A2-Central™

formerly Open-Apple

July 1989
Vol. 5, No. 6

ISSN 0885-4017
newstand price: $2.50
photocopy charge per page: $0.15

A journal and exchange of Apple II discoveries

AppleWorks 3.0 a blockbuster

Imagine what would happen if Claris asked Beagle Bros to make all the enhancements to AppleWorks that users had ever dreamed of. Now stop imagining. It's reality.

Claris announced AppleWorks 3.0 at the National Educational Computing Conference in Boston on June 20. Alan Bird, originator of Beagle Bros' *TimeOut* utilities and author of *TimeOut QuickSpell*, Randy Brandt, author of *TimeOut UltraMacros*, and Rob Renstrom, author of *TimeOut Graph*, have been at work on the AppleWorks 3.0 project since last summer.

Imagine a 16,000-line word processor with a new, more-advanced version of *QuickSpell* built-in; with real tabs and multiple tab rulers; with multi-line headers and footers; with more control-key commands (such as control-C for center); with the ability to print a specified range of pages; with the ability to print date and time codes (in a selection of formats); with right-justification; and with six new printer codes you can use to take advantage of whatever special features your printer supports.

Imagine a 16,000-record data base with horizontal scrolling in multiple-record layout; with multiple category sorts; with single category finds; with one-key entry of the current date and time; with 20 report formats per file that include new abilities such as multiple labels across a page and the ability to create multiple-record screen formats from report formats and vice-versa.

Imagine a 9,999 row by 127 column spreadsheet that is smart enough to determine whether a cell being referenced is a label or a value and act accordingly. Put (A2) in a cell and it will display "1" if that's what's in cell A2 or it will display "Wow!" if that's what's in cell A2. Imagine being able to turn numeric scores into letter grades and numeric cross-checks into verbal error messages. Imagine five new arithmetic functions like @LOG and @MOD, nine new trigonometric functions like @SIN and @ACOS, six new financial functions like @PV and @IRR, and six new logical functions like @TRUE and @ISERROR. Imagine @IF(@ISBLANK(A2),"You must fill in cell A2",""), to say nothing of the ability to copy and move *blocks* of cells to and from the clipboard.

Imagine a clipboard with no size limit other than the amount of memory you have. Imagine a clipboard with the ability to automatically format the data it contains for any AppleWorks module. Yes, you can move data from any module to any other module–from the spreadsheet to the word processor, for example–without first printing it to the clipboard or a file on disk.

Imagine being able to select subdirectories, as well as files, using point-and-shoot rather than try-and-remember–even when accessing ASCII text files. Imagine being able to switch drives and to dive into and back out of subdirectories by pressing keys while the current file listing is on the screen.

Imagine up to three custom printers, not just one, and built-in support for Brother, Epson, Juki, Okidata, and Panasonic printers, to say nothing of the ImageWriter II and the ImageWriter LQ.

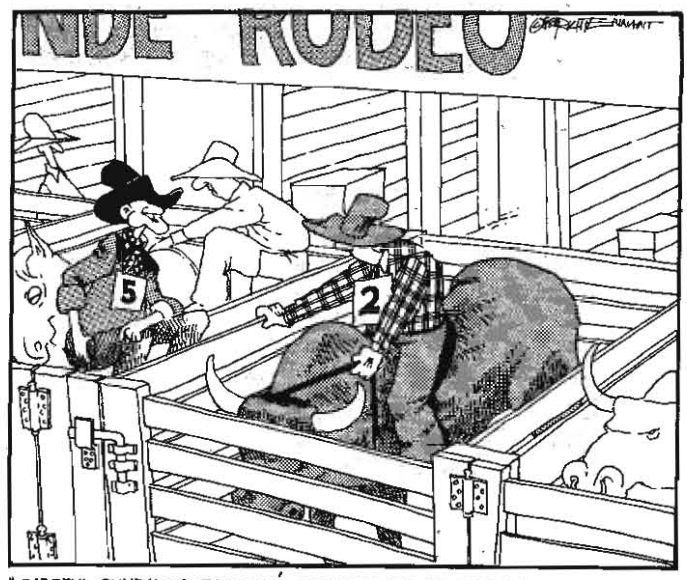
Imagine automatic support of aux-slot type memory cards (in addition to current support for standard-slot and IIgs-memory-slot cards). Imagine no retail price increase. Imagine an upgrade price of $79 and a special Claris program whereby ***A2-Central*** subscribers who don't have an older copy of AppleWorks to upgrade from can get AppleWorks 3.0 directly from Claris for $99 (see this month's catalog).

AppleWorks 3.0 still works on 128K machines, however, as before, it works better on machines with additional memory. Desktop size on 128K machines is now about 40K instead of 56K.

AppleWorks 3.0 comes on two double-sided 5.25 disks or one 3.5 disk. Side 1 of the 5.25 startup disk includes PRODOS, APLWORKS.SYSTEM, four files for various kinds of memory expansion cards, a configuration file, a printer information file, and the SANE mathematics file. Side 2 of the 5.25 disk set includes the word processor and data base segments. Side 3 has the spreadsheet. Side 4 has the main and custom dictionaries for the spelling checker. Sides 1, 2, and 3 also all have a file called SEG.AW, which holds the desktop and main menu segments.

Here's a complete list of the AppleWorks 3.0 program files:

filename	size (blocks)	description
PRODOS	32	disk operating system
APLWORKS.SYSTEM	26	main program file
SEG.00	9	memory segment for 128K machines
SEG.AM	9	" " for aux-slot memory
SEG.XM	9	" " for std-slot memory
SEG.RM	9	" " for IIgs memory
SEG.EL	12	SANE elementary math functions
SEG.ER	6	Configuration file
SEG.PR	9	Printer information
SEG.AW	97	Desktop Interface and Main Menu
SEG.WP	87	word processor
SEG.DB	86	data base
SEG.SS	76	spreadsheet
MAIN.DICTIONARY	402	(5.25 version is 254 blocks)
CUST.DICTIONARY	1+	(expands as you add words)



"CAREFUL SUNDANCE, THIS ONE'S BEEN FORCED TO USE A MACHINE WITHOUT A RAM DISK ALL WEEK AND HE'S ITCHING FOR A FIGHT."

It's possible to create minimum-configuration disks that don't include all of these files. You need only the one memory segment file (SEG.00, SEG.AM, SEG.XM, or SEG.RM) that will be used by your machine. If you don't use the spreadsheet, you don't need SEG.SS; the data base, SEG.DB; the word processor, SEG.WP nor the dictionaries. And after you've configured your printers you no longer need SEG.PR. (Under AppleWorks 3.0, SEG.PR contains the list of supported printers, as displayed when you add a printer, and all the printer control codes and sequences for those printers. Unlike earlier versions, it doesn't contain any variable configuration information about printers–that's been moved to the new SEG.ER, along with other configuration information. Third parties can now construct special SEG.PR files holding information for additional printers.)

Desktop enhancements

AppleWorks 3.0 now has a single standard interface for dealing with files on disks, which has the following features. Whenever a list of disk files is displayed on the screen (except when selecting custom dictionaries, which must be at the same directory path as the AppleWorks program files), the following keys can be used to navigate through directories ("OA" means hold down the open-apple key):

TAB	AppleWorks reads and displays the root directory of the next drive.
OA-.	(note: unshifted OA->) adds highlighted subdirectory to pathname and displays the subdirectory's contents (same as Return).
OA-,	(note: unshifted OA-<) deletes last subdirectory from pathname and displays prior subdirectory. Beeps in root directory.
OA-RETURN	Selects current pathname and returns to previous menu.
ESCAPE	Returns to previous menu without changing pathname.
RETURN	With files selected: loads selected files. With no files selected: File highlighted: loads the file Subdirectory highlighted: same as OA-.
Right/Left	Select and deselect files as before; beep if on subdirectory.
Up/Down	Move cursor up and down through directory listing, as before.
OA-Up/Down	Page up, page down.
OA-1/OA-9	Top of list, bottom of list (OA-2 through OA-8 do nothing).

AppleWorks 3.0 allows you to navigate through subdirectores by pressing keys.

There are seven ways to display lists of files on disk from within AppleWorks 3.0. No matter what order the files are actually in on the disk, in the AppleWorks file listing they are grouped in the following order: word processor files, data base files, spreadsheet files, text files (well, text files *are* listed first when you "Make a new file" from a text file or a DIF file), other files, and subdirectories. Within groups the files are listed alphabetically.

All six types of files are displayed when you "List all files on the current disk drive" or "Delete files from disk" (Other Activities menu), when you "Make a new file from a DIF file" (data base and spreadsheet), and when you "Make a new file from a text (ASCII) file" (word processor and data base–as with earlier versions of AppleWorks, you can actually use this command to load *any* type of file into AppleWorks, not just text files, but you can only load one file at a time).

When you "Get files from the current disk", only AppleWorks files and subdirectories are displayed. When you select a custom dictionary, only text files are displayed (the dictionaries have to be at the same pathname as the rest of the AppleWorks program). When you "Change Current Disk" using "ProDOS directory", only subdirectories are displayed (if you insist, you can also type in pathnames the old way by selecting "ProDOS directory" with OA-Return). AppleWorks 3.0 no longer supports "Make a new file from a Quick File file" (data base) or "Make a new file from a VisiCalc file" (spreadsheet).

With the new commands, there's almost no reason to ever use choice 2 from the "Add files" menu ("Get files from a different disk"). Always select "The current disk" and if it's not the one you really want, use Tab to quickly display the one you do.

Enhanced text file importing/exporting. Now that the word processor supports true tabs, more options are available for importing and exporting files. When you import a text file to the word processor, any embedded tabs (Control-Is) will be converted to AppleWorks tabs. When you import a text file to the data base, AppleWorks 3.0 asks:

```
Does the text (ASCII) file have:
1. Tabs between categories, Returns between records
2. Return after each category
```

If you select 2, AppleWorks will ask how many categories there are per record, just as it does now. If you select 1, however, AppleWorks will be able to determine that itself and will skip the question.

When printing to a text file from the word processor, the following menu appears:

```
Should the text (ASCII) file have:
1. Standard text format with Tabs
2. Spaces substituted for tab stops
3. Returns after each line
```

Option 3 is the equivalent of the way current versions of AppleWorks print text files. Option 3 gives you a Return at the end of each line and embedded spaces rather than tabs. It's ideal for creating text files you intend to upload to an online service.

Options 1 and 2, on the other hand, don't add Returns to the file. Returns appear only where they are in the original file, which is usually only at the end of paragraphs (this is how early versions of AppleWorks printed text files). Option 1 puts control-Is in the file; option 2 puts the correct number of spaces. Which is better depends on whether the software you're sending the text file to recognizes control-Is or not.

When printing to a text file from the data base, the following menu appears:

```
Should the text (ASCII) file have:
1. Tabs between categories, Returns between records
2. Return after each category
```

And when printing to a text file from the spreadsheet, you'll see:

```
Should the text (ASCII) file have:
1. Tabs between columns, Returns between rows
2. Return after each cell
```

(The AppleWorks 3.0 specifications also say that new subdirectories and exported text and DIF files will be created in the current data disk location unless the user enters a complete pathname. In previous versions, the user was forced to enter a complete pathname–a partial pathname wouldn't do. Based on the pre-release version of 3.0 Claris gave me to look at, this new feature wasn't implemented.)

Smart save command. When an AppleWorks file (but not a text or DIF file) is loaded from disk, AppleWorks will remember the pathname the file was in. OA-S still saves a file to the current disk location, as before, however, OA-Control-S now sets the current disk location to the file's original pathname and then saves the file.

When you use the main menu's Save Files option, AppleWorks 3.0 gives you a third option in addition to "Save the file on the current disk" and "First change to a different disk or directory". That option is "Save the file to its original directory." If the original directory can't be found because the user has switched disks, the "Getting errors trying to write at..." filecard will appear. It allows the user to insert the correct disk and then "Try again" or to "Try a different location."

Clipboard enhancements. Three enhancements were made to the AppleWorks 3.0 clipboard. First, the only limit on the size of the clipboard is the amount of available memory. Second, you can now copy and move spreadsheet columns and blocks to and from the clipboard (formerly you could only deal with entire rows). When copying or moving spreadsheet data back into a spreadsheet, the data on the clipboard overwrites pre-existing data.

The third clipboard enhancement allows you to use the clipboard to transfer data directly between the word processor, data base, and spreadsheet. When transferring data between the spreadsheet and the data base, spreadsheet rows are equivalent to data base records and columns are equivalent to categories.

When copying or moving data base or word processor data from the clipboard *to the spreadsheet*, new rows are inserted for each

record or line. The data will start in column A. If there are no Tabs in word processor text, each whole line will be placed in a single cell in column A. If there are Tabs, additional cells will be used for each Tab. Numbers in data base categories or between word processor Tabs will be converted to numeric values.

When copying or moving spreadsheet or word processor data from the clipboard *to the data base*, new records are inserted for each row or line. The data starts in the first category. If there are more columns/Tabs than categories, the excess will be ignored; if less, the remaining categories will be blank.

When copying or moving spreadsheet or data base data from the clipboard *to the word processor*, Tabs are automatically inserted into the text to separate columns/categories. Returns at the end of each line separate rows/records.

Other desktop changes. Some other changes in AppleWorks 3.0 include the following: the program no longer asks the user to press the space bar after display of the title/copyright screen during startup. If the system has a clock, AppleWorks 3.0 will read the system date during startup and not ask the user to enter the date.

OA-Delete now deletes the character the cursor is on (or gobbles characters to the right of the cursor) in all parts of the program. Delete alone works as before. OA-right-arrow and OA-left-arrow move the cursor to the first character of the next word, right or left, in all parts of the program. When Control-Reset is pressed, AppleWorks 3.0 tries to return to the main menu rather than dropping into the Monitor.

"Thermometers" similar to the one displayed while GS/OS is booting are used in AppleWorks 3.0 to indicate how quickly time-intensive tasks such as preloading files, sorting, printing, and searching dictionaries are moving along. A warning message appears whenever a duplicate filename is added to the desktop (the message appears in current versions only when adding a new file from disk; under 3.0 it will also appear when making a new file from scratch or when renaming a file.) Users can now print up to 255 copies of a file. And the OA-H(ardcopy) command now issues a form feed to advance the paper after printing what's on the screen.

Word Processor Enhancements

Spelling checker. The new built-in spelling checker is accessed by pressing OA-V(erify spelling). This brings up the following one-line menu at the bottom of the screen:

```
Verify spelling? All Word Block Dictionary Options
```

"All" checks the spelling of the entire document. "Word" checks the spelling of the word the cursor is on, or the word to the left of the cursor if it's between words. "Block" allows the user to select a block of text and then checks the spelling of all the words in that block.

"Dictionary" brings up a second menu that looks like this:

```
Custom Dictionary? Get existing Create new
```

Each time you start up AppleWorks 3.0, the configuration file is used to determine the name of the default custom dictionary. The first option allows you to switch from the default custom dictionary to another one. The second option allows you to create a new, empty, custom dictionary and start using it. Dictionaries selected using the "Dictionary" option remain in use through the current session. The next time the user runs AppleWorks, however, the custom dictionary listed in the configuration file, which we'll look at later, is reselected.

The built-in spelling checker is even better than *Quick-spell.*

In addition, each time you start up AppleWorks 3.0, the configuration file is used to determine whether you want to check spelling "in context" or "from a list", to determine whether you want a summary of the spelling changes you've made, and to determine whether you want your summary displayed on the screen or placed on the clipboard. (You can also choose to get the summary only without stopping to correct anything; this option was added so that teachers could quickly check homework-on-disk.)

The "Options" selection in the "Verify spelling?" menu allows you to select the alternate checking method or alternate summary options for the current spell-check. (Your method/summary selections are good one-time only and only if you select "All" or "Block" immediately after making the selection. Escape back to your document and you'll lose the selection. "Word" verifications are always in context and never include a summary.)

When you verify spelling *in context*, AppleWorks proceeds through the document stopping at each unknown word in the order in which they occur. The document will be displayed on the screen with the unknown word highlighted in approximately the center of the screen. At the bottom of the screen you'll see the following menu:

```
Unknown word? Replace Add to dictionary Ignore Skip Get suggestions
```

"Replace" allows you to type in a correction. First you'll have to specify whether the correction applies only to that instance or to all occurrences of the unknown word. "Add to dictionary" adds the unknown word to the current custom dictionary. "Ignore" skips all occurrences of the unknown word. "Skip" skips just that one instance of the unknown word. "Get suggestions" brings up a list of possible correct spellings for the unknown word. You can scroll through this list and replace the unknown word with the highlighted word by pressing Return.

When you verify spelling *from a list* instead of in context, you are presented an alphabetical list of all the unknown words in the document. You can use the four arrow keys to scroll through and select some or all of the words in this list. Selected words can be added to the dictionary (OA-A), ignored (OA-I), or replaced by typing in a correction (OA-R). You can also choose to correct selected words in context (OA-C), which is how you get the program to give you suggested spellings for the unknown words in the list.

No matter which method you use, after you've dealt with all the unknown words, you'll be presented with an in-context display of all double double word pairs and you'll be given an opportunity to remove one of the duplicates. When verifying from a list, you can delete double words anytime by pressing OA-D.

Looking only at spell-checking methods, AppleWorks 3.0's built-in spelling checker is almost a twin of *TimeOut Quickspell*. The spelling summary, on the other hand, is a new feature that *TimeOut QuickSpell* doesn't have.

The spelling summary is displayed on the screen or placed on the clipboard after you've finished checking a document. Here's a summary for the file "Tax Estimtr.Doc," which is a sample file from the National AppleWorks User Group that's included on the AppleWorks 3.0 disk:

```
Total words:      648
Unknown words:    5
Corrections made: 1

Unknown word          Correction          Count
============          ==========          =====
2a                                          2
2b                                          2
documention           documentation         1
estimator'89                                1
vincenti              <ADDED TO DICT>       1
```

"Total words" is a count of all the words in the document. Words that appear more than once are counted each time they appear. "Unknown words" shows how many unique, unknown words are in the document (this time words that appear more than once are counted just once); a list of these words and what action was taken with each follows. "Corrections made" shows how many of the unknown words were corrected. The list that follows shows each unknown word; whether it was corrected, added to the dictionary, or ignored/skipped; and a count of how many times it occurred in the document.

As mentioned earlier, you can configure AppleWorks to skip the summary completely, to have it appear on the screen, or to have it placed on the clipboard. And you can also configure it to immediately display the summary on the screen without stopping to allow you to make corrections. As shipped, AppleWorks 3.0 is configured for no summary, in context checking, and a supplied custom dictionary called CUST.DICTIONARY.

OA-Options menu; control-key commands. While the addition of a spelling checker is significant in itself,a number of other enhancements were also made to the word processor. An extra line was added

to the already crowded OA-O(ptions) menu, which allowed the addition of four new options. The added options are RJ (right justification), PD (print date), PT (print time), and SC (special code).

The addition of right-justification gives AppleWorks a full set of justification options.

The addition of right justification finally gives AppleWorks a complete set of justification options. The others are UJ (unjustified or left justification), JU (full justification), and CN (centered). In addition, all four of these options can be inserted in a document with control-key commands: control-N (normal or left justification), control-F (full justification), control-R (right justification), and control-C (center).

Two additional control-key commands have been added to AppleWorks 3.0 (beyond control-B for boldface on/off and control-L for underline on/off, which are in current versions). They are control-P (page break), which has the same effect as the NP (new page) printer option and control-A (add-on), which inserts a formatting token that can be used by add-on software such as the *TimeOut* series. The following chart shows the complete list of AppleWorks 3.0 word processor control-key commands:

```
control-B  boldface on/off
control-L  underline on/off

control-N  normal (left) justification
control-F  full justification
control-R  right justification
control-C  center

control-P  page break
control-A  add-on token
```

The print date and print time options insert the current date or time into a document as it is being printed. They work in much the same way as the PP (print page number) printer option. Like PP, they are displayed as carets on the screen. The format of dates and times is configurable within AppleWorks 3.0, as we'll see in detail later; these options print the user-selected format.

The SC (special code) option allows you to insert one of six user-defined printer codes into a document. These are also displayed as carets on the screen. As in current versions of AppleWorks, a description of what a caret represents appears in the center of the bottom fine of the screen when the cursor is on a caret. Even this description is user-definable for the six special codes. The codes can be used to turn on features some printers may have that AppleWorks doesn't otherwise support, such as alternate character sets (italics or foreign languages). Or they can be used for accents (for example, define a special code as "control-H ^"--control-H is the backspace command for most printers--and use it to put a caret over the preceding letter.)

Real Tabs. The addition of real tabs to the word processor has allowed a number of enhancements to AppleWorks 3.0, as seen earlier in the discussion of text file import and export options. Moreover, "real tabs" mean that pressing the Tab key no longer inserts spaces into a document; it now inserts a "tab token". When you adjust tab settings, the position of tabbed text in your document changes to match the new tab settings. This allows you to move whole columns of text right or left simply and easily. It also means the Tab key can now reposition text rather than just reposition the cursor. (OA-Tab in the word processor works like Tab alone did in prevous versions. There is now no way to back-tab in the word processor.)

Different parts of a document are allowed to have different tab settings through the use of tab "rulers". When you press OA-T(abs), you get a new prompt that looks like this:

```
Tab ruler?  Modify current  Create new
```

The "Modify current" choice allows you to change tab stops on the current ruler as in current versions of AppleWorks. The "Create New" choice gives you a new ruler, which initially has the same stops as the current ruler, and then allows you to modify the stops as you like. A ruler is in effect from the paragraph following the ruler until either a new ruler is defined or the end of the document. The tab display at the top of the screen always shows the settings of the current ruler–as you move the cursor through a document the display will change when you pass rulers, which, if you use OA-Z(oom), look like this:

```
--------Tab Ruler
```

Like any other printer option token, tab ruler tokens can be copied, moved, or deleted. However, the first tab ruler in a document doesn't have a token. To copy the first ruler you have to make a duplicate of it by pressing OA-1 to go to the top of the document, OA-T(abs), "Create New", and Escape. This creates a copyable token of the first tab ruler.

Four kinds of tab stops are available: left, center, right, and decimal. A normal, left-justified tab appears in the tab display at the top of the screen as a "<". Think of the point of the less-than sign as representing the tab stop and the arms as representing where the characters will go. A centered tab appears as "^". A right-justified tab appears as ">". A decimal tab appears as "."; decimal tabs line up a column of numbers on the decimal point, as shown in the following table:

```
symbol   kind                effect

  <      left                  <--stop is here
  ^      center              text centers
  >      right      stop is here-->
  .      decimal               $99.95
```

When a new word processor file is created from scratch or a file from an older version of AppleWorks is added to the desktop, initial tab stops appear every five spaces. All are left justified.

Multi-Line Headers and Footers. The OA-O(ption) HE (page header) and FO (page footer) commands now insert two printer codes each into your document, a begin code and an end code. Any printer options or text entered between the codes will be used as the header or footer. Headers and footers can have any reasonable number of lines (more than half a page isn't reasonable). When copying, moving, or deleting headers they must be treated as a block, however, users can copy/move/delete parts of text and formatting that is between begin and end markers.

Other enhancements. The print menu now looks like this:

```
Print from?  Beginning  This page  Cursor  Page to page
```

If you select the new "Page to page" option, you'll next be asked to select a printer. After that you'll be asked for beginning and ending page numbers. You'll know what page numbers you want to print, of course, because you'll be smart enough to OA-K(alculate page numbers) and see before you try to print.

OA-Z(oom) has been changed slightly so that all printer options, including headers, footers, and most carets, are hidden in zoom-out mode. Zoom-in shows everything, as before.

Placing a Boldface Begin or Underline Begin at the beginning of a document and not matching it with a Boldface End or Underline End will boldface or underline the entire document.

When using the Copy and Move commands, line numbers are no longer removed from the bottom of the screen.

On machines with more than 256K of memory, the maximum number of lines in a word processor document is more than 16,000.

As mentioned earlier, OA-Delete removes the character the cursor is on. OA-right/left arrow moves the cursor to the first character of the next word to the right or left. In addition, the unshifted forms of OA-< and OA-> (actually OA-, and OA-.) move the cursor to the beginning and end of the current line.

Data Base Enhancements

The multiple-record layout in the AppleWorks data base has been greatly enhanced by adding horizontal scrolling, left-side titles, and selectable user-defined layouts.

The multiple-record layout now scrolls horizontally much like the spreadsheet. The unshifted forms of OA-< and OA-> move the cursor to the left and right edge of the document. OA-right/left-arrow move the cursor to the left and right edge of the screen. (The functioning of these key combinations in the spreadsheet and word processor is identical, with the exception that in the word processor OA-right/left-arrow moves the cursor one word at a time rather than one screen at a time.) As in the spreadsheet, you can force the left-most categories to stay on the screen during horizontal scrolling by setting

titles. The command for this is OA-T(itles). When you press OA-T, the cursor must be to the right of (rather than on top of) the categories you want stuck on the screen.

When you press OA-L(ayout) from a multiple-record layout, you get a new menu that looks like this:

```
1. Change the existing record layout
2. Get a layout from a report format
```

(If no report formats have been defined, this menu doesn't appear.) If you choose to get a layout from a report format, you're next presented a list of available formats. Select one, tell AppleWorks whether you want the cursor to go down or across when you press Return, and your multiple-record display changes to match that of the report format. This gives you an easy way to flip between different multiple-record "views" of your data.

Likewise, when you tell AppleWorks you want to create a new "tables" report format for printing, you'll see a new menu that asks:

```
Create a new "tables" format:
1. From scratch
2. From the current record layout
```

If you select 2, all the layout work you've done in multiple-record format will be transferred to your new report format. Further modifications are possible if you desire.

You can now have up to 20 report formats per file, instead of just 8 as before. If you are working with a labels-style format, the OA-O(ptions) command in the data base now includes an item called "columns" at the bottom of the "left and right margins" section. This allows you to print several labels across a page. You can specify any number between 1 and 24, although anything over 5 makes for a very narrow label.

The OA-F(ind) command now displays the following menu:

```
Find text? Anywhere In a specific category
```

"Anywhere" produces a search just like the one current versions of AppleWorks do. If you select "In a specific category", you'll be presented a list of categories in the file and allowed to select just one. This allows you to find your clients named "Smith" without finding your clients who live on "Blacksmith Road", or who live in "Smithsville", or who work for the "Smithsonian".

The OA-A(rrange) command now displays the following menu:

```
Arrange (sort) on? Category (nnnnnnnn) Several categories
```

In the position where I put "nnnnnnnn" in this example you'll actually see the name of the category the cursor was on when you pressed OA-A. Selecting that produces a sort you'll recognize from current versions of AppleWorks. If you select "Several categories" instead, you can select up to three and an arrangement order for each. The screen resembles the OA-R(ecord select) screen while you do this. As you would expect, the sorts are done in reverse order from the way you select them, so that the last sort is done on the first category you select. For example, if you select the "city" category and then the "last name" category from a mailing list data base, the final arrangement will be by city, with records sorted by name within the city groups.

Other enhancements. After pressing OA-L(ayout) in single-record format, you can now press OA-T(itles) to make category names appear in inverse.

If you enter a "@" in a date field or a time field and press Return, AppleWorks will replace it with the current date or time.

On machines with more than 256K of memory, the maximum number of records in a data base document is now more than 16,000.

Spreadsheet enhancements

The two primary enhancements to the spreadsheet in AppleWorks 3.0 are the addition of a large number of new functions and the ability to reference text in formulas.

There are 26 new spreadsheet functions. Here's a complete list:

Arithmetic functions

@MOD(NUMBER, DIVISOR)
Returns the remainder of: NUMBER divided by DIVISOR.

@PI
Returns 3.1415927

@EXP(NUMBER)
Returns e raised to the power of NUMBER

@LN(NUMBER)
Returns the natural logarithm of NUMBER

@LOG(NUMBER)
Returns the bsse 10 logarithm of NUMBER, called @LOG10 by some other spreadsheets.

Trigonometric functions

@DEG(NUMBER)
Converts NUMBER, which is an angle in radians, to degrees.

@RAD(NUMBER)
Converts NUMBER, which is an angle in degrees, to radians.

@COS(NUMBER)
Returns the cosine of NUMBER, which is an angle in radians.

@ACOS(NUMBER)
Returns the angle in radians whose cosine is NUMBER.

@SIN(NUMBER)
Returns the sine of NUMBER, which is an angle in radians.

@ASIN(NUMBER)
Returna the angle in radians whose sine is NUMBER.

@TAN(NUMBER)
Returns the tangent of NUMBER, which is an angle in radians.

@ATAN(NUMBER)
Returns the angle in radians whose tangent is NUMBER.

@ATAN2(X-NUMBER, Y-NUMBER)
Returns the angle in radians whose x and y coordinates are X-NUMBER and Y-NUMBER.

Logical functions

@TRUE
Returns the value TRUE (1)

@FALSE
Returns the value FALSE (0)

@NOT(NUMBER)
Returns FALSE if NUMBER is TRUE and vice-versa

@ISBLANK(REF)
Returns TRUE if cell (REF) is empty, else FALSE

@ISERROR(REF)
Returns TRUE if cell (REF) is "ERROR", else FALSE

@ISNA(REF)
Returns TRUE if cell (REF) is "NA", else FALSE

Financial functions

@IRR(RANGE, GUESS)
Returns the internal rate of return of the cash flow in RANGE. @IRR uses iteration. It repeats the calculation until the values converge. If they don't converge after 20 iterations the function returns an error. In this case, try a different value for the GUESS parameter.

@FV(RATE, TERM, PMT[, PV, TYPE])
Returns the Future Value of a series of equal payments when the interest rate per period, number of periods, and payment are known. The optional PV parameter in brackets allows you to give a beginning value to the cash flow (defaults to zero if not specified). The optional TYPE parameter specifies whether payments occur at the beginning (1) or end (0) of the period (defaults to zero).

@PV(RATE, TERM, PMT[, FV, TYPE])

@TERM(RATE, PMT, PV[, FV, TYPE])

@PMT(RATE, TERM, PV[, FV, TYPE])
These functions return Present Value, Term, and Payment of a series of equal payments. You must know the interest rate per period and two of the values to calculate the third. The optional FV parameter allows you to give an ending value to the cash flow. See @FV for an explanation of the TYPE parameter.

@RATE(TERM, PV, FV)
Calculates the interest rate of a lump sum transaction when the Term, Present Value, and Future Value are known. The Term, Present Value, and Future Value of lump sum transactions can be calculated using the other financial functions with Payment set to zero. The interest rate of a transaction involving a series of payments can't be determined by formula--don't try to use this function to do it.

If you're interested in more information on how the financial functions work, a little book called *Your Best Interest*, by yours truly, is a great place to learn. ($9.95: IB-001).

AppleWorks 3.0 has 32 built-in printers, three custom printers, and six new user-defined Special Codes.

Other enhancements. As mentioned in the section on the clipboard, blocks of spreadsheet data can be copied and moved to and from the clipboard with AppleWorks 3.0. Print All now prints as many columns as it can on the first page, then prints the next group of columns, then the next group of columns, until all columns are printed. In earlier versions, printing stopped after the first group of columns. When copying "Within worksheet", the user can press OA-Return to select either "No change" or "Relative" and AppleWorks will copy all subsequent cell references in that manner without asking again. OA-N or OA-R also work for this. The maximum number of rows in the spreadsheet is 9,999 on machines with 256K of memory or more.

New configuration menu

In earlier versions of AppleWorks, items 6 and 7 on the Other Activities menu were "Select standard location of data disk" and "Specify information about your printer(s)". In AppleWorks 3.0 these have been replaced by an item that reads, "Select standard settings for AppleWorks".

The "Standard Settings" file card includes the following items:

1. Change preloading
2. Select standard Spelling Checker settings
3. Change date format
4. Change time format
5. Select standard location of data disk
6. Specify information about your printer(s)

Item 1 allows the user to select which modules (word processor, data base, spreadsheet, none, any two, or all) will be loaded into memory when AppleWorks starts up. The more modules you preload, the longer it takes and the more desktop memory you use up. On the other hand, you can switch from one preloaded module to another very quickly. The default preload for 5.25 disks is "none" and for 3.5 disks it's "all."

Item 2 allows you to configure the spelling checker. There are three configuration options–name of the custom dictionary, standard spelling method ("In context" or "From a list"), and standard summary setting ("No summary", "Put summary on clipboard", "Put summary on screen", and "Screen summary only; don't correct spelling"). Defaults are CUST.DICTIONARY, "in context", and "No summary".

Item 3 allows you to select a date format that will be used throughout AppleWorks. Here's the date selection menu:

1. Mon DD, YYYY (April 11, 1988)
2. MM/DD/YY (4/11/88)
3. DD Mon YYYY (11 April, 1988)
4. DD/MM/YY (11/4/88)

With the options that spell out the month name (1 and 3) the year won't be displayed if it hasn't been specified. Slashed dates must have a year–the current year will be used if none is specified. In data base date fields, month names are abbreviated to three characters no matter which format you choose here. However, items 1 and 2 put the month first in data base date fields, items 3 and 4 put the day first. The default setting is number 1.

Item 4 allows you to select a time format. The default is AM/PM format. Here's the time selection menu:

1. AM/PM twelve hour format
2. Twenty-four hour format

Item 5, "Select standard location of data disk", is unchanged from current versions of AppleWorks.

Item 6, "Specify information about your printer(s)", works much like before but includes several new options. As mentioned earlier, the SEG.PR file that comes with AppleWorks 3.0 contains information for more printers. Here's a complete list:

```
Apple
     Daisy Wheel, Dot Matrix, ImageWriter, ImageWriter II, ImageWriter LQ,
       Scribe, Silentype
Brother
     HR10, HR20, HR25, HR35
Epson
     MX, MX/Graftrax+, FX, RX
Juki
     5500 series, 6100, 6200, 6300, 6500
Okidata
     192, 193, 82A, 83A, 84, 92, 93
Panasonic
     1080, 1091, 1092
Qume
     Sprint 5, Sprint 11
```

If this list isn't long enough, AppleWorks 3.0 allows you to have three custom printers, not one as in earlier versions. But even better than that, 3.0 allows you to customize the codes of built-in printers, as well. This means you could add an ImageWriter II to your list of printers and then change just the codes for boldface begin and end. You wouldn't need to enter a long list of new codes for your custom printer's other options as you do with current versions.

But beyond even that, you now get six Special Printer Codes. These codes are selected in the word processor using the new OA-O(ptions) SC (special code) feature mentioned earlier. Six distinct special codes can be saved for each printer, however, only one set of names for the codes can be saved. The names appear at the bottom of the word processor screen when you put the cursor on a Special Codes caret. Using the Special Codes, you no longer have to give up boldface to get italics, and so on.

AppleWorks 3.0's underlining powers have been strengthened. It can send the correct underlining codes to any printer that has underlining abilities.

When entering printer codes, the "end of code" key is now OA-Return instead of the caret. This means all 128 ASCII characters can now be entered and used as printer codes.

New file formats

AppleWorks 3.0 can read and write files created on earlier versions of AppleWorks. However, because of the new features in AppleWorks 3.0, older versions of AppleWorks will not be able to read files created by AppleWorks 3.0 *if* new features are used. If new features are *not* used, however, AppleWorks 3.0 saves files in the old standard format.

Files will be saved in the new format when the following features are used:

```
Word Processor
     One or more tab characters
     Any type of tab other than left justified
     Multiple tab rulers
     Headers or Footers
     Any of the new OA-O(ptions)
          Right Justify
          Print Date
          Print Time
          Special Codes
     More than 7,250 lines

Data Base
     Using more than 8 report formats
     More than 6,350 records

Spreadsheet
     Any new function
     Any formula that returns a label
     The use of literal strings ("for example") in functions
     Data below row 999
```

The new file formats will be made available to developers.

Claris reports that more than 750,000 copies of AppleWorks have been sold and that AppleWorks and *AppleWorks GS* own 40 per cent of the market share for integrated microcomputer software.

Miscellanea

The next question everyone has is whether they will need to upgrade their *TimeOut* programs to work with AppleWorks 3.0. The answer is yes. Upgrades will be available from your Beagle Buddy, if you have one, or directly from Beagle Bros at a price of $10 per package.

Beagle Bros has also acquired the rights to all StyleWare products (except *AppleWorks GS*, of course) from Claris. They will be released in new Beagle Bros packaging, for the most part at StyleWare's prices, on July 1.

Roger Wagner left some *HyperStudio* tips on GEnie some of you might be interested in.

1.) Here's how to make a disk that boots into your own stack. Start with a *copy* of the /HYPERSTUDIO disk. To give yourself some space, delete all of the non-folder files in the main directory except PRODOS (but don't delete anything that's inside a folder). Copy the files HYPERSTUDIO and HS.TD (Tape Deck, required for stacks with sounds) from /HS.DEMO to the main directory of your new disk. Also put the stack you want to use into the main directory of this disk and call it HOME.STACK.

Finally, go into the new disk's System folder and change the name of the file called START to FINDER and change START.HS to START. Now, when you boot this disk, the START program will start HYPERSTUDIO and it will load and display the HOME.STACK.

2.) Although HyperStudio doesn't offer a built-in palette editor, you can "fool" it into letting you use your own palette on any given card. To do this, first go to *PaintWorks Gold*, or *8/16 Paint* (640 version), and create any graphic, large or small, with a custom palette. Then save this graphic to disk. Now go to *HyperStudio*, and use the Add Clip-Art function to add just a small portion of the custom palette graphic to a card. This will change the palette for that card to your custom palette. Once added, you can even go back and delete the graphic added to the card and *HyperStudio* will retain the custom palette for that card. The custom colors will be shown in the Colors menu, and you can use them in the paint tools. If you change your mind later, you can restore the standard palette by choosing Standard Palette in the Options menu.

3.) *HyperStudio* for the moment requires an Apple IIgs with 1.25Mb of memory, and at least one 3.5 disk drive. *HyperStudio* was designed, however, to work in 768K minimum, and will with the next update.

4.) Demo stacks on the HS.DEMO disk run from about 50K to 100K in size. A "real" stack might average about 200K. For 768K systems, you'll have to keep individual stacks fairly small, but stacks can be chained together in such a way as to be transparent to the user. A 1.25Mb machine is not likely to have any particular memory problems unless you're trying to embed a lot of sounds in one stack.

Stack size varies depending on a number of factors: number of different backgrounds on various cards; how visually complex the backgrounds are; number of graphic objects used over and above the backgrounds; and number of embedded sounds and/or the largest single sound used in a stack. Briefly:

Backgrounds: For any particular background used on a card, look at the number of blocks used on the disk when the graphic is saved in a compressed format. This will be just slightly less than the amount of memory used by that background in *HyperStudio*. Solid color backgrounds compress very well and take very little memory. If a background is repeated on several cards in a stack, each new card keeps only a pointer to the original background, so no additional memory is required.

Graphic objects: When adding a graphic object, keep in mind that at the worst case, a 1/4 screen graphic would take 8K (1/4 of 32K). However, when compressed, this is likely to be much smaller, and graphics less than 1/4 screen use still less memory. It's not unusual to have each graphic object only use 2-6K of memory per image.

Sounds: Sounds take about 10K of memory per second. However, if the sounds or graphic objects are left as disk files (extended data), then only the items on the current card are loaded into memory. This means you could have megabytes of graphics and sound all used by a stack that was only a few hundred K itself, and only need as much memory as was required to load the longest sound on any particular card.

Ask (or tell) Uncle DOS

Fundamental computing

What's the best book around for beginning computer users to read to understand the fundamentals?

Robert Weiser
Hershey, Pa.

Why do so many programs start with PRINT CHR$(4)? I know this prints a control-D, but why?

Bradley Albing
Macedonia, Ohio

A very general and gentle overview of dealing with the elementary, fundamental, real-world issues of personal computing is presented in John Bear's ***Computer Wimp****, published by Ten Speed Press. We carry it in stock ($9.95: TS-001).*

Computer Wimp *won't tell you the secret of CHR$(4), however, because it doesn't discuss Apple IIs specifically, just personal computers in general, and CHR$(4) is an Applesoft convention. Applesoft was written before the Apple II had a disk operating system. When the disk operating system was added, Applesoft itself couldn't be changed, so instead the operating system was designed to intercept lines PRINTed by Applesoft that began with control-D and interpret them as disk commands. Thus, the PRINT statements you are seeing are commands for the disk operating system. A complete description of how this system works is in Chapter 12 of* ***The DOStalk Scrapbook****, by Tom Weishaar and Bert Kersey ($14.95: TB-007).*

Initial colons

I've been reading a book about Applesoft and I keep running into program lines that begin with a colon. I don't understand what this is for and I can find no explanation anywhere for the use of a colon in this location. At first I thought it was a typo, but I've seen it enough I've begun to think it must have some significance.

Bill Coleman
St. Petersburg, Fla.

Many Applesoft programmers use colons to indent lines that are within loops to make the loop structure more evident. If you try to use spaces for this, Applesoft will automatically delete them. Applesoft won't delete colons, however, and colons don't cause syntax errors or change the way the program executes. Here's a slightly modified version of one of Bert Kersey's masterpieces that demonstrates the technique:

```
100 HOME : LIST : BUZZ=49200
110 A$="!/-" + CHR$(92)
120 FOR A=1 TO 48
130 : B=PEEK(BUZZ)
140 : FOR C=1 TO A : NEXT
150 : X$=MID$(A$,A-INT(A/4)*4+1,1)
160 : VTAB 3 : HTAB 12
170 : PRINT X$;X$;X$
180 NEXT
190 GOTO 120
200 REM Press Control-C to quit
```

When you LIST this program on the screen, you'll notice some subtle differences from what we've printed here. For example, there won't be a space after the colon in lines 130 and 150. We aim for maximum readability when we print listings here, rather than a *mirror* image of Applesoft's nonsensical LIST formatting.

Current time w/o prompt

I've seen several methods of extracting the current date and time from the IIgs clock for use in an Applesoft program, but these methods only yield a snapshot. Is there any way of displaying the running date and time on the screen while an Applesoft program is executing? I've created several utility programs in Applesoft and I would like to include the running clock on the screen along with my menus.

Is it possible to eliminate the ']' prompt from briefly appearing on the screen when Basic.system is run?

Dan Seibold
San Jose, Calif.

There are two different ways to accomplish this and similar tasks. They go by the names 'polling' and 'interrupting'.

With polling, your program repeatedly takes snapshots of the clock and updates the screen whenever it isn't doing anything else. Typically, you'd create a loop that would take snapshots of the clock inside the same subroutine that collects keypresses. This subroutine would check to see if a key had been pressed and, if not, would get the time and update the screen. For this to work, you'd have to remove all of your INPUT statements and change them into keyboard PEEKs.

Here's some routines you might find useful. Replace all your INPUT X$ statements with GOSUB 30000 : X$=A$ and all your GET X$ statements with GOSUB 30050 : X$=A$. Make sure you aren't using A$ for anything else in your program:

```
30000  REM INPUT A$
30010  A1$=""
30020  GOSUB 30050 : PRINT A$;
30030  IF A$=CHR$(13) THEN A$=A1$ : RETURN
30040  A1$=A1$+A$ : GOTO 30020
30050  REM GET A$
30060  REM first get time and update screen
30061  : VP=PEEK(37) : HP=PEEK(1147)
30062  : VTAB 1 : HTAB 75
30063  : PRINT CHR$(4);"FLUSH"
30064  : PRINT PEEK(49043);":";PEEK(49042)
30065  : VTAB VP+1 : POKE 1403,HP
30070  IF PEEK(49152)<128 THEN 30060 : REM key down?
30080  A$=CHR$(PEEK(49152)-128) : POKE 49168,0 : REM
yes
30090  RETURN
```

The line 30000 INPUT A$ routine simply calls the line 30050 GET A$ routine repeatedly until it sees you've pressed Return, CHR$(13), and then passes what's been input back to you. It uses A1$ as a temporary place to store the line, character by character, as you type it in.

The line 30050 GET A$ routine begins by getting the time and displaying it in lines 30061-65. In line 30070 it looks to see if a key has been pressed. If not, it loops back and updates the screen time again. If so, it figures out what the character was in line 30080 and

returns to its caller.

Line 30061 saves the current cursor position. Line 30062 places the cursor in the upper right corner of the screen, where the time will be displayed. Lines 30063-64 use Myron Kerney's 'Flushing out the time' trick (April 1988, page 4.23) to take a snapshot of the clock and display it on the screen. Line 30065 returns the cursor to its previous position.

Try the routine and see how you like it. It's better than Applesoft's own INPUT routine in some ways–it accepts commas. It's worse in others–no cursor. Adding a cursor is beyond the scope of your question, so we'll slyly back down the hall here and leave that as an exercise for the reader, as they say.

Interrupts are usually even more complicated. You need a piece of hardware (probably the clock itself, in this case) to generate an electronic signal that tells the microprocessor to stop what it's doing and go do something else instead. That something else would be a small assembly language program that would get the time and update the screen for you. It's too complicated to explain all the details of how to do this here, but if you're interested, it's covered in Chapter 25 of ***ProDOS Inside and Out,*** *by Dennis Doms and Tom Weishaar ($16.95: TB-006). One advantage of interrupts is that you won't have to change your program at all.*

Interrupts are usually used when the task to be accomplished just can't wait–for example, when an incoming character has arrived from a modem and a program has to save it before it gets overwritten by the next incoming character. Polling doesn't work as well for this.

Polling does work well for updating the time on the screen, however, because it only happens when your program isn't doing anything else. Turning on interrupts has a tendency to degrade performance because of the time spent servicing interrupt calls.

Applesoft sends the ']' prompt to the screen when it's started up. The prompt is captured by both DOS 3.3 and ProDOS to regain control of the computer after getting Applesoft going. You'd have to patch the operating system to prevent it, which is something we don't like to do for cosmetic purposes.

Drive alignment

I have an Apple IIc with a new 5.25 Applied Engineering external drive. Anything that I save on disks in this drive cannot be retrieved (I can't even read the directory) on another IIc-Plus with an external drive. The drive speed verifies ok.

Joseph Oakes
Glover, Vt.

The drive head on one drive or the other is probably out of alignment. To figure out which drive is the one causing the problem, initialize a few disks with each one, labeling them to show which machine the disk was created on. Then try reading these with the opposite drive as well as with a few other drives. If the alignment of one of the drives is a problem, you'll have trouble reading disks created on that drive on all other drives.

Realigning a drive requires special equipment and the technical services of a dealer or manufacturer. If it's the new drive that is causing the problem, let Applied Engineering know and they'll fix it for you under their warranty program.

Fingerprint vs TransWarp

My Fingerprint interface card won't work when I add Applied Engineering's TransWarp accelerator board. Why?

David Langlois
Oswego, N.Y.

When you press the button attached to the Fingerprint card, the card tries to 'grab control' of the computer. When you're using a TransWarp, it has already used the same technique to grab control, however, and isn't willing to give it back. The user is the big loser in this battle; you can't have both cards active at the same time.

Georgia's EduNET

I want to brag on a project one of your subscribers has initiated this year to provide opportunities to isolated classroom teachers. The project is a component of the Regional Teacher Education Center here at Georgia College and is called EduNET. It allows teachers to easily and inexpensively acquire the information and the communication opportunities they need to continue to grow professionally.

What we're doing is this: first, we're placing 2400 baud modems (Epic) and the software to run on them (Point-to-Point) in all of the 200-plus schools in our service area that will agree to accept them. Second, we're providing multiple, state-wide, toll-free telephone lines so that the education community in our service area need not be daunted by the specter of outrageous telephone bills. Third, we're providing a multi-caller information service that provides for up to 64 simultaneous callers, remote information providers and editors, and that has public conferences, private email, file sharing, and searchable data bases.

Currently, the system uses software from Russ Systems (DBS or Data Broadcasting System) running on a network of Apple IIs served by a Corvus drive. Similar software for the AppleShare environment is expected to evolve from Russ System's **Lets Share** Apple II networking interface. **Lets Share** is a vast improvement over Apple's **Aristotle**.

We're very excited about the tremendous potential to do good things for teachers and those who support them using this approach. Educators outside of Georgia can take a peek at our system by calling in on their own nickel, if they like–the number is 912-453-1897.

Dr. Frank Lowney

A2-Central™

[redacted]

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 prices (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 drivel 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 unfilled 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 **GEnie mail: A2-CENTRAL**
Printed in the U.S.A. **913-469-6502**