A2-Central™

formerly Open-Apple

October 1992
Vol.8, No.9

ISSN 0885-4017
newstand price: $3.00
photocopy charge per page: $0.20

A journal and exchange of Apple II discoveries

Feeling midiSynful

by Dennis Doms

The synthLab software that comes with System 6 for the Apple IIGS gives you an environment for experimenting with creating music based on sampled sounds. The problem we'll look at this month is that the stock Apple IIGS does not include a musical input device. The solution is to connect a separate device designed for that purpose.

IIGS users are familiar with the standard ways of connecting devices to their computers. The standard IIGS keyboard and mouse connect through the Apple Desktop Bus. The Macintosh also uses this bus; Macintosh and IIGS keyboards can be used interchangeably on the two systems.

Since computer users may not buy all peripherals from the same manufacturer, there has been an impetus to create general standards that apply across different brands of products. For example, modems and printers can be connected through serial ports, which adhere to technical standards that insure that the devices will be electronically compatible and able to communicate at some basic level. Connecting a modem or printer may not guarantee that all of the features of the device can be used (that's up to the controlling software) but there is at least some expectation that a modem that adheres reasonably to the serial standard will work if the software knows how to command it.

Several years ago the music industry was faced with a similar dilemma; computer technology was being integrated into devices not only as a means of sound creation but also as a means of controlling most of the features of the instrument. Eventually it became clear that it would be advantageous to use the capabilities of one instrument, or even a separate "non-musical" device, to control other instruments. The standard that emerged was a combination of hardware and software protocols called *MIDI* (for Musical Instrument Digital Interface).

The MIDI hardware involves cabling that uses DIN-5 connectors (similar to the connector used on the IIc, but wired differently). The designated maximum length of a cable is fifty feet. The cable is shielded, with a twisted pair of wires inside. Pins 1 and 3 on the DIN-5 are not connected; pin 2 is the shield ground, pin 5 is the signal, and pin 4 is +5 volts.

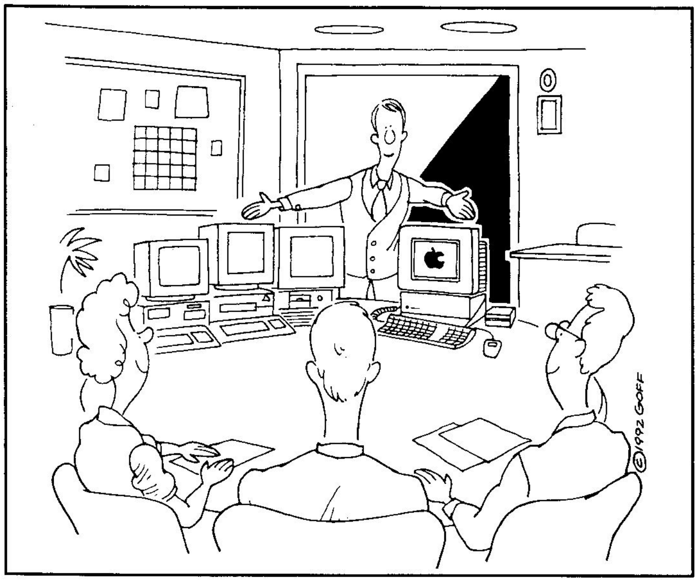
The cables are run between interfaces that consist of some combination of "In", "Out", and "Thru" ports. The "In" and "Out" ports are the signal input and output ports respectively; MIDI "Thru", if present, duplicates the input signal on its output (it's essentially a "splitter" to provide a copy of the input signal). Devices commonly supply at least MIDI "In" and "Out" to guarantee two-way communication. There are external connection boxes that will split out a "Thru" signal if your devices don't have one.

The interface communicates asynchronously at 31,250 baud (31.25 kilobaud, plus or minus one percent) with an eight-bit data word, one start bit, and one stop bit. This is not a mode supported directly by the 6551 serial interface adapter chip used in the IIc or Super Serial Card, though there have been external MIDI adapters for the IIc (one from Passport Designs) and an internal MIDI interface for the IIe (also from PassPort). You may see these interfaces described as "6850 cards", which refers to their use of a 6850 interface adapter chip.

The IIGS and Macintosh use a different serial communications chip (the Zilog 8350) which can be adapted to MIDI though an inexpensive external interface. Apple's model is $99 retail; third-party adapters like those from Opcode or Alltech can be found discounted under $50. Apple's unit includes MIDI cables; the Opcode unit I bought doesn't. I made a couple of cables from some shielded twisted-pair microphone cable and a couple of DIN connectors I picked up at Radio Shack for about $12. By way of compensation, the Opcode adapter does have one MIDI in and three MIDI out ports; the Apple interface has only one of each.

In addition, the IIGS System Software includes support for a slot-based (IIe-style) MIDI interface. Applied Engineering's slot-based Audio Animator also incorporates a MIDI interface. If you're interested, some samples of MIDI hardware circuits are shown in Hal Chamberlin's *Musical Applications of Microprocessors.*

The complete MIDI specification (hardware and software) is available through the International MIDI Association, 8426 Vine Valley Drive, Sun Valley, Calif. 91352, 213-768-7448. The current specification is. Since the hardware is basically just a means to facilitate the communication, the software protocol is usually the focus of MIDI discussions. Many MIDI devices are available and the number of features they implement vary widely. This makes talking in specifics hard, so



"SO THESE ARE OUR OPTIONS: VANILLA, VANILLA, VANILLA, OR MOCHA MOON PIE BANANA NUT."

we'll look at the elements of the protocol first and then how they apply to synthLab and MIDI Synth.

A MIDI message consists of a *status* byte followed by zero or more data bytes. Status and data bytes can be differentiated; status bytes always have the high-order bit set and data bytes have it clear. That limits data values to the range 0-127 (or -64 to +63 depending on the interpretation), so expect to see many parameters limited to those ranges. If more resolution is needed, more data bytes can be used. By using the high-order bit specifically to identify status and data bytes, MIDI prevents the confusion of commands with data and should be able to recover somewhat if a byte is "lost" in transmission.

The messages sent over MIDI are divided into two general types: *system* and *channel*. System messages send information between devices on the MIDI bus. Of these, there are three subcategories: *common* messages meant for all connected units; *real-time* messages, also meant for all units, which consist only of status bytes that individual devices can act on or ignore (since these don't use data bytes, real-time messages can be sent in the middle of other messages); and *exclusive* messages that use a manufacturer's identification code to identify a specific device the message applies to.

MIDI instruments have 16 channel numbers available to separate messages into identifiable groups. A receiving device can be configured to accept input on the desired channels and route this data to its voices in different manners.

From here on the indiscriminate use of the word "voice" can get very confusing, so let's clarify some terms. A *voice* is a sound generator (the electronics that actually create a sound). Many modern synthesizers (including the IIGS) have multiple voices. A *channel* is a MIDI term referring to one of the aforementioned 16 channels that MIDI can transfer data on. Sometimes the two terms are used interchangably, but the correspondence is not always one–to–one.

The overall way the transmitter — or in MIDI terms, the "master" — and receiver ("slave") interact with multiple channels is called a *mode*. Instruments may not have multiple voices to reproduce multiple channels of MIDI voice information, so voices may be need to reallocated. There are two controllable attributes for voice and channel assignments: *Omni* and *Mono*. From combinations of these two attributes we arrive at four modes.

One of the references I've looked at refers to the concept of modes as being confusing. This is an understatement. I've found at least three different descriptions of the attributes and one account saying that the definition of the Mono attribute has changed in recent history due to the influential implementation of a major manufacturer (Yamaha). Here's the way it seems to work, thinking of each from the viewpoint of the receiver.

Enabling Omni allows reception of data on all channels; disabling it allows the reception only on one selected channel, called the *basic channel*. For most synthesizers, the default basic channel is 1, but any channel can be assigned as the basic channel.

Enabling Mono causes the slave to use only one voice to play the data received; disabling it allows a different voice to be allocated to each channel according to the slave's internal voice mapping configuration.

One description says the master and slave are operated in the same mode, but synthLab and a couple of other synthesizers I checked seemed to send out data on all channels regardless of the mode — the mode selection seemed to affect only input.

This gives the four possible modes for the transmitter and receiver to operate in, based on the possible combinations of Omni on/off (for the channels recognized on reception) and Mono on/off (for the ability to assign voices).

Mode 1 (omni on, mono off) is referred to as "Omni" mode. A synthesizer in Omni mode will receive all channels and assign each channel to a different voice. To make the channel-voice assignments you need to configure the receiver per its instruction manual.

Mode 2 (omni on, mono on) isn't considered generally useful enough to warrant a generic name. In this mode, the synthesizer listens to all channels, but uses the messages to control only one voice.

Mode 3 (omni off, mono off) is "Poly" mode. The synthesizer listens to only its basic channel and assigns incoming data to different voices polyphonically. You are probably wondering how a single channel can drive multiple voices; it turns out that among the types of MIDI messages is one that allows a *program* (or *patch*) *change*, which many synthesizers interpret as an assignment to switch to a different voice.

Mode 4 (omni off, mono on) is "Mono" mode. Here the slave listens to only the basic channel and plays incoming data on the corresponding voice (if the basic channel is changed, a different voice will be used).

There's also a fifth "mode." It's created not by a change in the above possibilities but by the way some newer synthesizers can operate. These synthesizers can receive on multiple channels using Omni mode and can also use program changes on each channel to designate multiple voices. Such a synthesizer resembles several separate "Poly" mode synthesizers housed in a single unit. A synthesizer operating in this fashion is referred to as operating in "Multi" mode. You may see this mode referred to in some MIDI software (including synthLab).

Let's recap: there are four modes of which Mode 2 is seldom used so we can replace it with the fifth mode giving us four modes from five. Why would anyone find the common implementation of the MIDI specification confusing?

That should get us far enough along to look at synthLab's use of these features. A basic IIGS and synthLab can be used as a playback device for synthLab songs with some control over the sound quality. Add a sound digitizer and you can create new sound files, making some adjustments with synthLab. Add a sound generation MIDI device and synthLab can play recorded sequences on the MIDI device as well as on the IIGS sound circuitry. Add a MIDI device that's a controller and it can be used to "play" the IIgs through synthLab; in this configuration synthLab will also function as a multi-track recorder.

Playing through the IIgs. When you start synthLab (and click the arrow to go past the title screen) the first workscreen you see is the sequencer deck. The sequencer consists of positioning controls similar to those of a tape deck (on the left side of the screen below the selection buttons) and track controls resembling a multi-track recorder (along the right half of the screen).

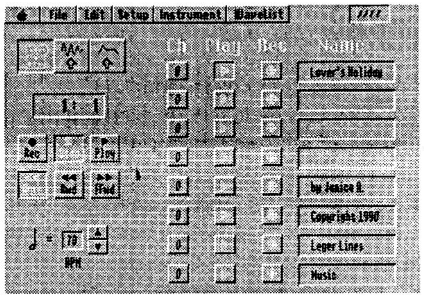
The virtual tape deck implements the familiar record, stop, play, rewind, and fast forward buttons, and a less familiar "auto" rewind button that causes the sequencer to reset itself to the beginning of the current sequence when stopped. If you want to be able to stop and then resume playing from the point at which you stopped, make sure the "Auto" label within the button is not highlighted (click the button to toggle the setting). Below these controls is a speed adjustment control that is defined in terms of beats per minute ("BPM"). The two arrows to the right of this indicator adjust the current setting (the default is loaded as part of a sequence file).

The MIDI Synth equivalent of a "song" is the *sequence*. The sequence itself is made up of MIDI codes organized as a function of time; the MIDI information determines when a note will be played, what instrument plays it, and so on (you can view the data for the currently loaded sequence using the "View Sequence..." of the "Edit" menu). Sequences can be loaded (or saved) from the synthLab "File" menu.

As we discussed in the August issue, the File menu can also be used to load waveforms into instruments and to load instruments themselves; this allows fine-tuning individual synthesizer components. When you load or save a sequence, these connected elements will be loaded or saved with it. Incidentally, in our August article we said we didn't know where the data format for MIDI Synth's records would be documented besides the MIDI Synth ERS; we now know this information has been incorporated into Byte Work's *Programmer's Reference for System 6.0.*

In order to enable the use of MIDI with synthLab you need to select "MIDI..." under the "Setup" menu. This brings up a dialog box that lets you select the MIDI mode (among those discussed previously, synthLab supports the "Omni", "Poly", and "Multi" modes), whether you want to use MIDI input (letting synthLab receive information from another MIDI device) or output (telling synthLab to send MIDI data to another device) or both, the basic channel, and the velocity compensation. The last element is one not discussed in the section on modes; the velocity compensation tells synthLab how much to react to any velocity component in the MIDI data; the larger the number, the more pronounced the reaction.

The MIDI device I used to test synthLab was a consumer-level synthesizer, a Yamaha PSR-400 (about $300), which includes 100 voice programs, a 61-key keyboard with touch response, and many other features. It has two MIDI receive modes, described as "Remote" (which seems to correspond to "Mono") and the more recognizable "Multi". Individual channels can be enabled or disabled for reception of messages and voice programs can be assigned output channels, so it fills most common requirements.



Lover's Holiday

by Janice R.

Copyright 1990

Leger Lines

Music

With synthLab set to "Omni" mode, changing the "Instrument" causes the current basic channel to change to match, so if your keyboard seems to be "dead" when playing from synthLab, double-check the setup. Under Omni mode, changing the synthLab instrument results in the MIDI data from your keyboard being played as synthLab's set instrument even if the channel is changed. With the keyboard in "Remote" (again, nonstandard) mode, synthLab played any input data on the voice it was currently set to; with the keyboard in "Multi" mode synthLab assigned the MIDI data to program voice 0 (piano). (I suspect this is because most of synthLab's included songs don't incorporate program changes, and synthLab has no way to add them.)

With synthLab in "Poly" mode, when the keyboard is played only the data being sent out on the voice channel matching synthLab's selected basic channel will be played.

With synthLab in "Multi" mode the IIGS will play each channel on the associated instrument. That is, if synthLab's instrument number 2 is a clarinet sound then all the MIDI data received on voice channel 2 will be played as a clarinet

So maybe we're not too lost; everything seems consistent.

The "Sequencer..." item in synthLab's "Setup" menu lets you select aspects of how the synthLab sequencer will act. The settings allow you to pick which "clock" (actually a timing signal) synthLab will use — either its internal clock or one supplied over the MIDI port from an external source (the MIDI master consistently sends timing commands to make sure all the MIDI devices can stay on the same beat).

There is a "metronome" you can enable that will produce a click on synthLab's speaker for you to synchronize your playing to; you can also select to have a "count off" occur before the sequencer actually starts playing (or recording).

A "Key start" button allows you to have the sequencer start recording at the first MIDI keypress. You can also select the number of beats per measure and the synchronization interval ("Beat to") for adjustments.

Most of the rest of the "Setup" items are straightforward. The "Track output..." item lets you select whether each of the 8 sequencer tracks will be sent to the IIGS speaker to play, to the MIDI bus, or both. Think of each sequencer track as a single tape recorder; the sequencer consists of eight tape recorders (actually "MIDI message recorders") all synchronized.

"System controls..." include the master tuning of the synthLab voices, the volume of the "local keyboard" (synthLab's instrument voices), and the octave range transposition (you can transpose the synthLab voices up or down an octave if you think the tune will sound better at a higher or lower pitch).

"Volume..." lets you select the output volume for each of synthLab's sixteen possible instruments, as well as the overall system volume (this adjustment is used in conjunction with the IIGS speaker volume you normally have set in the Control Panel).

Using the sequencer "deck" generally parallels using a multitrack recorder. If you aren't used to the concept, multi-track recorders allow you to select the individual recording channels ("tracks") for either playback or record as the recorder operates. Real multi-track recorders can vary from a couple to 32 or more tracks. Most professional recording studios use 24- or 32-track recorders, though some purists regard this as overkill. Most home hobbiest units support 4- or 8-tracks.

Incidentally, when we say "track" we mean a monaural (non-stereo) track; to create stereo you assign multiple tracks to record the sound from different spatial locations (real or simulated). Since the synthLab sequencer doesn't actually emulate a stereo mixer we'll skip over this complication.

The controls for the eight tracks are arranged in four columns on the right side of the synthLab sequencer display. The first column is the "Channel" for each track; this determines what MIDI channel will be assigned to that track. The fourth column contains label areas where comments can be typed.

The second and third columns contain the "Play" and "Record" buttons that determine if the track is to be used for play back or for recording new information. The practical way to use multi-tracking is to record a few tracks (as many as you can competently master at once) with at least part of your music, leaving the "extra" tracks to add more embellishments later. When you get ready to add parts, you can set the previously recorded tracks to "Play" and select unused tracks (or tracks you're ready to discard) to record the new additions. By working this way, a single person can add several instruments and sound like many people playing in concert.

Only one track can be recorded at a time in synthLab, though you can play back any or all of the tracks. I experimented with recording tracks with the Yamaha synth with synthLab in "Multi" mode, recording one track, then playing a new instrument (selected by using the MIDI channel selection to switch to a corresponding synthLab instrument voice).

The complete home studio. "Complete" may be too optimistic a word; music technology is changing about as fast as computer technology now that they are somewhat interrelated, but starting with the IIGS you already have the core of a MIDI music studio.

The IIGS already qualifies as a MIDI sound source using the built-in Ensoniq synthesizer. It also works as a MIDI controller to control the sound source, even serving as a sequencer since it can remember and store sequences of MIDI instructions. At least, this becomes true with synthLab now being distributed as part of system software.

Three items are needed to make the IIGS a true musical instrument. One is an interactive MIDI controller (such as a keyboard) to allow entering data in real time; synthLab doesn't have a way to enter musical data (though it can save and retrieve the data once it's available). That means you need a MIDI interface to connect the IIGS and the interactive MIDI controller. Finally, although the IIGS has an internal speaker, you really need an external speaker of some quality. This can be a relatively inexpensive extension speaker with a small amplifier such as those used with personal stereos or it can be a high-fidelity stereo system connected through a stereo card. But something other than the tinny IIGS speaker is strongly advised; making the Ensoniq use the internal speaker is like hooking a single LED to the IIGS video output.

If you plan to disseminate songs as computer data then you have everything you need. If you'd like to let other people hear your songs without having to cart your studio equipment around with you you'll need some type of conventional recorder; most people would probably settle on a reasonably good cassette deck. You can also buy a *mixer* to allow you to adjust volumes and balance while recording onto tape, but from there we'll leave you to the services of a good audio consultant. If you plan do use the IIGS to augment a home studio based on musical instruments, you probably should talk to someone at a good professional music store.

synthLab also wasn't meant to displace commercial programs, but it is a good place to start experimenting. For more detailed information, one company you may want to contact is Sound Management, Electronic Music Products, P.O. Box 3053, Peabody, Mass. 01961, 508-531-6192 or 800-548-4907, which provides product sales and advice on computer music products and MIDI interfacing.

Apple Writer: A guided tour

by Ron Evry

Granted, AppleWorks is the overwhelming favorite Apple II word processor today. But *Apple Writer* was the number 1 word processor before that. Even today it has an enthusiastic following and it has features found in no other Apple II word processors.

In July, *Apple Writer's* author, Paul Lutus, agreed to let the Apple II RoundTable on GEnie distribute the software as freeware. This was the first major hit of the RoundTable's "Lost Classics" effort, which is the brainchild of Tim Tobin, the RoundTable's librarian.

Since *Apple Writer* now has a very attractive price, it seems sensible to share some knowledge about *Apple Writer's* operation and capabilities. This series of articles will examine *Apple Writer's* features, giving tips and advice to those interested in learning what *Apple Writer* has to offer.

***Apple Writer* was once the most popular software package made for the Apple II.** And deservedly so. However, somewhere along the line Apple Computer decided to market AppleWorks exclusively and took *Apple Writer* off the market. The fact that Paul Lutus has released *Apple Writer* as freeware means that a very, very powerful tool has been made available free to every Apple II user. Its basic word processing functions are so simple to learn that anybody can be running it in about five minutes. But it has a host of other built in features designed to accomplish many tasks, and best of all, it has two very powerful programming functions that enable anybody to fully customize the program.

One of these functions is a built in programming language called WPL ("Word Processing Language"). WPL is remarkably compact, having only fourteen basic commands and an additional twenty-one "printer" commands. These enable a user to perform any number of complex functions. WPL programs exist to do mail merge, count the number of words in a document, read a series of files on a disk and pull out desired information from them into a new file, display text material to students and then quiz them on it while keeping and saving the score, and an assortment of other tasks. These are limited only by the user's imagination.

The other powerful tool available to *Apple Writer* users is the built in glossary function. Similar to the *Time-Out UltraMacros* program for AppleWorks, *Apple Writer's* glossary enables the user to enter and save a series of keystrokes, which can include *Apple Writer* commands, and execute them. These keystrokes can be as simple as a frequently used word or as complex as commands that execute a series of WPL programs, one after another, at the press of a button.

Not only can glossary entries call up WPL programs, but entries can be nested into each other up to eight deep. They can be used to simplify insertion of imbedded printer commands (a glossary called Special comes with the *Apple Writer* disk, which contains printer codes for the ImageWriter), or perform mathematical functions like a pop-up calculator. Different glossaries can be called up as needed, or a start-up WPL program can be put on a disk to automatically load them.

Besides *Apple Writer's* built in customizing features, there are a number of keyboard shortcuts available all of the time. Many of these are similar to AppleWorks commands, such as the Find function, which enables the user to search for strings and replace or delete them, either singly or throughout a document. Split screens can be had with a single keystroke, enabling the user to work on two different parts of a document at the same time. Blocks of text can be deleted or copied easily and you can peek at a document on a disk without disrupting the file currently in memory.

Extremely large files can be loaded in pieces directly off of a disk, and conversely, parts of a file can be saved to disk separately. Multiple files can be joined together into single text pieces. Custom-designed "help" screens can be created, disks can be formatted from within the program, and there is even a built in telecommunications module that will work with a modem.

So how does one use *Apple Writer*? In this first installment we will try to explain how to get the program off and running in as short a time as possible. The next part will explain how to use some of the more complex built-in features, and then, if there's an interest in future installments, we can explore the use and making of glossaries and WPL programs. In addition to that, I'll provide the names of some people to contact for incredibly jazzy patches to *Apple Writer.* These enable you to use as much memory as your machine has for a document, to put custom characters and high resolution graphics in your printed output, and to produce better quality output on a LaserWriter than a Macintosh can!

Assuming you have an *Apple Writer* disk already (if you have downloaded it and you want to be able to boot it, you'll need to add the file ProDOS to it), you should make a backup copy and store the original in a safe place. Of course if you do patch the heck out it and need to start over, you can always download it again off GEnie.

If you have an Apple IIe or an Apple IIc, you can boot the program right up. If you have an Apple IIGS,you will need to play with the program's machine code a little bit before you will be able to print your documents. The instructions for IIGS owners are at the end of this article.

Okay, so you've booted up the program, and the first thing you see is a screen that gives the number of the version of *Apple Writer* you are using, a copyright notice and an instruction to press RETURN. Please do so. The next thing you will see is a blank screen with a bar at the top. There are a bunch of letters and numbers displayed on the bar. Press the ESCAPE key once and the bar is replaced by tab markers. These can be changed by moving the cursor to the tab position you want (or where you want to get rid of a tab marker) and pressing CONTROL-T Then type "S" to set or "C" to clear. Pressing "P" will eliminate all of your tabs, if you need to do so. Next time around, we will show you how to make your tab settings permanent, if you so desire.

Pressing the ESCAPE key a third time will totally remove the bar and tab line, giving you a completely blank screen to work with. Pressing ESCAPE once more puts you back to where you started.

If you'd like to start writing a document from scratch, just go ahead. The computer is ready for you. You'll find that the cursor moves as fast as you can type. If you are a superbly fast typist you may see an asterisk appear on the bar line just to left of that "Z". This is telling you that you're typing so fast that the computer is storing your keystrokes until they can be put on the screen. If you'd like to see this feature, hold down a key and let it repeat itself over and over. Then do the same thing with your DELETE key until you've gotten rid of all those letters you don't want. Notice how fast it all is?

Remember, you don't have to press the RETURN key at the end of every line. *Apple Writer* has a built in word-wrap feature that moves the cursor down to the next line by itself. Generally, you should only press RETURN at the end of a paragraph, to automatically move to the next line.

Once you've started writing, you may have occasion to move around the screen with your cursor, perhaps to make changes, or to read some part of the document not currently on your screen. The simplest way to move around, of course, is to use the arrow keys. A faster way to move around is to hold down the Solid Apple/Option key and press the arrows. The left and right arrows will jump from word to word and the up and down arrows will jump twenty-two lines. To move all the way to the beginning of a document, hold down the Control key and press "B" (either upper or lower case will work with control characters). To jump to the end, press Control-E. See how logical this all is? (By the way, if you're a confirmed Appleworks user and you'd like to use the Open Apple key instead of the Control key, this can be arranged easily through the use of glossaries. More about that next time around).

If you need to make changes in your document, there are a number of ways to do this. The simplest way is to position your cursor to the right of the characters you'd like to get rid of and press the DELETE key as many times as you need to. This technique can be time consuming if you have lots of words, sentences or paragraphs to delete, so there are some keyboard shortcuts available to you.

To get rid of words, move your cursor to the right of the first word you need to delete. Then look up at the arrow type character on the extreme left of the top bar. It should be pointed to the left. If it is not, then press Control-D to change its direction. You will notice that Control-D is a "toggle" key. You can push it over and over and it will change the direction of the arrow character back and forth. Next, press Control-W ("W" for "word") and it will delete an entire word. You can repeat this word after word, until you've eliminated whatever you like. Now suppose you eliminated too many words. You'd like to get them back, right? Just change the arrow direction with Control-D to point to the right and press Control-W until you've gotten back the words you want. This feature is handy for moving words around in a document. Simply delete them from one place, move your cursor someplace else, and put them back.

This feature is not limited to single words. Pressing Control-X will do the same thing to paragraphs, grabbing everything up to the next carriage return. There is a limit of 1,024 characters in the text buffer area, so be careful about using it for very large blocks of text. Anything larger than that will push the first characters out of the buffer for good. If you really have a need to manipulate very large blocks of text, there is another way of doing it that uses the disk as a buffer. More on that next time.

Sometimes, you may want to simply type over words and change the text without deleting it first. *Apple Writer* has an overstrike feature that will do this for you. Just press CONTROL-R to toggle the "replace" mode and you may then type directly over characters you need to change. When you wish to leave the "replace" mode, press CONTROL-R again. An "R" will appear on the top bar just to the left of the "Z" while you are in that mode.

Eventually you will want to print your work. Ninety per cent of the dot-matrix and daisy-wheel printers in existence should work with *Apple Writer* "out of the box." To print your file, make sure that your printer is on and properly connected to your computer. Then press CONTROL-P. When you see the prompt that says "[P]rint/Program:", type "NP" and RETURN and watch the paper fly.

If your print-out is skipping lines, then you will need to tell *Apple Writer* not to send line feeds to your printer. To do this, press CONTROL-P, then type "LI0" and RETURN. This sets your line feeds to zero. Next time around, we will show you how to make this setting permanent and also explore the various print formatting options available to you. If you're curious and would like to play around with these options in the meantime, press CONTROL-P, type in a question mark and press RETURN. You will see the array of pre-set Print commands displayed. Have fun, and remember that you can always reboot and get back the default values.

If your printer is some very obscure machine that won't work at all, there are some adjustments that you can make. Some time or other in the future, we can explore that topic, if there is a need to do so.

A unique feature of *Apple Writer* is that the program is entirely memory resident. What this means to the user is that once it is loaded into the computer, the *Apple Writer* disk can be removed from the drive (except to access the "help" screens and the WPL programs and glossaries on the disk). Apple II users with one disk drive will find this very handy, as the need for constant disk swapping and flipping required for programs like AppleWorks is not necessary.

You will need data disks to save your documents on. It helps to have some ProDOS formatted blank disks on hand, inasmuch as the *Applewriter* disk has very little space on it to store files. If you have just typed a document and found that you have no data disks on hand, you can format one from within *AppleWriter.*

First of all, press CONTROL-O to bring up your Options screen. Press "I" ("Format Volume") and at the prompts tell *Apple Writer* what slot and drive numbers your blank disk is in. Be careful that you have removed *Apple Writer* from that drive or you will erase everything on the disk permanently! A typical 5.25 disk drive is in slot 6, drive 1. You will need to name your volume at the prompt, and you will be given a chance to back out at the last moment, just for safety's sake.

If you have an Apple IIGS and you have patched *Apple Writer* , you may find that the formatting option does not work, so be prepared with some blank media. In a pinch, you can save your work to /RAM5, exit the program and format a disk. If you have a hard drive, you ought to have room for your file.

Once you have a blank disk, you may save your file in a variety of ways. By pressing CONTROL-O, you can press "H" and set your prefix volume to the name of your data disk at the prompt. You must use ProDOS protocol in writing your volume name here (ex: /MY.DATA, not MY.DATA). Once you have done this, then you will have an easier time saving and loading files to that particular disk.

When you are ready to save your file, you just press CONTROL-S (for "save") and type the name you wish to give your document. If you are writing a long document, you may wish to save it to disk every once in awhile as you are writing it, in case the electric company decides to shut down your power while you are working, or some other silly thing like that happens.

After the first time you save your file, the top bar will display the name you saved it under. The next time you press CONTROL-S, the name of your file will be displayed for you at the prompt. Unless you need to save it under a different name, hold down the right arrow key until you have gone past the filename, then press RETURN. You will be asked if you wish to delete the old version of the file that is on the disk. This simply means that you are replacing what you have saved before with what you are saving now. If you do not want to do this, press "N" to go back and save it under a different file name.

If you wish to save a file on a disk that is not the set prefix volume, you must use the complete ProDOS file name (ex: /MY.DATA/LETTER.1). You may include existing subdirectories if you wish (ex: /MY.DATA/LETTERS/LETTER.1), and if you'd like, you can use the Options menu (CONTROL-O) to create a subdirectory. If you are typing in a complete filename, typing "./" will do for the volume that is already set or will accept whatever volume has been put in that drive.

Once you have saved your file, you can clear it from memory by pressing CONTROL-N (for "New"). You will be asked if you wish to delete the file from memory, just to prevent you from accidentally erasing something you have not saved to disk.

Loading a file from a disk is very much the same process as saving a file. You press CONTROL-L (for "Load") and type in the name of the file you wish to put into memory. The same rules apply about pathnames and preset prefixes. The only difference here is that you may need to find out the names of the files you can load. There are different ways to do this as well.

If you have preset the volume prefix of your data disk, then press CONTROL-L, then a question mark, then press RETURN. You will be given a catalog of available files and a prompt for the name of the one you wish to load. Another option is to press CONTROL-O, and at the Options menu, select "A" ("Catalog"). Here you may press RETURN to catalog a pre-set volume prefix, or type in a full path name. If you do not know the names of your on-line volumes, select option "F" first ("List Volumes On-Line"), then type in the path name.

You may wish to leave *Apple Writer* at some time. To quit into ProDOS, press CONTROL-Q, and from the Additional Functions Menu, press "J" ("Quit *Apple Writer*"). You will be asked if you wish to lose memory contents, just in case you forgot to save your work first.

This information should be enough to get you started with *Apple Writer*. Next time, we will explore many of the unique features that make this program so versatile. Until then, have fun using and exploring *Apple Writer*.

As I promised, here are Don Lancaster's patches for enabling your Apple IIGS to print with *Apple Writer*. First, Lancaster recommends using a Super Serial Card in your IIGS rather than these patches if at all possible. If you can't do that, here's the patch information:

Filename: AWD.SYSTEM, T$0C

Apple Writer v2.1

Offset from begining of file	Old Value	Change to
$2DC7	$A0	$60
$2F7E	$01	$10
$2F85	$31	$13

Apple Writer v2.0

Offset from beginning of file	Old Value	Change to
$2DB0	$A0	$60
$2F67	$01	$10
$2F6E	$31	$13

For those of you who don't know how to do a patch with this information, we've included a program on this month's ***A2-Central-on-Disk*** that will make the patch to v2.1 for you. The program is also available online in our library on GEnie. It's called AW.PATCH.BXY.

Miscellanea

Not too long ago, we received a copy of Cecil Fretwell's new *UltraCat* in the mail. I've found it real handy for cataloging our back issues. And if we ever start any Macintosh-based disk publications, I'll use it to catalog them as well since it reads HFS and DOS 3.3 disks as well. One thing that distinguishes this utility from others is the fact that it translates file types to English. *UltraCat* also provides a detailed summary of disk information such as, space used, space available, the number of standard and of hidden files, directories, and even how much memory *UltraCat* used to analyze all this information. There's a unique way to search for "lost" files called masking. UltraCat is an Apple IIGS application and comes complete with a very readable manual for $25.00. For further information write to Cecil Fretwell, 2605 Highview Avenue, Waterloo, IA 50702.

I've got GEnie's Tim Tobin on the phone right now with an area code correction. His correct phone number is 310-813-5697. We printed it wrong on page 8.63 in the September issue. He also reports that the next Lost Classic to come to the surface is Don Thompson's WPL Extension Kit for Apple Writer 2.1.

Game lovers have a few things to look forward to in the not-so-distant future. I couldn't tear my daughter Jamie away from Procyon's booth at the Apple Central Expo. The reason? A game that they are marketing called *Pick 'n Pile*. It's one of those addictively obsessive little arcade game that could just get you fired if you were so careless as to install it on your hard drive at work. (Gulp) For further information cont+act Procyon, Inc., P.O. Box 620334, Littleton, Co. 80162-0334, 303-993-4649.

My son Eric was intrigued with DreamWorld's (P.O. Box 830, Iowa City, Iowa 52244-0830) upcoming *TimeLord* by Chris McKinsey as well as Interplay's (3710 S. Susan, Santa Ana, Calif. 92704, 714-545-9001) *Out of this World*. He was able to snag a demo of *TimeLord*, which should be out sometime this fall. It's an Ultima-style game with incredible graphics and sound. *Out of this World* is already out for MS-DOS machines with a version for the Super Nintendo not far behind. And thanks to Bill Heineman's love of the Apple IIGS, we're going to get it too.

The library of GEnie's Apple II Roundtable has demo versions of *Pick 'n Pile* (file #19264) and *TimeLord* (file #19244) ready for the download. There is also a small file (#19276) that fixes a tiny bug in the PNP demo. Enjoy.

Seven Hills (2310 Oxford Road Tallahassee, Fla 32304-3930, 904-575-0566) is a prolific company. They've just introduced a new utility called *Kangaroo*. Appropriately. *Kangaroo* allows you to hop from one folder to another. It remembers where it has been, which makes accessing things faster than ever. *Kangaroo* also provides the ability to perform many functions from within the application you are using. You can perform quick file searches, change filetypes, delete files, rename files—well, you get the idea.

Did you know that there is a hidden feature in AppleWorks GS 1.1 that allows you to access font sizes larger than 48k? Simply hit the OPTION-SHIFT combination while in the Choose font submenu. This tip is especially useful to access the larger sized TrueType fonts available with *Pointless*. And while we're on the subject of *Pointless*, to figure out which version you have, double-click on the *Pointless* icon in your control panel, then press CONTROL-OPEN APPLE-A. You'll see "Pointless 1.0 Revision 1", if you have the update installed.

As vendors of Apple II software dwindle, local user groups become more and more important. I'm interested in knowing the status of our users groups, both nationally and internationally. How have they changed? How has the composition of your membership shifted through the years? What sort of meetings or membership campaigns have worked and what hasn't? We'd like to hear from you on this issue. It's an important one that needs addressing. Perhaps if we can share our successes and our frustrations, we can find a way to meet the needs of a changing and challenging computing environment.

If Morgan Davis had been present at the *A2-Central* Summer Conference, Jay Jennings could have used Davis' new product, *RADE* instead of the multiple cans of bug spray that provided sight gag after sight gag at the Roger Wagner roast. *RADE* is a Real-time Applesoft debugging environment with powerful features such as breakpoints, variable watching, program flow tracing, single statement stepping, variable modification, online help, and debugging history available from within applications. It is compatible with ampersand extensions and MD-BASIC programs. It requires an Apple IIGS and retails for $49.95. For further information contact The Morgan Davis Group (sounds like a jazz quartet, don't you think?) at 10079 Nuerto Lane, Rancho San Diego, Calif. 619-670-0563, fax 619-670-9643.—edr

Ask (or tell) Uncle DOS

Mixed reviews

I found the article "How to be an Apple II Programmer" by Matt Deatherage in the September issue of ***A2-Central*** to be one of the most worthless articles I have ever read on computer programming. Although the article is almost four pages long, the little useful information contained within it would easily have fit within a single page. How many of you who read this article feel you learned anything significant about programming your Apple II from this article?

Even the little useful information contained within this article was sometimes misleading. The part of the article that disturbed me personally was the tiny section which begins "Next you need a development environment if you don't have one." This topic is so important to software development that there should have been much more than the very small discussion we got. In this section, Deatherage makes the statement, "If you program your Apple IIGS in Pascal and C ....;" and then goes on to plug products from ByteWorks. The only other language he specifically discusses in this section is assembly language. Why is he so vague on other languages? What's behind the omissions?

In this letter, I'll ask the question Matt Deatherage refused to ask. What if you don't wish to program your Apple IIGS in Pascal, C, or assembly language? What if you wish to program in the easiest to learn of all the computing languages, BASIC? This question would probably interest most of you who wish to program on the Apple IIGS because most of you were introduced to computer programming on your Apple II by Applesoft. Would most of you choose to learn a completely new computer language when, what you want to do can be done with a computer language with which you're already familiar? If you have an interest in BASIC and wish to go beyond Applesoft, the obvious choice is Micol Advanced BASIC for the Apple IIGS from Micol Systems; the only serious Apple IIGS BASIC currently supported.

Deatherage cannot claim ignorance about the Apple IIGS version of Micol Advanced BASIC for the simple fact that I discussed Micol Advanced BASIC with him at Apple Computer several years ago when we met there. We were present at KansasFest last year; ***A2-Central*** made an announcement about Micol Advanced BASIC version 4.0's release earlier this year and it has been continuously offered in their catalog.

If Mr. Deatherage had any interest in any languages other than Pascal, C or assembly, he could have called us; we are very easy to reach. He would have found out that not only are we still selling Micol Advanced BASIC, we are still very actively supporting it. For example, version 4.2 of Micol Advanced BASIC for the Apple IIGS is currently in beta testing and will be released shortly.

When he discussed the 8 bit Apple II computers, the Apple IIe and the Apple IIc, the only products he specifically mentions are assemblers. Beyond this he only makes general statements. How many of you who program on the Apple IIe or IIc program in assembly language? What about Micol Advanced BASIC for the Apple IIe and IIc; the only higher level computer language for 8-bit computers still being supported and promoted? On this topic, he is silent. By the time you read this letter, version 4.5 of Micol Advanced BASIC for the Apple IIe and IIc will have been released.

Another disturbing part of this article was when Deatherage compared ORCA/M with Merlin; he implies that ORCA/M is the more desirable of the two products. Judging from the phone calls we receive here at Micol Systems, Merlin is preferred by people who program in assembly language. While Merlin is liked by almost all of the callers, ORCA/M has received a mixed reaction; some liked it, others don't. I see no reason why these responses wouldn't be representative of what the Apple II market feels about the relative merits of each product.

When I called Resource Central to complain about the above mentioned article, I was offered the possibility of writing an article on BASIC programming for an upcoming issue of ***A2-Central.*** Normally, because I am busy improving and promoting Micol Advanced BASIC, I can't take the time for such things. However, because this article by Matt Deatherage says so very little about BASIC, I will make the time to write such an article if there is enough interest. If you wish to see and article about BASIC on the Apple II, please let ***A2-Central*** know. Mention what topics, in particular, you'd like covered. I will do my best to be as informative and objective as possible; something I feel is lacking in Matt Deatherage's article.

Stephen Brunier, President
Micol Systems
9 Lynch Road
Willowdale, Ontario
Canada M2J 2V6

I liked Matt Deatherage's article, "How to be an Apple II programmer," (***A2-Central***, Sept 1992, pp 8.57-8.60). But what I would like to know is: Who's hiring?!

After having and studying an Apple II Plus and an enhanced Apple IIe for 15 years, I have become quite an expert on using and programming the 8-bit systems. I would like to make a career change and work for a company using my knowledge of Apples and ProDOS 8. The work could be full-time or part-time, using an employer's computer systems in their offices or on my Apple II system at home.

So who is hiring? Would you publish or send me a list of companies interested in hiring expert (self-taught) Apple II programmers?

James P. Davis
Hayward, Calif.

As Matt indicated in his article, one of the best places for Apple II information is the online services. It's also a good place to introduce yourself to potential Apple II employers. If you don't already have a modem, get one and join GEnie or America Online. Become active, show off your knowledge, answer questions intelligently. Write some freeware and make sure it gets uploaded to the major boards. It's not a secret that this is exactly how the now-famous (and employed) Apple II celebrities like Andy Nicholas and Jay Jennings got their starts. You never know.—edr

Kfest kudos

I attended my first KansasFest this year! How great it was! From the moment I arrived (at 10 p.m. Monday), I could see that this event was going to run smoothly. And it did. The meals were fine and the accommodations entirely adequate. Events came off on schedule and, from outward appearances, without a hitch. But the best part of all was the people I met. Everyone was friendly and anxious to communicate. Understand that I am 63 years old and strictly an amateur programmer and hardly the one rushes to for advice in this group. I felt very comfortable. I particularly enjoyed talking to HangTime and Bo Monroe. I'm just sorry that I didn't have the opportunity to meet Tom Weishaar personally. I would especially like to thank Bo for all the time he spent with me (even at the airport) and for the insight he rendered. I knew of him only from my subscription to ***Script-Central***—but what a great guy.

My thanks to you and your crew for putting on a highly professional conference and I look forward to seeing you all next year.

Woodruff Ogden
Walnut Creek, Calif.

More fax

Regarding Peter van Dongen's letter in the July issue of ***A2-Central*** (AE fax problems), I learned from some helpful person on GEnie (I wish I could remember the name) that the fax CDev need only be inactive during the boot time. After booting the system, you can go to the Finder, make the CDev active, and have access to the phone book and all the fax software parameters as you are normally used to doing. Just remember to make it inactive when you are finished.

Michael Nickolas
Lynn, Maine

What's the Plus in IIc-Plus?

Every one in a while, I hear about an Apple IIc Plus. How does it differ from a regular IIc? Is the software compatible?

Geraldine Wright
Hyattsville, MD

Major differences in the two computers are that the IIc Plus has a built-in 3.5 drive, rather than a 5.25, it runs twice as fast, and the power cord plugs directly into the computer, it doesn't have to run through a power "brick" first. Software is compatible. The Apple IIc Plus was released late in 1988 and is no longer sold by Apple. See our November 1988 newsletter for more information.—TW

User group members wanted

I was a *Nibble* subscriber and didn't know what I was missing. Received this first copy and can't wait for the next.

I've just become president of the PenApple User Group in Redwood City, Calif. and seek to attract new members. Our meetings are open to anyone who has interest in Apple II or Macintosh computers.

William Lakner
1399 Kentfield Avenue
Redwood City, Calif. 94061

A2-Central™

Publisher: **Tom Weishaar**
Editor: **Ellen Rosenberg**

with help from:
Denise Cameron, **Dennis Doms**, **Sally Dwyer**, **Dean Esmay**, **Richard Ginter**, **Mary Johnson**, **Jeff Neuer**, **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): $34 for 1 year; $60 for 2 years; $84 for 3 years. All back issues are currently available for $2 each; bound, indexed editions of our first six 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 $90 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 20 cents per page per copy distributed.

WARRANTY AND LIMITATION OF LIABILITY. We 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. OUR LIABILITY FOR ERRORS AND OMISSIONS IS LIMITED TO THIS PUBLICATION'S PURCHASE PRICE. In no case shall our company or our 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**

Assorted queries

A few short questions:

The label from my *Nibble* subscription said my subscription expired in 2-93. Why does your label say 1-93?

Does anyone know where I can buy a secondhand Apple IIGS motherboard for my IIe besides Shreve Systems and/or Pre-Owned Electronics? i.e.< $350.00?

I'm looking for books on interfacing my IIe to the outside world. Any ideas?

Howard K. Jones
Riverside, R.I.

A few short answers:

*The expiration date on your **Nibble** label showed the first issue that you wouldn't receive and our label shows the last issue that you will receive if you don't renew. You haven't lost an issue, it's all in the semantics.*

You might want to try Shreve Systems (3804 Karen Drive, Bossier City, LA. 7112, 800-227-3971) one more time. They are currently running a special on the Apple IIe motherboard upgrade. While supplies last, you may send them your Apple IIe logic board; they, in turn, will send you back a IIGS board, mounted in the bottom of a IIe case, for only $149. Call for further details.

*Two books published by Vernier Software (2920 S.W. 89th Street, Portland, Ore 97225, 503-297-5317) will help you connect your Apple II to the outside world. The titles are **Chaos in the Laboratory** and **How to Build a Better Mousetrap**. They can be ordered directly from the publisher or from us. And if this subject interests you, as it obviously does, don't miss Art Coughlin's article "Science Fiction Theatre" in the April 1992 issue of **A2-Central**.—edr*

Missing language

In your back issues you've mentioned lots of languages for the Apple II but I've never seen you mention *Hyper-C*. This freeware program is highly compatible with ANSI C and the examples in the K & R "C" reference. I have it in its entirety on a 3.5 disk with full documentation, samples, conversion charts (from other "C" standards) and a hires library. I hope to put out some worthwhile 8-bit software with this magic. If anyone sends a SASE with 3.5 (preferably with goodies on it, but blank will do) to me, I will pass along a copy.

Stephen Craft
LPO 16068, P.O. Box 5064
New Brunswick, N.J. 08903-5064

GEnie subscribers might be interested to know that Hyper-C is also available in GEnie's A2Pro RoundTable Libraries (page 530, option 3). The library also contains a patch (file # 1917) to allow it to run on an Apple IIGS, the documentation (file #1741) and a file called HC.MakeSys.bxy (file #2835) that makes system files using Hyper-C.—edr

Last month in *A2-Central* on disk:

Directory: /A2.ON.DISK.9209/

Filename	Blocks	Description
GENERAL.STUFF________	1	Stuff for all Apple II systems
BRANDT.TRANCRPT_____	15	Conference w/Randy "Macroman" Brandt
CREATURE.GAME______	26	New shareware arcade game
MAMMALS.CLIPS.1______	99	Double Hires clip art
SCRAMBLER.GAME_____	30	A neat four-letter word game
IIGS.STUFF____________	1	Stuff for IIgs users in here
MCOPY.COMMAND______	99	MS-DOS copy util - reqs shell
RESLIN.GS__________	83	ResLin, a
SCARABAEIDAE.FE____	11	Finder Extra - removes Finder.DATA files
SPACE.ARROW.2.1____	411	A great one/two player space arcade game
SPACE.ARROW.AID____	65	Design your opponents for Space Arrow
TEXT.COLOR.UTIL______	9	Alt for changing text colors
TECH.NOTES__________	1	Finally! The Apple tech note updates!
TN.91.12.UPDATE______	72	All current notes from 3/91 - 12/91
TN.92.06.UPDATE______	496	6/92 Apple Tech & Filetype Note update

Last month in Script-Central:

:Script.Central:ScriptC.Sep92

AnimIcons.Triad	87	Sampling of commercial pkg
ColorEdit.NDA	00	Lets you edit colors of current app
CursorAnims2.2	61	More animated cursors
Doctor.Who	442	All about the infamous doc
Dr.Who.Snds	449	Sounds
HyperAnimaux	171	Kid's memory game from France
Mile.Minder	180	Keep track of your car's expenses
Puzzler	36	Addictive and mind-joggling
Script.Cntrl.08	1292	The recurring features: Q & A, Tips & Tricks, The Bath room, Sounds, Mail & more
TheLounge	263	From GEnie's A2 RoundTable
TopTen	33	Fun from KansasFest
xCat.XFCN	58	Reads directories from HyperTalk

Last Month in Time-Out Central:

About TOCentral	6	
Browser	12	
Letters	9	From the readers
Picstack.Docs	6	Major product of this issue
Picstack.Macro	52	The macros that make it go
TO.Picstack	9	Use DHR pix with AppleWorks
Checkbook.1992	34	A file from last month
TO.DHGR.Viewer	3	Orginally from TO-C 11/91
TimeOut.Huddle	15	Randy's soapbox
JEM.Software	7	Info about UltraMacros 4.0
Nelken.Series	47	UM 4.0 tutorial
Quality	7	TO.Grammar press release
Pictures	1365	For use with Picstack