel Church Road
New Cumberland, Penn. 17070

AppleWorks stores dates internally in a special format (see our March 1987 issue, page 3.9). When you copy records to the clipboard for mail-merge, the dates are left in this special format. To get them to print correctly, you have to print the records to the clipboard. And to do that, you'll first have to set up a "table" format that includes the categories that appear in your mail-merge document (since it worked last month, in your case you probably already have the format set up and just forgot to use it).

Full-page DB printouts

I want to expand my AppleWorks database printouts to two pages so that I can place military forms and paperwork on my computer and throw my typewriter under the biggest tank I can find.

Jim Holler
Boone, N.C.

The only way to incorporate AppleWorks database information into a two-page document would be to use the word processor and mail-merge functions. You could position the "merge" fields in a multiple-page "form template" the way you like and then use Mail Merge to insert data from the data base into a form printed as a word processor document. This limits you to the 30 categories allowed by AppleWorks and you would need one of the later versions of AppleWorks (2.0 or higher), which include the mail-merge function.

*If you need more fields, you may have to move your data from AppleWorks to a more powerful stand-alone database. For more information, see the description of **DB Master** in this month's catalog, which we decided to carry as a result of the following letter:*

DB Master praised

After attending an excellent presentation by Barney Stone to the Philadelphia Area Computer Society in which he demonstrated *DB Master Professional*, I purchased the program. I have used *DBase III+* on my IBM's for several years because I needed a relational database. After working with *DB Master Professional* for a few weeks, I can say that if I had to make hardware choices based upon software, I would have bought Apple II's for work. Not only does *DB Master* do everything I need a database program to do, it does it without having to know how to program. The report generator is excellent. The tech support by phone is great. This program is a wonderful example of the power of the Apple II.

David Greenberg
Philadelphia, Penn.

AppleWorks MouseText

Have you ever wanted to include MouseText in an AppleWorks document you are working on? Run the following program to create an AppleWorks word processing file called *MOUSE-TEXT.DEMO*. Load this file into AppleWorks and you will see it contains all the MouseText characters. You can open-apple-<C>opy them to any other file. They will be displayed on the screen and will even stay there after you save and later reload the file. The only disadvantage is that they will not print out as MouseText on a printer; they will print as their ASCII upper case equivalent.

```

100 D$ = CHR$(4) : A$ = "MOUSETEXT.DEMO"
101 FOR A = 8192 TO 8704 : POKE A,0 : NEXT
102 FOR A = 8197 TO 8275 : POKE A,61 : NEXT
103 FOR A = 8202 TO 8272 STEP 5: POKE A,124: NEXT
104 POKE 8192,8 : POKE 8193,8
105 POKE 8196,79 : POKE 8278,2
106 POKE 8279,8 : POKE 8280,2 : POKE 8283,10
107 FOR A = 8493 TO 8560 : READ B : POKE A,B : NEXT
108 PRINT D$;"create";A$;"",tawp"
109 PRINT D$;"bsave";A$;"",tawp,A$2000,L$200"
110 DATA 208,0,208,61,0,0,187, 192,32, 193
111 DATA 32,194,32,195,32,196,32,197,32,198
112 DATA 199,32,200,32,201,32,202,32,203,32
113 DATA 204,32,205,32,206,32,207,32,208,32
114 DATA 209,32,210,32,211,32,212,32,213,32
115 DATA 214,32,215,32,216,217,32,218,32,219
116 DATA 32,220,32,221,32,222,255,255
117 END
    
```

Tom Hoover
 Lorena, Texas

We've included *MOUSETEXT.DEMO* the resulting file, instead of this program, on this month's disk.

LaserWriter printing service

In response to the letter from Charles Cook in your December issue regarding the availability of LaserWriter printing for *Publish It!* (page 4.87): Yes, there is such a service! I offer LaserWriter print output for a wide variety of Apple II products, including AppleWorks, *Publish It!*, *Springboard Publisher* and most Apple IIgs applications.

My modem line is 612-422-7823 and is in operation at 2400 baud 24 hours a day to receive files.

In addition, I also offer a graphic conversion service and will convert files among the major Apple II formats (*Print Shop*, *Newsroom*, single, double, and 320 or 640 super high resolution), Macintosh files (any format) as well as line art to be scanned can also be converted to the desired Apple II format. For more information and pricing structure, dial the modem line listed above or call my voice line at 612-421-8393 (5 to 10 PM, CST).

James Hirsch
 12310 Jonquil St NW
 Coon Rapids, Minn. 55433

Up to the second, 300 baud

Here's what I got when I called the National Bureau of Standards as you suggested last month ("Up to the second," page 5.14):

```

Time: UTC (NBS)
EH:MM:SS msADV OTM
02:12:40 045.0*
02:12:41 045.0*
02:12:42 045.0*
02:12:43 045.0*
02:12:44 045.0*
02:12:45 045.0*
    
```

It's not the same display you got and there's no "?" for HELP. Press any key and it disconnects. I was anxious to see the msADV column adjust and the On-Time Marker become "#". You said this would occur if I echoed all characters back. Does that mean something other than using full duplex?

Dan Muehsam
 Swedesboro, N.J.

That's the display you get when you call in at 300 baud. The display I published last month is the 1200 baud display.

None of the communications packages we're familiar with have the ability to echo received characters (this is a function usually associated with "hosts" rather than with "terminals"). It wouldn't be that hard for a small assembly language program to do it, however, if we could just get someone out there to write it.

A2-Central Developer Conference

See the inside pages of this month's catalog for complete details. In summary, you can spend July 21 and 22 in Kansas City listening to and asking questions of Apple II experts like Bill Mensch, Roger Wagner, Barney Stone, Mike Westerfield, Tom Leonard, Alan Bird, Mark Simonsen, Bill Basham, Gary Little and the A2-Central gang. If you've lost your catalog, call or write and we'll send you another one.

8-bit Finder

You guys, as well as most others, always overlook the Apple Desktop II (the third generation version of MouseDesk from VersionSoft) when discussing hard disk or 3.5 program selectors. While I haven't had the opportunity to try out *ProSEL*, I find *Desktop II* to be fast, easy to work with, and very flexible. It has several features that can be extremely valuable as well. When setting up path names in the program selector, you can specify auto load of a program on boot or auto load of a program on first use if you have an Apple memory expansion card. In addition, *Desktop II* loads itself into an Apple memory expansion card if it finds one on boot, making program swaps lightning fast. The program also has full blown icon-based disk and file maintenance utilities. To top it all off, it's free, having been included on the System Disk v1.0 that came with the Apple IIgs. It is of equal value to users of machines other than the Apple IIgs because it's a ProDOS 8 program. I think your readers should be aware of its existence.

Vern L. Mastel
 Mandan, N.D.

If you want an icon-based interface, you're right. We've always felt the program's size, at more than 150K, was excessive for a program selector, however.

IIgs HGR problems

I have an Apple IIgs and I think I have encountered a bug in Applesoft. Occasionally I type HGR (or HGR2) and the computer displays, but does not clear, the respective high-res pages. Mysteriously, an examination of the pertinent locations in memory reveal all zeros, indicating the screen should be blank! The only way to remedy the situation is to press Reset, after which subsequent HGRs result in clearing the screen as they should. Why is this happening?

Adam Levine
 Brooklyn, N.Y.

The failure, which occurs only on the IIgs, is a result of a previous program leaving some hardware switches in an improper state, not a bug in Applesoft. AppleWorks 2.0 is one of the programs that does this.

What the IIgs displays on its screen is actually controlled by the contents of memory banks \$E0 and \$E1. Applesoft and most other ProDOS 8 programs don't know anything about

these banks, so, normally, what is written into banks \$00 and \$01 is shadowed into banks \$E0 and \$E1. "Shadowing" means the Apple IIgs hardware watches what is written into the screen display areas of banks \$00 and \$01 and automatically writes it into banks \$E0 and \$E1 as well. Some programs turn this feature off, write screen display information directly to banks \$E0 and \$E1, and use banks \$00 and \$01 for other purposes. Such programs should always turn shadowing back on when finished, but some forget to do this.

Shadowing is controlled by the Shadow Register on the IIgs, which is at \$C035. Routines that fix the Shadow Register should always use a read-modify-write sequence. For more information, see "AppleWorks 2.0gs bugs", in our December 1987 issue, page 3.86.

More Micol musings

I agree with quite a few of the concerns Elliot Lifson expressed last month about *Micol Advanced Basic* (pages 5.15-5.16). Yes, the extra steps you have to go through before manipulating array strings is a hassle, remembering to put the X or Y after NEXT is a pain, and trying to remember whether the DATA or DIM statements come first in the program is troublesome.

On the other hand, being able to use local variables in your functions and procedures greatly outweighs any inconveniences that might be suffered from some of the other idiosyncrasies.

I'm very fluent in about 12 different dialects of Basic on 4-5 different computers. I know that when I get hold of a new one it's going to take some time to "re-learn" how to do different things. People who expect to pick up a new version of even such a simple language as Basic and know it inside and out without a certain amount of experimentation are fooling themselves.

Micol Advanced Basic, like all other IIgs Basics, is not 100 per cent compatible with Applesoft. I like Applesoft, too, but I'm not so foolish as to think the Applesoft way of doing something is the best way.

Micol Advanced Basic is not the perfect Basic for the Apple IIgs. But I've tried them all and it comes the closest so far. Best of all, you can send email to MICOL.SYSTEM on GENIE and let them know what you like or don't like. They seem to be very receptive to suggestions.

Oh, and one more thing. You mention in your reply to the above-mentioned letter that *Micol Advanced Basic* was "for non-professional programmers." Um...well...that seems to be a bit misleading. I know of at least one professional programmer who's using it for commercial purposes <ahem>.

Jay Jennings
 Wasilla, Alaska

International situation critical

My problem (and that of thousands of other Apple II users in my country) is that Apple Netherlands is killing the Apple II.

- the IIgs upgrade for the Apple IIe isn't available here.
- Apple Netherlands hasn't run a single ad for the Apple IIgs since its introduction.
- Apple Netherlands discourages dealers from selling the IIgs. In all those beautiful Apple Centres you don't see any IIgs machines. There is only one serious Apple II dealer left, Compudress in Kamerik.

- The people here at Apple practically deny the existence of the Apple IIgs in a highly-tighty manner—they seem to think there is only one type of Apple user, the Mac user.
- When you write Apple Netherlands with an Apple II question, you can expect to get no answer.

I do understand that Apple isn't a computer firm just for hobbyists and small businesses, but I do also believe there is a lot of potential in this reservoir of Apple II users. Why is Apple throwing away this potential knowledge, enthusiasm, and revenue? Apple Netherlands has antagonized the Apple II user. We shiver, we are left out in the cold.

Jack van Soest
Vlaardingen, The Netherlands

Here is a letter from a lonesome Apple II user. Why am I lonesome? Because France has been invaded by the Macintosh and the Apple II is no longer sold. What's wrong with Apple in Europe? Just to show how much they disgust me, I have some brochures about the Commodore Amiga on my desk.

Nicolas Renon
Villeurbanne, France

I'm the happy owner of an Apple IIc, but, unfortunately, I'm living in France, where the only Apple II to be sold since 1987 is the IIgs. That means it's now impossible to buy a UniDisk 3.5. Is it possible for me to buy such a drive mail order from the U.S.A.?

Denis Gasteuil
Bordeaux, France

How can I obtain the GS/OS System Disk? Here in France the Apple II has been completely abandoned in favor of the Mac. I was told by the dealer who sold me the IIgs that Apple France might get GS/OS in six months.

Erwin Solomon
Ville D'Avray, France

Please help. It's nearly impossible to obtain IIgs system disk updates here in Germany.

Udo Huth
Wittmar, West Germany

Everybody here in Germany is switching to IBM clones. With plenty of software available at low prices and good support for the hardware they now have taken the place of the Apple II family. There are still a few die hards like me, coming the long way of KIM, AIM 65, and Apple II. But I need an economical, up-to-date system at home, and soon there will be no other alternative but IBM.

Peter Zechner
Munich, West Germany

Some months ago a rumor appeared in some Italian magazines that John Sculley had said Apple would no longer support the Apple II in the European market. If this is true, what can we do to change the situation? In Italy the situation for Apple II users is so critical that anything is welcome.

Maurizio Stroppiana Pinoso
Genova, Italy

My Apple dealer can't get me a SCSI card ROM upgrade. Someone told me that in the U.S. this upgrade is free. What can I do?

Alvise Majer
Gorgonzola, Italy

Apple Italy is about to sink the IIgs—poor support, lack of accessories and devices. Surviving is very hard. We love the IIgs, but Apple Italy doesn't love us. Neither the Apple IIe nor the Apple IIc are available here. Will the IIgs come next? We won't upgrade to the Mac, we'll die with the Apple II.

Walter Ferlazzo
Genova, Italy

I contacted Apple International in Hong Kong about a problem with my IIc and they did not have the information. The last time I wrote Apple Cupertino for information, I didn't get a reply for seven months.

Stuart C. Keen, Chief Engineer
Office of Security, American Embassy
Beijing, China

A neighbor of mine here in Subic Bay has just purchased an Apple IIgs from our (if you'll pardon the expression) 'computer store', the Navy Exchange. Now, you must understand, the Filipino women who work as order takers in the Exchange don't understand computers at all. The IIgs came with System Disk v1.1. I don't have a IIgs and haven't been paying much attention, but in re-reading back issues I've learned that System Disk 4.0 came out last year and that my neighbor needs a ROM upgrade to use it. Any idea how we can get system disk or ROM updates when our nearest 'authorized Apple dealer' is seven thousand miles away?

Kenneth C. Mitchell
Subic Bay, Philippines

Why would anyone complain about Apple Canada? Just because as of December we still have ROM 1.0, no GS/OS, and little support? We might even get AppleLink eventually! Apple's

offices consist of people who are never there—just answering machines. The 800 numbers are unlisted, warranties are hidden, and not even ProDOS 16 is available.

Last time I talked to them, Claris Canada didn't have AppleWorks 2.1. They were still selling v2.0 and telling customers to send to the U.S. for a free update.

I have never been so sorry in my life. Apple Canada is a farce and a shambles. I don't even want to know what their definition of sales or service is. I love Apple's computers, but I no longer have faith in the company behind them.

Ephraim Dickstein
Halifax, NS

More than ten per cent of our subscribers are outside the U.S. In the last few months we've been inundated with reports of poor service and support from Apple II users all over the world. Unlike the situation in the U.S., the international problems seem centered in Apple's own offices, rather than at computer stores. This is not to insinuate that Apple's support for the II here in the U.S. is generous.

Apple seems to think that its lack of support for the II is adequately addressed by suggesting that Apple II owners switch to the incompatible Macintosh. We feel that a company that is willing to abandon hundreds of thousands of Apple II users around the world will also be willing to abandon hundreds of thousands of Macintosh users a few years from now. To buy another computer from the same company under these circumstances is foolish. Since Apple wants us to change operating systems anyhow, we'd be much smarter to change to a less proprietary operating system, which has a better chance of being supported in the future, now.

One step Apple could take would be to allow U.S. dealers to engage in mail-order sales to countries in which it has chosen not to support the II.

One step we have taken has been to license GS/OS for our monthly disk—thus we can provide system disk updates to our disk subscribers as soon as they are available here in the U.S.

It might help to write to the head of the appropriate Apple division, although one would assume they should already be aware of the situation. They are Del Yocam for Apple Pacific and Michael H. Spindler for Apple Europe (Apple Computer, 20525 Mariani Ave, Cupertino, CA 95014).

I'm convinced no one in the upper reaches of Apple's hierarchy has the time to take any message about the Apple II seriously, including those published in Apple II publications. But it might help to send letters to Macintosh and general interest computer publications pointing out that Mac users are likely to get the same treatment in a few years that Apple II users are getting now.

Finally, I encourage those of you who live in the U.S. to discuss the situation with your stockbrokers. Apple is leaving Apple II revenue on the table all over the world, including in the U.S., through slovenly marketing of the Apple II family. The ears we need to reach with that message are Wall Street analysts. Apple is supposed to be a marketing-oriented company that listens to its customers, but every day we get letters showing that it doesn't. Maybe it will listen to Wall Street.

A2-Central™

© Copyright 1989 by
A2-Central

Most rights reserved. All programs published in A2-Central are public domain and may be copied and distributed without charge. Apple user groups and significant others may obtain permission to reprint articles from time to time by specific written request.

Written, edited, and published by:

Tom Weishaar

with help from:

Dennis Doms	Sally Dwyer	Dianne Plumberg
Tom Vanderpool	Steve Kelly	Joyce Hammond
Dean Esmay	Jean Weishaar	

A2-Central, titled Open-Apple through January, 1989—has been published monthly since January 1985. World-wide copies (in U.S. dollars; airmail delivery included at no additional charge): \$28 for 1 year; \$54 for 2 years; \$78 for 3 years. All back issues are currently available for \$2 each; bound, indexed editions of our first four volumes are \$14.95 each. Volumes end with the January issue; an index for the prior volume is included with the February issue.

The full text of each issue of A2-Central is available on 3.5 disks, along with a selection of the best new public domain and shareware files and programs, for \$84 a year (newsletter and disk combined). Single disks are \$10. Please send all correspondence to:

A2-Central
P.O. Box 11250
Overland Park, Kansas 66207 U.S.A.

A2-Central is sold in an unprotected format for your convenience. You are encouraged to make back-up archival copies or easy-to-read enlarged copies for your own use without charge. You may also copy A2-Central for distribution to others. The distribution fee is 15 cents per page per copy distributed.

WARRANTY AND LIMITATION OF LIABILITY. I warrant that most of the information in A2-Central is useful and correct, although errors and mistakes are included from time to time, usually unintentionally. Unsatisfied subscribers may cancel their subscription at any time and receive a full refund of their last subscription payment. The unfulfilled portion of any paid subscription will be refunded even to satisfied subscribers upon request. MY LIABILITY FOR ERRORS AND OMISSIONS IS LIMITED TO THIS PUBLICATION'S PURCHASE PRICE. In no case shall I or my contributors be liable for any incidental or consequential damages, nor for ANY damages in excess of the fees paid by a subscriber.

ISSN 0885-4017
Printed in the U.S.A.

GENIE mail: A2-CENTRAL
913-469-6502