A Database Publication

# apple user

Vol. 5 No. 7 July 1985 £1

## Apples at the heart of respiratory medicine

Make the most of extra memory

Enhanced labelling routines

New DOS patches listed

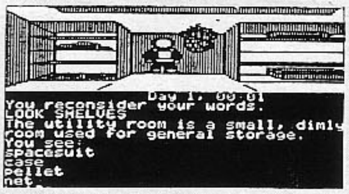
Flowcharting with MacPaint

REVIEWS: FingerPrint · Printerrupt · Cut-price Apple adventures

## News

● Asian pirates finally clone the Apple IIe, Macintosh network run by an Apple IIe, and more. 5

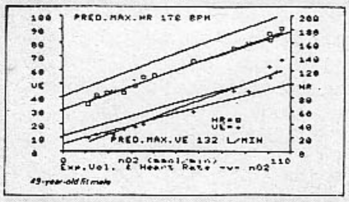
## Fun & Games



```
                 Day 1. 00:01
You reconsider your words.
LOOK SHELVES
The utility room is a small, dimly
room used for general storage.
You see:
spacesuit
case
pellet
net
```

● The accent's on cut-price adventures this month, plus a look at Adventure Writer for the DIY enthusiast. 9

## Application



● Apples at the heart of the Pulmonary Function Laboratory in Papworth Hospital. 17

## Pascal

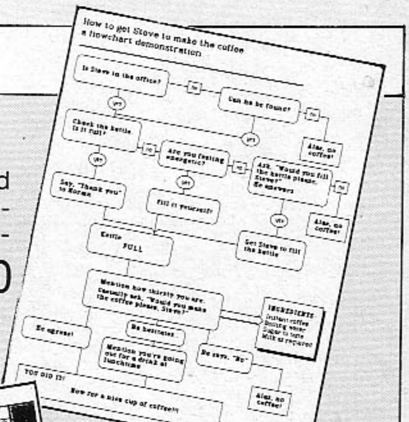
● Stuart Bell continues his series of Pascal Tutorials, using files of records to handle large volumes of information. 19

## Utility

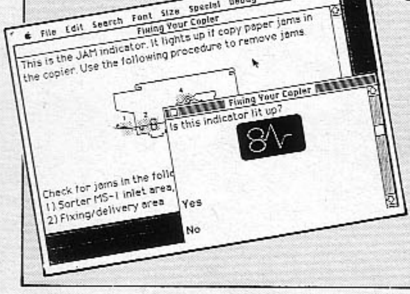
● Reassign the default load address of binary files with this useful routine. 22

apple user
Volume 5
Number 7
July 1985

## Macintosh


How to get Steve to make the coffee
a flowchart demonstration
Is Steve in the office?
Can he be found?
Check the kettle. Is it full?
Are you feeling energetic?
Ask, "Would you fill the kettle please, Steve?" He answers
Say, "Thank you" to Norma
Fill it yourself!
Alas, no coffee!
Kettle FULL
Get Steve to fill the kettle
Mention how thirsty you are. Casually ask, "Would you make the coffee please, Steve?"
INGREDIENTS Instant coffee Boiling water Sugar to taste Milk as required
He agrees!
He hesitates
He says, "No"
Mention you're going out for a drink at lunchtime
YOU DID IT!
Now for a nice cup of coffee!!

● Duncan Langford shows how easy flow-charting is with MacPaint. 40



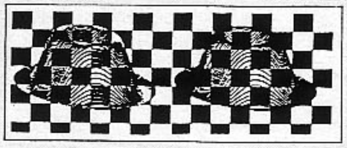
This is the JAM indicator. It lights up if copy paper jams in the copier. Use the following procedure to remove jams.

Is this indicator lit up?

Check for jams in the follo
1) Sorter MS-1 inlet area,
2) Fixing/delivery area

● Bill Hill gets a sneak preview of Guide, a powerful package tipped to become the next Filevision. 39

## Review



● Bill Allen compares Thirdware's Finger Print with Dark Star's Printerrupt. 24

## Programming

● Use that extra memory on your extended 80 column card with the help of Paul Overaa's routines. 31

## MicroLink

● Your chance to join MicroLink, the new micro communications system tailored to your needs. There's full details of what this newly-opened service offers Apple users, together with an application form. 34

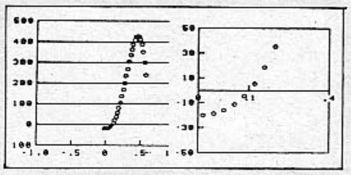
## Appletips

● Beat the bug in the ProDOS chain command, and add lines to your program which can't be deleted. 44

## Books

● Books on Business Basic for the Apple III and Epson printers are reviewed this month. 46

## Graphics



● In Part XIV of the Apple User Graphics Library, Peter Gorry provides some powerful labelling routines. 49

## DOS

● Francis Riley provides some excellent enhancements for your DOS. 53

## New Products

● The latest ways to spend your money. 58

## Feedback

● Three pages of readers' letters on using MacTerminal, the RWTS and more. 61

## Classifieds

● Bargain offers from your fellow Apple users. 64

## Order form

● Subscriptions, back issues, binders to keep them in. 66

*Apple User Managing Editor* **Derek Meakin** *Features Editor* **Cliff McKnight** *Technical Editor* **Max Parrott**
*Production Editor* **Peter Glover** *Advertisement Manager* **John Riding** *Sales* **Chris Riding**
*Design* **Heather Sheldrick** *News Editor* **Mike Cowley** *Editor-in-Chief Database Publications* **Peter Brameld**
Telephone: 061-456 8383 (Editorial) 061-456 8500 (Advertising) 061-480 0171 (Subscriptions)
Telex: 265871 MONREF G Quoting Ref. 79:MAG001 Telecom Gold Mailbox: 79:MAG001 Prestel Mailbox: 614568383
**Published by Database Publications Ltd, Europa House, 68 Chester Road, Hazel Grove, Stockport SK7 5NY.**

News trade distribution: Europress Sales and Distribution Limited, 11 Brighton Road, Crawley, West Sussex RH10 6AF. Circulation 0293 27053.

*Apple and the Apple symbol are the registered trade marks of Apple Computer Inc. Apple User is an independent publication and Apple Computer is not responsible for any of the articles in this magazine, nor for any of the opinions expressed.*

Writing for Apple User: Articles and programs relating to the Apple are welcome. Articles should preferably be typed or computer-printed, using double spacing. Unsolicited manuscripts, discs etc, should be accompanied by a self addressed stamped envelope, otherwise their return cannot be guaranteed. Unless agreed, material is accepted on an all rights basis.

While every care is taken, the publishers cannot be held legally responsible for any errors in articles or listings.

ABC 12,850 – JULY -DECEMBER 1984

Subscription rates for 12 issues, post free:

£12 UK
£13 Eire (IR £16)
£20 Europe
£15 Rest of world (surface)
£30 Rest of world (airmail)

# Deal signed with China

APPLE has signed a deal allowing it to sell micros and peripherals to China. Distribution will be through ACI Kaihin, a company closely aligned with the Chinese government.

Under the agreement ACI Kaihin also becomes a certified software developer, a move designed to spur the development of Chinese applications software and the localisation of existing software.

In addition, the agreement establishes a working group comprising representatives of Apple, ACI Kaihin, and the China National Education Equipment Corporation of the ministry of education to develop ideas and concepts for the use of micros in China's schools and colleges.

The deal coincides with China's recent outline of a four part high technology strategy for the country's modernisation. This covers integrated circuit development, the use of mainframe, mini and micro computers, increased communication and software development.

# The bank's selection

THE prestigious Bank of America has added the Macintosh to its list of standard equipment available for its personnel.

The decision clears the way for as many as 1,000 Macintoshes to be installed by the bank within the next 18 months.

Bank of America vice-president Max Hopper said: "We anticipate the Macintosh will help a growing number of our employees use personal computer technology to improve and expand job performance".

# Asian pirates clone Apple IIe

**From JOHN MacGIBBON**
*in New Zealand*

**ASIAN hardware pirates have finally cracked Apple's major line of defence for the Apple II series – they have cloned the IIe model. Previous Apple II clones originating from Asia have been based on the older II+ series.**

Rumours of the breakthrough have been circulating for a few months, and the development is confirmed in an unsolicited circular received in New Zealand from Bolar Enterprise of Taiwan.

In typical Asian business language, the circular announces: "Now we start to offer the Apple IIe for exporting now".

The address is just a room in a Chang Chung Road building, and the supplier is undoubtedly one of a number of backroom companies hawking the product.

FOB prices for individual sales of 64k computers with an 80 column card range from US$220 for a one-piece model, to US$250 for a model with a separate keyboard. Quantity discounts are offered, and there is also a "main board" model selling for US$160.

Similar offers by other Taiwanese exporters are doing the rounds in New Zealand, and are sure to attract interest, given the sky-high prices of genuine Apples in this country.

One Taiwanese company is represented by a local agent who passes on customers' orders and money. The goods are mailed direct to customers, who are responsible for clearing them through customs themselves.

The Kiwi agent loads the price by 25 per cent to cover his time and effort and to finance a guarantee.

Disc drives and a monitor have to be bought separately, but there is no shortage of perfectly adequate Asian made equipment available locally. Japanese Apple-compatible disc drives can be had for about NZ$400, while monitors sell for about NZ$300.

Then there is the competition – real Apples. In New Zealand, a genuine 64k/80 column Apple IIe, without disc drive or monitor, retails for over NZ$3000, eight times the Taiwanese FOB level.

However, the more usual way to buy a 64k Apple is bundled with one drive, 80 column card and a monitor, for $4055 – which is still a hefty price.

Apple will presumably be considerably miffed about the Taiwanese "IIe", for one reason why Apple developed the IIe, which contains a number of custom chips, was to foil hardware pirates.

The strategy worked for two years, and gave valuable breathing space while the company worked through the failure of their Apple III and Lisa range, and established the Macintosh.

The legal situation for Apple copies is uncertain in New Zealand, as there has never been a definitive court ruling on the copyright issue.

Local Apple agents CED Distributors have twice taken importers of "copies" to court. However in both cases financial hardship prompted defendants to settle out of court.

# Cider makers prefer Apples

WHEN it decided to set up a local area network in the offices of its pectin division, cider manufacturer H.P. Bulmer appropriately chose Apple computers.

Two years later the company decided that another network was needed at its factory some distance away. This was to be linked via modems to the existing system.

Instead of installing a second network, however, the firm upgraded the existing one, enabling it to hardwire links from the 56 mbyte central file server to all the required terminals in the factory.

The network, run on Nestar Systems Plan 3000, currently has 12 Apple II and IIe computers on it, sharing the resources of one central printer. Certain terminals, such as stock control and engineering, have printers connected directly to their Apples.

The key usage of the network is in sharing data among users and passing it from one program to another.

Information from a database concerning production batches is automatically reformatted and used as input for financial modelling functions.

By extending the network to the factory, the management can now obtain up-to-the-minute information about production, stock levels and other matters.

Says Lawrence Lloyd, pectin commercial manager: "We have designed several front-end menus which lead the user simply, clearly and logically through the programs, and most employees only require a couple of days to become familiar with the system.

"We find the electronic mail facility particularly useful. You don't have to wait for a memo to be delivered to get a reply, and being password-protected it only reaches the person it is intended for.

"Because of this high level of security for all data on the network we are currently investigating the possibility of installing a similar network in a joint venture with a Brazilian pectin manufacturer.

❯

*Symbiotic was on hand once again to register visitors to the Apple exhibition – with a little help from its advanced networking system*

# Euro showcase lives up to its name

**WITH more than 5,000 businessmen and educationalists visiting the show over the three days, Apple '85 once again turned out to be an up-market event catering for the discriminating buyer.**

All that was missing from the Champagne Suite at the Novotel was the popping of corks out of expensive bottles.

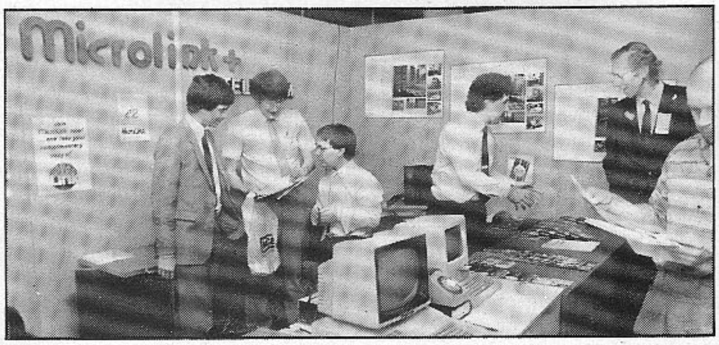
Even one usually dour Apple man was moved to describe the show as "a sparkling affair".

Billed as the European Showcase for Apple, the fourth annual exhibition more than lived up to its name as companies vied with one another to unveil the latest crop of technological advancements.

As Microscope described the event: "The fruits of those working to enhance Apple's products were clearly on show".

Apple itself unveiled the LaserWriter printer and the AppleTalk network for the first time – and there were even demonstrations of the much-delayed Jazz on a giant Macintosh.

MacCharlie, a product that surrounds a Macintosh and lets users run IBM PC software on its screen, arrived accompanied by kilted pipers.



*Electronic mail created a lot of interest*

Distributors Technology Advancement of Reading are offering two versions, one costing £1,650, the other £2,695. However they are likely to be in short supply up until the end of the year.

Thunderware's Thunderscan digitiser was unveiled on the P&P Micro stand. Its sensor replaces the Imagewriter's ribbon and allows it to scan a document and read the image into a screen window. The final picture can be manipulated with MacPaint. Price is around £230.

Pace Micro Technology launched four new communications products for the Apple including the long awaited auto dial board for the Nightingale modem at £49.

The Psion stand was showing its three dimensional chess program running on the Macintosh for £49.95. Originally written for the Sinclair QL, it won the world micro chess championships in 1984. Thorn EMI Computer Software previewed Apple II versions of its Perfect range of business software running on CP/M and MS-DOS systems.

Systematics International chose the show to launch a database for the Macintosh which will sell for around £250. Forming part of the MacBusiness range, it is written in Lisa Pascal and runs under the Mac operating system.

However it was Blyth Software who took over centre stage with Omnis 3, the first package to support networking on the Macintosh.

Priced at £445 it was officially endorsed by Apple UK boss David Hancock as "the cornerstone of future strategy".

He went so far as to suggest that the power of Omnis 3 – believed to be 12 months ahead of any similar development – would be important to Apple world-wide.

*Jazz on a giant Macintosh*

## MicroLink on show

**MICROLINK, a revolutionary new Telecom Gold service for Apple users, was up and running for the first time at Apple '85.**

**Subscribers to MicroLink pay a one-off registration fee of £5, plus a standing charge of £3 a month. This is compared to Telecom Gold's normal £40 registration and £10 a month minimum charge.**

**With MicroLink, almost everyone will be able to access Britain's national database via PSS for the cost of a local call.**

**It also provides subscribers with a personal electronic mailbox, giant bulletin board without queueing, free telesoftware and free access to MicroSearch, a revolutionary product locater.**

**And – for a bonus – there is a regularly updated news service covering all that is happening on the Apple scene world-wide.**

# Mac net is run by a ll

A SCOTTISH Apple dealer has developed a low-cost method of networking a group of Macintoshes – using a tried and trusty Apple II as the network controller.

The system has been developed by McQueen Systems of Edinburgh, main Apple dealers, and is the brainchild of the company's resident communications expert, Bob Carter, who has written the controlling software for the Apple II.

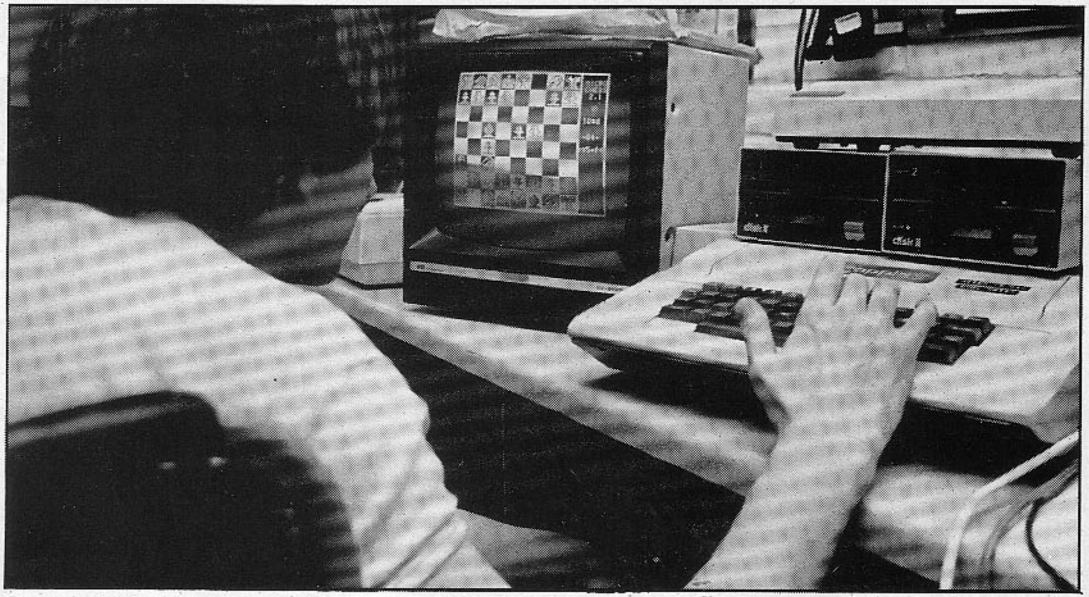
It is based loosely on the well-tested BBS bulletin board system, and is used by McQueen's staff to send internal electronic mail. It can also be accessed by staff who take their Macs on the road with them by dialling up on an outside line to the main switchboard, with one of the extensions patched into the Apple II.

Among its facilities is the ability to redirect mail already sent through the network. For example, a staff member who comes across a piece of information may first want to send it to Bob or one of the other senior McQueen's staff, who can then redirect it for the attention of everyone to whom it might be relevant.

*Apple User* readers who are already familiar with Telecom Gold or the BBS bulletin boards will recognise many of the features – mail storage, the ability to scan mail headings before reading messages in detail, and so on.

Although it is now running quite efficiently and already in daily use, Bob – a former hardware man who has developed a high degree of comms expertise, particularly on Apples – is still adding new features.

The Macs use the standard MacTerminal comms package, and are linked into the network by inexpensive Buzzbox modems running at 300 baud.



*Chess comes easier on an Apple*

# Apples help in hunt for jobs

**SEVERELY handicapped and disabled residents of Cheshire Homes are using Apple computers to entertain, educate and train themselves for jobs.**

With backing from the Manpower Services Commission, Robin Nixon and Steve Ludlow have set up extensive computer facilities at two Cheshire Homes – Seven Springs and Heatherley – using Apples.

To "interface" a resident to a micro may need individual input controls, and these are produced in workshops at each Home.

The computer software may also need to be modified, and this too is done on the premises, with the new versions being made available to other Cheshire Homes.

Experience to date has been highly encouraging, say those involved in the scheme.

Computers have helped residents with poor control to write perfect letters and produce geometric computer graphics.

A programme called Compaid – Computer Aid for Speech Impaired and Disabled People – was started by Lorna Ridgway, then chairman of management at Seven Springs early in 1982.

Robin Nixon told *Apple User:* "We were mainly concerned with solving the communication difficulties of some of our more severely disabled residents using custom input devices and software.

"To get the project under way we converted an old storage room into a computer room and took on eight previously unemployed trainees on a part-time basis under the Opportunities for Volunteering scheme.

"Under the supervision of Stuart McKears, the computer tutor, the trainees' tasks were to learn the basics of programming and computer use and pass these on to the residents.

"At the time there was only one readily available program for the Apple II – Mac Apple, a very flexible mini word processor and design package which could be operated by up to eight switches using a scanning matrix display.

"Mac Apple was written by Pat Poon of Kings College London and is still one of our most used pieces of software.

"For the last 18 months Compaid has been funded by the Manpower Services Commission community programme. We have four full time computer trainees, eight part time trainees and three workshop trainees.

"We have now reached the stage where we are able to conduct assessments of disabled people for computer use, and produce a newsletter".

# Briefcase Apple

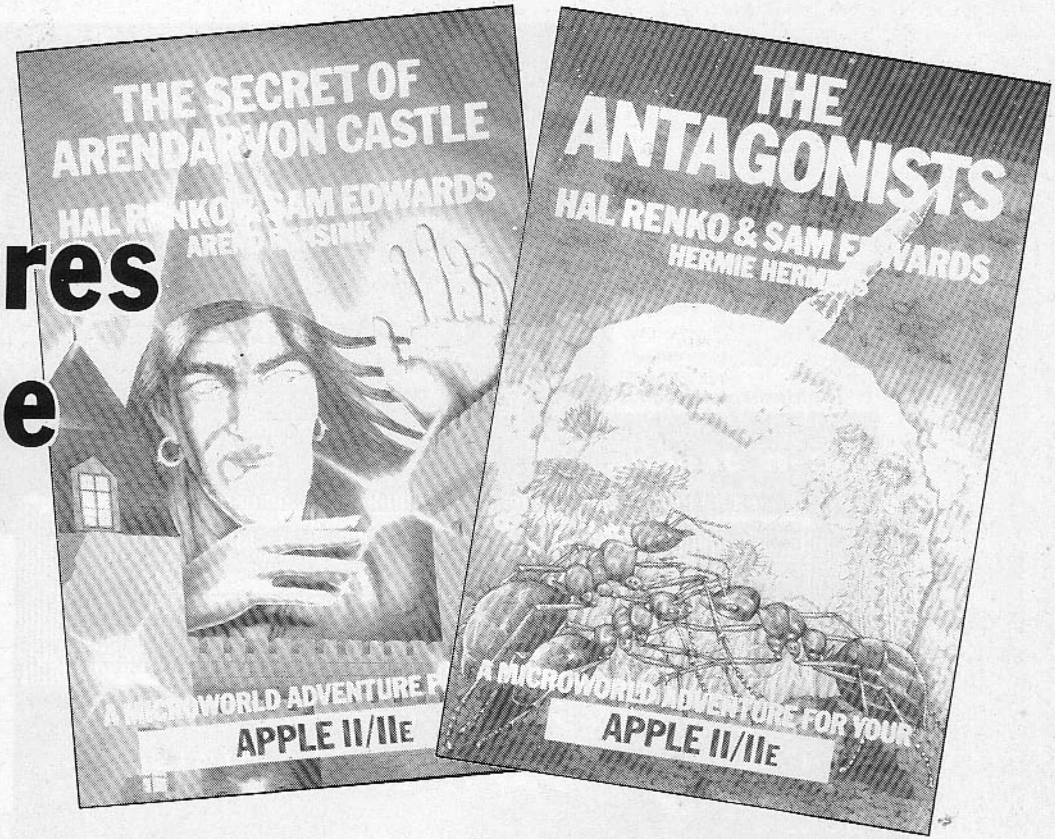
SYSTEMS house HG Computer Services has brought out the Apple-in-a-briefcase, complete with communications capability, for £1,875.

The system includes an Apple IIc, LCD monitor, rechargeable 12v battery pack and acoustic coupler – all contained in a specially designed, padded briefcase.

The user can run all existing programs for the IIc and also send and receive information wherever there is a phone.

by CLIFF McKNIGHT

# Adventures can come a lot cheaper



# ... but your fingers pay for the privilege

LET'S face it, the Apple isn't what you'd call a cost-effective games machine. With adventure games costing anything over £25, you could soon run up a hefty overdraft if you weren't careful.

Fortunately for the intrepid, but cost-conscious, adventurer, there are quite a few games around now that will bend the brain without straining the purse. With July being traditionally a lean month – every penny going towards the holiday – I thought I'd highlight a few of the bargains to be had.

Equal first in the frugality stakes must go to The Secret of Arendarvon Castle and The Antagonists from Addison-Wesley. You can be playing either of these adventures for a mere £5.95 – if you don't mind doing a bit of typing.

Both are published as paperback books. The first 60-odd pages set the scene while the other half consists of information on how to play the game, plus a complete listing for you to type in.

As you can imagine, it's no joke typing in a complete text adventure. However, the task is broken down into seven sections, the suggestion being that you do one section each day for a week. This is quite a good idea, although obviously you don't have to stick to the scheme.

The programs are coded so that you won't spoil the game by typing it in. For example, most of the typing involves data statements containing such chunks as 6R49 5SKN 83T9 and so on. Each line contains 12 chunks of four characters, and a typical night's typing might involve 90 such lines. It's the sort of task you'd have to split into sections in order to minimise the risk of errors, despite the presence of error-checking routines in the programs.

If you've got some patience and a few empty evenings, either of these books will provide you with a reasonably challenging adventure. If you don't fancy typing the games you can buy a disc from Addison-Wesley at an additional cost of £6.85 each. This way you pay a total of £12.80 for each adventure and save yourself the typing job.

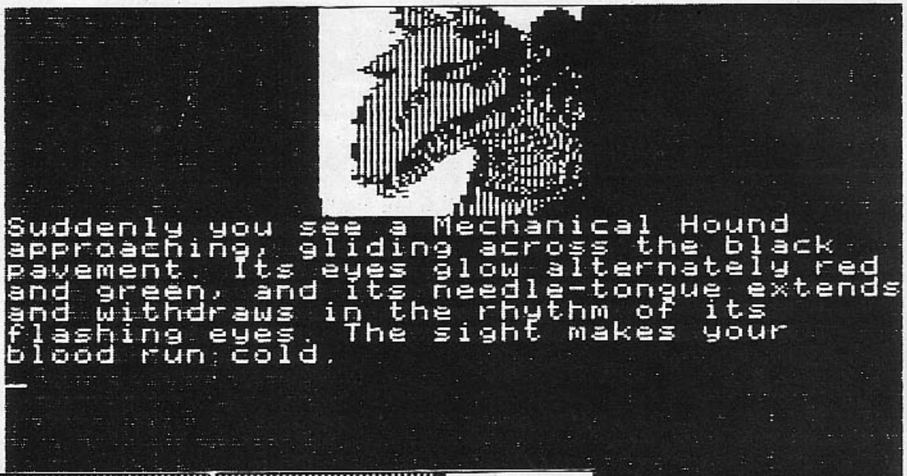
The Antagonists is a tale of the future when insects dominate the world. To help in completing the adventure the book provides such items as the diary entries of the last human survivor, a book of flowers, and a brochure from the Insectoland fun-park.

Arendarvon Castle has you following in the footsteps of a journalist who has mysteriously disappeared. The book consists mostly of the castle guide book, with a few notes from your predecessor.

The books are well-produced and make far better manuals than any I've seen accompanying adventure games. There's an evening's reading in each of them, and you'd need to read them before playing the games. My own preference is for The Antagonists, being a sci-fi rather than a detective fan. However, they're both excellent value and worthy of investigation.

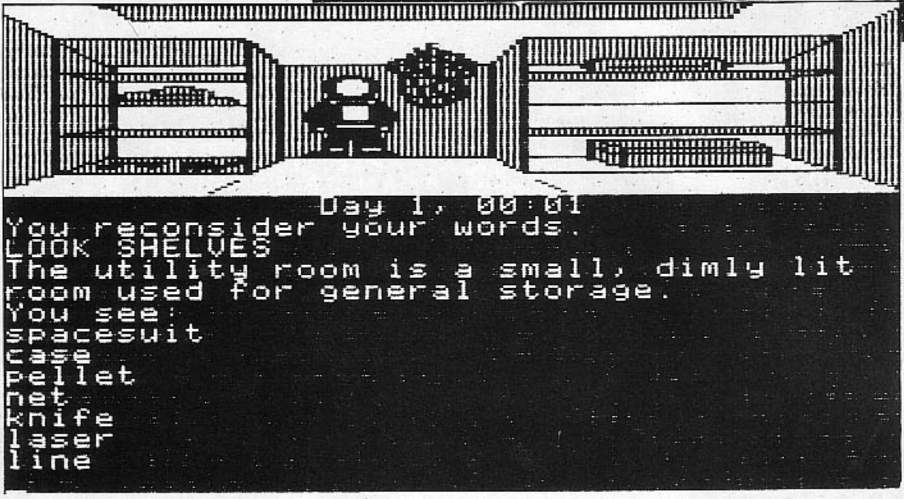
*Title: The Secret of Arendarvon Castle.*
*Authors: Hal Renko, Sam Edwards & Arend Rensink.*
*Publisher: Addison-Wesley.*
*Requirements: 48k.*

*Title: The Antagonists.*
*Authors: Hal Renko, Sam Edwards & Hermie Hermens.*
*Publisher: Addison-Wesley.*
*Requirements: 48k.*



```
Suddenly you see a Mechanical Hound
approaching, gliding across the black
pavement. Its eyes glow alternately red
and green, and its needle-tongue extends
and withdraws in the rhythm of its
flashing eyes. The sight makes your
blood run cold.
_
```

*The Mechanical Hounds pursue Montag in Fahrenheit 451*



```
             Day 1, 00:01
You reconsider your words.
LOOK SHELVES
The utility room is a small, dimly lit
room used for general storage.
You see:
spacesuit
case
pellet
net
knife
laser
line
```

*You may not Rendezvous with Rama if you choose the wrong objects*

# Spinning webs to good effect

LAST January I mentioned a series of adventures based on books. These, I said, would appear under the Trillium name. Wrong again! Apparently, some copyright difficulties caused a name-change to Telarium, which in my opinion is a better name anyway. It presumably derives from words like telary meaning web-spinning – an appropriate root for a series of adventure games, don't you think?

The games are well-packaged in glossy, fold-out packs and take the form of hi-res games. Each occupies two double-sided floppies which gives you some idea of the size. However they retail for around £20, which is far less than you'd expect for products such as these.

Fahrenheit 451 is based on Ray Bradbury's classic novel and has been produced in collaboration with him. He recommends it for "anybody curious about what happens to Montag after the book ends" which tells you how it stands in relation to the book. Personally, I find this a better approach to a book-based game than simply working around the story as written.

Rendezvous With Rama is based on Arthur C. Clarke's science fiction novel about the exploration of an alien "spaceship". In many ways it's a surprising choice for a game because the book was more concerned with raising questions rather than answering them. Arthur C. Clarke has written a secret new ending for the game, so if you found the original ending unsatisfactory you might prefer this one.

Incidentally, the review copy contained at least one fatal bug. When you're in the crew room examine the locker and then GET UTILITY SUIT. If you have a duff copy of the game, you'll get BAD OBJ and it will hang. If this happens to you return the game from whence it came and insist on a new copy.

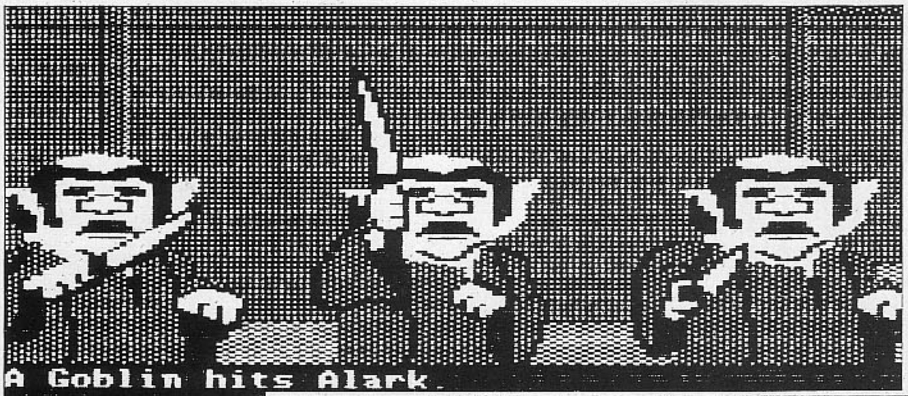
The other two games I've seen from this series present a slightly different approach to the relationship between book and adventure game.

Amazon isn't based on a book but is written by Michael Crichton, author of "Congo", "The Andromeda Strain" and others. It's a funny mix of text and graphics, not quite like the usual hi-res adventure. Also it had a distinct "linear" feel for most of it. However it has an excellent Save Game facility which allows you to name each saved position and presents a menu of them for restoring purposes.

Like Rendezvous, it has some arcade-type action and interesting sound effects. There's also a parrot called Paco to keep you company, and some hints which look as though they've been generated on a Mac. Unfortunately, they don't seem to relate to the game, so they weren't much help.

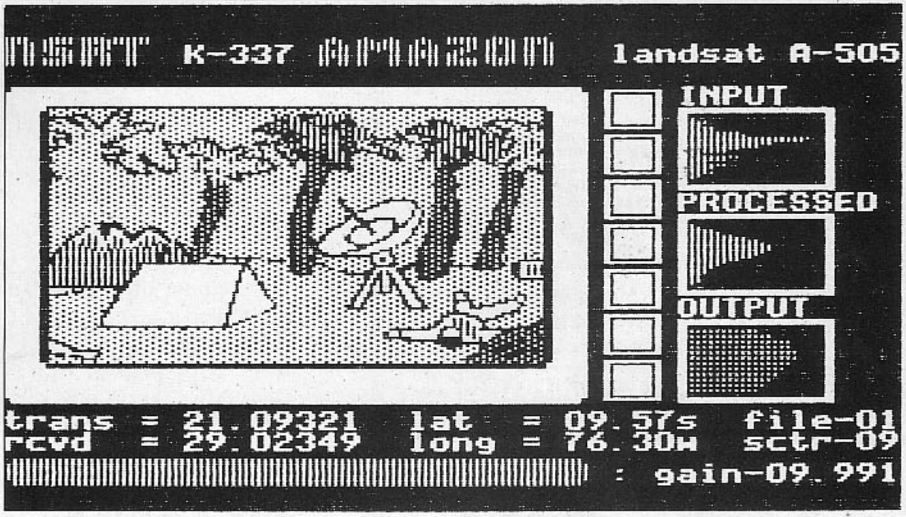
Shadowkeep turns the relationship between game and book completely around. In this case the game came first and inspired a novel by Alan Dean Foster, author of "Dark Star" among other things. It's a fantasy role-playing game rather than an adventure and if it looks like being successful I'm going to buy shares in a disc-making company.

In order to play it, you must make playable copies of the four



A Goblin hits Alark.

*Goblins are only one of the perils in Shadowkeep*



```
trans = 21.09321   lat  = 09.57s   file-01
rcvd  = 29.02349   long = 76.30w   sctr-09
                                 : gain-09.991
```

*The remains of the previous party to visit Amazon*

game sides and you'll also need a save game disc. That's half a box of discs gone straight away!

It's the usual style of characters with different qualities, buying weapons and armour, casting spells and so forth. A party can have up to nine characters so you can assemble a really motley crew. The game is nicely implemented and the manual is one of the better ones I've seen for this type of game.

Finally this month I'd like to draw your attention to Adventure Writer. If you're fed up playing other people's adventures and fancy writing your own, this is the package for you.

The packing proclaims: "Your own adventure program – the first time you try" and I must admit this is true. However, if it gives the impression that it's easy to write an adventure game you'll have a surprise in store.

Adventure Writer certainly will enable you to write a properly constructed text-only adventure game. It's an implementation of Gilsoft's "The Quill", with which several commercially produced games have already been written for other micros.

However, like most things in life, you get out what you put in. To produce a full-length game, you need to do a lot of work on paper before you go near the disc, and you'll need to have read the 100-plus pages of the manual too. There's a sample adventure to type in which provides quite a detailed tutorial on the package's features.

Essentially, Adventure Writer does all the "house-keeping" for you and provides a command structure in which to write your adventure. Once you've learned this structure, programming your game goes considerably faster than it would in an ordinary high-level language.

If structure gives the impression of stricture, it's a false one. The program doesn't care whether an object is a magic sword or a pair of false teeth, it will keep track of it. Hence the final game will be very much your own making.

Not only does this package give you the means of writing your own adventures, it makes you appreciate what has gone into adventure games you've already played. Denise and I had been writing an adventure game for a couple of years and got further into it with Adventure Writer in a couple of weeks than we had previously. Of course our original had been a hi-res graphics adventure and it took a while to draw the pictures . . . well, that's our excuse.

Once you've written and tested your game, it can be turned into a game proper, which doesn't need the Adventure Writer disc to be booted. You're free to sell your games without any kind of licensing agreement, but the game must include a statement that it was written with Adventure Writer.

Your game can have up to 252 locations and 255 objects. You can also have 255 messages – like "What do you want to do with the false teeth?" – in addition to the usual location descriptions, and a vocabulary of 254 words. We're half-way through our adventure and are nowhere near half-way through the capacity of the system.

Your game can even have the usual Save/Restore facility, and there are various sound commands to add an extra dimension to the text. At around £25 it will cost you the same as you might pay for someone else's game, but just think how many you could write. Who knows, you might even sell a few.

As someone who's played many adventure games and tried to write more than one, I just wish I'd had Adventure Writer several years ago.

*Title: Fahrenheit 451.*
*Author: Ray Bradbury & others.*
*Publisher: Telarium.*
*Requirements: 64k.*

*Title: Rendezvous With Rama.*
*Author: Arthur C. Clarke & others.*
*Publisher: Telarium.*
*Requirements: 64k.*

*Title: Amazon.*
*Authors: Michael Crichton, David Durand & Stephen Warady.*
*Publisher: Telarium.*
*Requirements: 64k.*

*Title: Shadowkeep.*
*Author: Not credited.*
*Publisher: Telarium.*
*Requirements: 64k and 5 blank discs.*

*Title: Adventure Writer.*
*Author: Graeme Yeandle.*
*Publisher: Code Writer.*
*Requirements: None*

# The Apple in respiratory medicine

THE East Anglian Regional Pulmonary Function Laboratory at Papworth Hospital, Cambridge, has made extensive use of Apple II micros since early in 1978. The first system based on a 110V Apple II was purchased for a specific data-acquisition and analysis application in an associate laboratory at Addenbrookes Hospital, Cambridge.

The experience gained with this system showed it to be entirely suitable for the application and a second system was acquired in 1979 for use in the Papworth laboratory. Since then a further seven systems have been purchased for use at Papworth, Addenbrookes and Newmarket laboratories.

At Papworth in particular, the Apples are the workhorses in the laboratory and are used routinely throughout each working day.

By DOUG SHAW and TIM HIGENBOTTAM

Their reliability has been outstanding. Apart from minor problems caused by disc-drive head misalignment after extensive use, the nine computers themselves have suffered only two breakdowns in almost seven years – both caused by failure of the power supply.

Within the laboratory group, the policy is to standardise as far as possible on the selection of peripherals.

Where printing speed and/or graphics printing capabilities are required, the choice has been the Anadex 9500 series (latterly the Superscribe) printer.

Qume daisywheel printers are employed for word-processing applications where high-quality presentation is of importance.

Three systems are equipped with Eicon 8in 2mbyte disc drives which operate in conjunction with either one, or two Apple 5¼in disc drives.

Other peripherals include two Apple graphics tablets and a selection of peripheral cards including, for 80 column work, the Videx Videoterm and the Microsoft Z80 co-processor.

The CCS asynchronous serial interface provides printer drive on all but two systems which are equipped with the Apple parallel interface.

Four systems are equipped with 16k memory extension cards and a Mountain Hardware ROM burner is available for use on all systems for dedicated firmware applications.

Data-acquisition and control applications were catered for initially by in-house designed cards produced at Addenbrookes hospital instrument workshops.

A number of applications make use of the AI-02 8 bit ADC card by Interactive Structures, while applications requiring greater precision utilise the 12 bit U-A/D card supplied by U-Microcomputers of Warrington.

Applications requiring digital to analogue conversion are catered for by the 12 bit, two channel DAC card supplied by MC Computers of Newbury.

With the exception of word processing software and one or two relatively minor applications programs, all software used within the group is produced in-house.

In the early days this was a result of much of the commercially available software tending to be too restricted in application.

This situation is undergoing rapid change, however, and as greater flexibility and sophistication are introduced into commercial software so is it more likely to be taken into use.

For example, one data handling system at Papworth is soon to be upgraded with the addition of an Eicon Discache and the data files transferred to run under a single user Dataflex DBMS.

The majority of applications make use of Basic or Assembler or a combination of the two running under DOS 3.3.

The acquisition of the ProDOS development kit during 1984 renders it likely that, subject to a satisfactory evaluation, most applications will be transferred to run under ProDOS, particularly if, and when, this system is made available on hard disc alternatives to Profile.

Some CP/M based applications are available, notably Wordstar and Fortran.

The Pulmonary Function Laboratory at Papworth handles about 1,200 cases a year, Addenbrookes some 1,600 and Newmarket 500, involving between 8,250-13,200 investigatory procedures.

In the past this necessarily gave rise to a considerable amount of clerical work by both technical and secretarial staff.

An early priority was therefore given to the provision of a database for handling clinical test results in order to minimise time spent on "unproductive" activities.

The present system, which has been evolving over the past four years, has taken over all routine laboratory calculations and enables a physician to interpret and report on investigations and provides an automatic report printing facility.

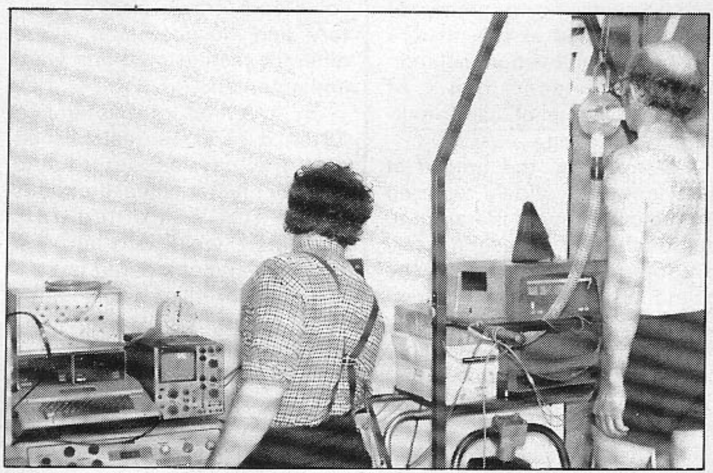
Concurrent statistical files giving a breakdowr and distribution of laboratory work-load are maintained by this system.

Since the same system is operating within the pulmonary function laboratories at Addenbrookes and Newmarket Hospitals, a wealth of information is accumulating for Regional epidemiological investigations.

While data handling is undoubtedly an important and cost-effective function it makes little use of one of the Apple II's outstanding features – the ease with which the computer can be interfaced to a wide range of laboratory instruments.

We have exploited this feature to advantage and on-line data acquisition is used extensively during clinical investigations at Papworth.

An overall assessment of a



*A subject undergoing a fitness assessment in the exercise testing laboratory.*

** Doug Shaw is Senior Medical Physics Technician, Department of Respiratory Physiology, Papworth Hospital.*

*Tim Higenbottam is Consultant Physician and Clinical Physiologist at Papworth and Addenbrookes Hospital, Cambridge.*

patient's cardio-pulmonary system can be made by means of an exercise test where the patient either walks on a treadmill or rides an exercise bicycle fitted with a dynamometer brake.

During a period of graded exercise, or work-load, the expired oxygen and carbon dioxide are measured by mass spectrometer. These values, together with the heart-rate and the minute-by-minute volume of breath, are analysed by the computer and presented graphically in comparison with the accepted normal ranges for heart-rate and ventilation.

Figure I shows how the graphical presentation clearly differentiates between the results of a test carried out on a normal middle-aged man and a young man in the early stages of an intractable pulmonary condition.

During the summer of 1981, and in collaboration with research associates at the Catholic University Medical Centre in Rome, development was undertaken into a unique method of measuring respiratory drive.

This deceptively simple terminology describes a complex system of neuro-receptors which, in association with the brain, serves to regulate the rate and depth of respiration appropriate to the level of activity.

Respiratory drive is largely conditioned by the partial pressures of oxygen and carbon dioxide in the circulating blood, and our test method makes use of this fact by changing the partial pressures with a simple re-breathing technique.

The test has been in routine clinical use since 1981 to investigate a range of pulmonary and neural disorders.

More recently the technique has been effectively used in patients to be assessed for, and subsequent to, heart-lung transplantation.

The computer scores heavily over traditional methods in this type of investigation, since, with the aid or a mass spectrometer to deal with the gas analysis, it can accurately measure a large number of variables during the period of each breath.

Since there is an average number of 40 breaths during a test, the results of post-test analysis represent a statistically powerful means of assessing respiratory drive.

A virtually identical technique was recently reported to have been adopted in one of the principal respiratory laboratories in the United States.

In the spring of 1984, and again in collaboration with our Italian colleagues, we investigated the relationship between the ventilation of the air spaces in the lung and the perfusion of the pulmonary capillaries.

It has long been known that the ventilation/perfusion ratio (V/Q) could be significantly altered by certain disease processes, but the methods of making diagnostic measurements on the basis of V/Q were very time-consuming, difficult of interpretation, and thus rarely used.

The purpose of the investigation was to ascertain whether the combination of computer and mass spectrometer could be used to speed up the investigatory technique and therefore enable changes in V/Q to be used as a diagnostic aid for the physician.

The idea was to make use of a mixture of trace concentrations of a number of gases having differing solubilities.

The mixture was to be inhaled by the subject and rate of disappearance of the gases measured when the subject was breathing air.

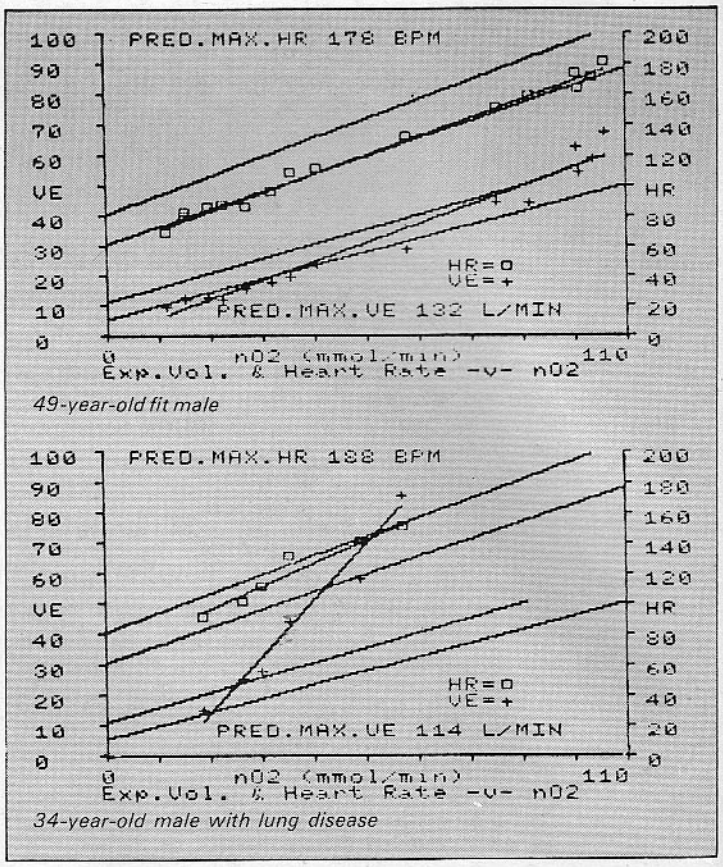
The relationship between individual pairs of disappearance curves were then to be analysed by the Simplex algorithm of Linear Programming to give the V/Q relationship for a compartmentalised model of the lung.

It was well known that two of the gases used in the mixture had been employed in isolation to make measurements of specific aspects of cardio-pulmonary function.

For example, the uptake of acetylene was widely reported in medical research literature to be suitable for measuring cardiac output as an index of the pulmonary capillary blood flow.

Similarly, a routine measurement in the majority of pulmonary function laboratories is that of the transfer factor for carbon monoxide.

This is, in effect, a measurement of the efficiency of the lung in transferring gases between the air spaces and the blood.



49-year-old fit male

34-year-old male with lung disease

*Figure I: Graphic displays of exercise test results with overlays of normal value ranges*

It was reasoned therefore that, since transfer factor was known to be affected by pulmonary perfusion and current methods of measuring cardiac output involved a patient-invasive technique, there would be decided advantages in combining these two tests together.

Consequently there began a parallel investigation into the viability of this proposal.

With the aid of a number of volunteers drawn from laboratory and hospital staff, three different testing methods were implemented.

By the end of September, 1984, it had been established that the cardiac output and transfer factor measurements using the combined test correlated well between test methods and compared favourably with predicted values and traditional methods of testing.

Furthermore, the addition of a third, insoluble gas to the mixture as a reference enabled measurement of the distibution of ventilation to be made – while the relationships between the disappearance curves of carbon monoxide and acetylene could be used to make a determination of the lung tissue volume.

Thus it would appear that four common measurements made in cardio-pulmonary investigations could conceivably be replaced by a single, short duration.

While the fortuitous development of this test was not the principal aim of our Italian colleagues' visit, the potential advantages of lower patient risk and financial savings proved sufficient to justify further development.

On the basis of the encouraging intial results, the Chest Heart and Stroke Association have awarded a £37,000 research grant to the Papworth laboratory to fund the clinical validation of the technique.

The prototype apparatus on which the original investigations were carried out has been replaced by a sophisticated valve system operated directly by the computer.

And it is planned to extend computer control to cover tuning, calibration and mass-centre switching of the mass spectrometer during the two-year validation period.

# Structured files make for tidy records

## By STUART BELL

IN the last tutorial we looked at the way in which Apple Pascal allows us to use records in order to group together related information. This month we shall examine the use of files of records in order to handle large volumes of related information.

A structured file is a series of records, stored one after the other, on disc. The other common type of file used in Apple Pascal is a textfile. This is simply a series of characters, such as a Pascal program.

If these records are always examined in the same order, starting with the first, then the second, and so on, then this is said to be a sequential access file.

Standard Pascal allows only this type of file, but, as we shall see next month, Apple Pascal allows us to access any record in any order in the file – this is termed random access.

Before we use a file in a Pascal program, it must, like all variables, be declared. If we have a record of people's names and ages, and wish to have a file of these records, then we could use:

```
type      dets = record
                   name:string[20];
                    age:integer
                  end;
var  detsfile : file of dets;
```

This means that the file called detsfile consists of a series of records of the structure described in the declaration of dets.

Before a file can be used, it must first be opened like a conventional filing cabinet. Pascal provides us with two statements to do this.

The one to use depends on whether the file already exists, or is to be created. If we wish to use a new file, then we use the instruction Rewrite. If a file of the same name already exists, then it will be deleted. If we wish to use an existing file, then to open it we use Reset.

Both Reset and Rewrite statements are of the same form.

After the instruction comes two variables, enclosed within brackets. Technically, we are calling a procedure with two parameters.

The first variable is the internal name of the file – that is, the one used in the declaration at the top of the program. The second name is the external name – that is, the name of the file as it will appear in the disc directory.

For example, if we wanted to open the file declared above, which does not exist already, and which is to be called NAMES&AGES, then we would use the statement:

```
rewrite(detsfile,'NAMES&AGES');
```

If it existed already, then we would use Reset rather than Rewrite. The external name of the file may be held within a string which might, for example, be entered by the user. For example:

```
write('Please give name of file to be used: ');
readln(fname);
rewrite(detsfile,fname);
```

Before this, you could have used a version of last month's program to list for the user all the possible file names (the ones of type "data"). The variable fname would need to be declared as a string.

Before we consider how we read from and write to a file, let us note how the file is closed after use. The statement is Close, with the internal name of the file, followed by Lock. For example:

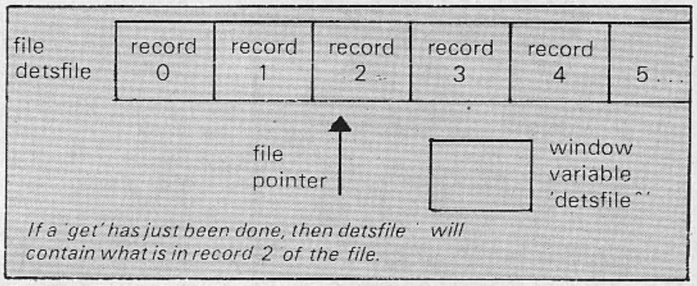
```
close(detsfile,lock);
```

The Lock ensures that the file is properly saved to disc. See page 28 of the Apple Pascal Language manual for alternatives to Lock. If you do not lock a new file, the file will not appear in the disc directory.

The easiest way of understanding how files or records work is to imagine a filing cabinet drawer full of records – each record can be pulled out, looked at, and then replaced.

Pascal uses a file pointer to point to the next record to be accessed. This is rather like a finger skipping along the top of the drawer of the cabinet.

When a file is opened, the file pointer is at the start of the file. When the first record has been dealt with, the file pointer automatically advances to the next record, and so on.



*If a 'get' has just been done, then detsfile ^ will contain what is in record 2 of the file.*

*Figure I: File pointer and window variable*

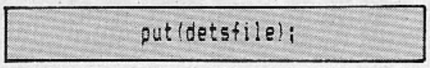
Associated with the file pointer is the window variable for a file. This window into the file is rather like a little desk, on to which each physical record from the cabinet can be put in order to look at it.

The name of the window variable is always the name of the file followed by the ^ symbol, for example, Detsfile^. The relationship of file, record, file pointer and window variable is shown in Figure I.

To write data to a file, we first put the information into the window variable. This can be done directly, for example:

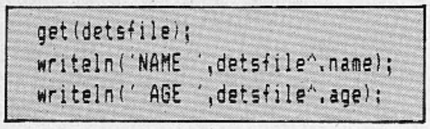
```
detsfile^.name:='Stuart Bell';
detsfile^.age:=30;
```

Alternatively, another record can be filled, and then the whole record copied into the window variable. To write the data to the file we simply use:



```
put(detsfile);
```

When a Put is executed, the file pointer is advanced to the next location in the file, and the contents of the window variable are written to disc. To read the information back, we use the Get statement. Here's an example:



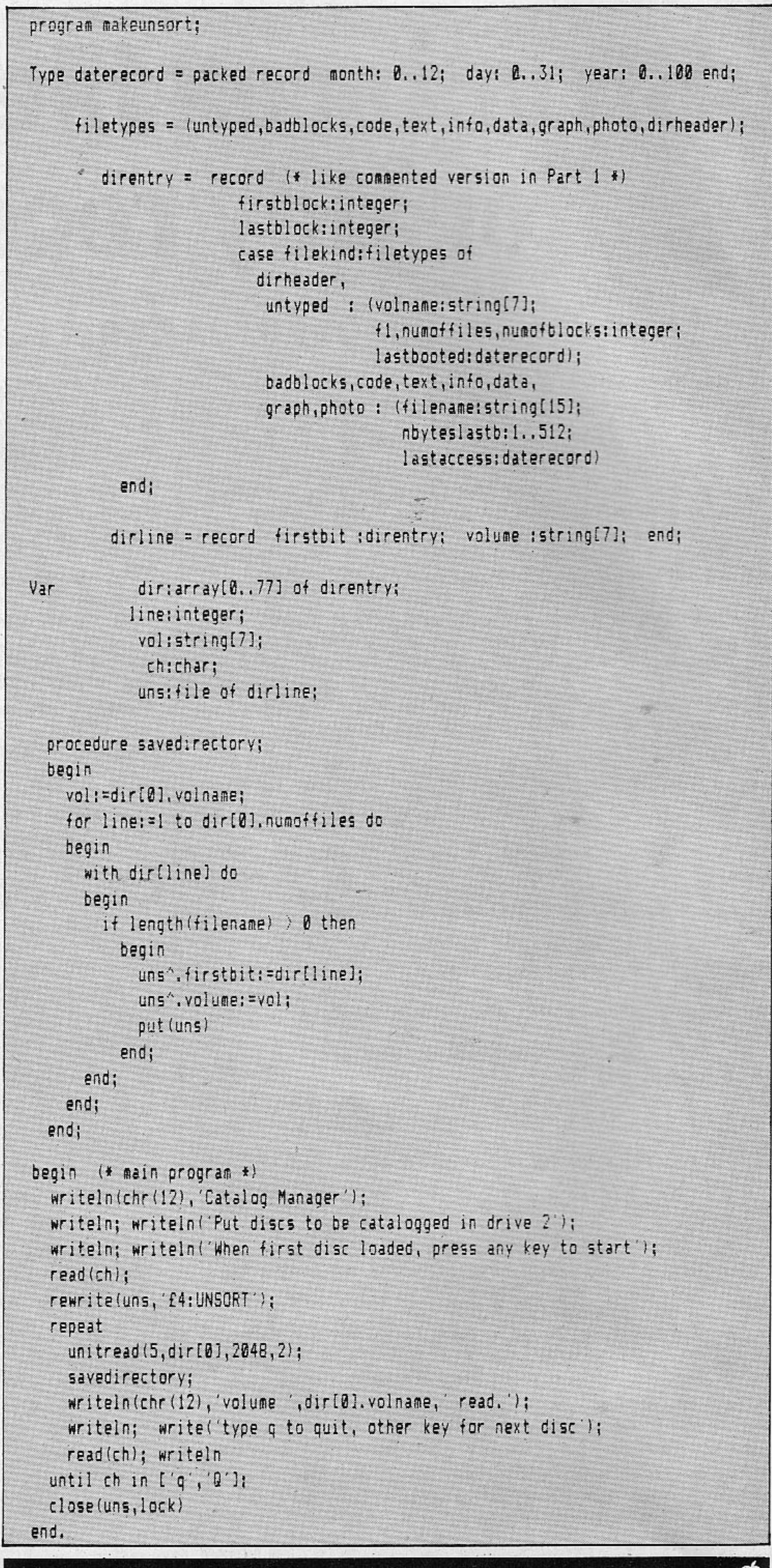
```
get(detsfile);
writeln('NAME ',detsfile^.name);
writeln(' AGE ',detsfile^.age);
```

Note that Get works just like Put, advancing the file pointer before transferring the data. Note that so that the first record of a file can be read properly, doing Reset of a file causes the first record to be read into the window variable, before a Get is done. Also, the first Put will always be to the first record of a file.

The program Makeunsort demonstrates most of the features of Pascal that we have looked at this month, but error-handling has been left out to simplify it. It uses the record declaration introduced last time. Its purpose is to create a file of all the directory entries on a series of discs, together with the name of the disc.

Next month, we shall see how we can use random accessing to sort the file into alphabetical order, hence creating a master list of every Apple Pascal file that you have.

I have about a thousand files on a hundred discs, and find such a cataloguing system invaluable, particularly when I can't find a file!



```
program makeunsort;

Type daterecord = packed record  month: 0..12;  day: 0..31;  year: 0..100 end;

     filetypes = (untyped,badblocks,code,text,info,data,graph,photo,dirheader);

       direntry =  record  (* like commented version in Part 1 *)
                      firstblock:integer;
                      lastblock:integer;
                      case filekind:filetypes of
                        dirheader,
                         untyped  : (volname:string[7];
                                    f1,numoffiles,numofblocks:integer;
                                    lastbooted:daterecord);
                         badblocks,code,text,info,data,
                         graph,photo : (filename:string[15];
                                       nbyteslastb:1..512;
                                       lastaccess:daterecord)
         end;

        dirline = record  firstbit :direntry;  volume :string[7];  end;

Var         dir:array[0..77] of direntry;
           line:integer;
            vol:string[7];
             ch:char;
            uns:file of dirline;

  procedure savedirectory;
  begin
    vol:=dir[0].volname;
    for line:=1 to dir[0].numoffiles do
    begin
      with dir[line] do
      begin
        if length(filename) > 0 then
          begin
            uns^.firstbit:=dir[line];
            uns^.volume:=vol;
            put(uns)
          end;
      end;
    end;
  end;

begin  (* main program *)
  writeln(chr(12),'Catalog Manager');
  writeln; writeln('Put discs to be catalogged in drive 2');
  writeln; writeln('When first disc loaded, press any key to start');
  read(ch);
  rewrite(uns,'£4:UNSORT');
  repeat
    unitread(5,dir[0],2048,2);
    savedirectory;
    writeln(chr(12),'volume ',dir[0].volname,' read.');
    writeln;  write('type q to quit, other key for next disc');
    read(ch); writeln
  until ch in ['q','Q'];
  close(uns,lock)
end.
```

appletip

# Free space in DOS 3.3

For those people who enjoy playing around with DOS 3.3 the following information may come in handy. Below are some locations of free space in DOS 3.3 (these assume a 48k/64k system). These areas are very useful for adding your own patches to DOS.

**$B6B3-$B6FD (46771-46845)=75 bytes**
**$B78D-$B792 (46989-46994)=6**
**$BA69-$BA95 (47721-47765)=45**
**$BCDF-$BCFF (48351-48383)=33**

This gives a total of 159 bytes available for patches.

**Jason W. Smith**

# Take a BLOAD off your mind

## Reassign the default load address of a binary file with this utility by LEE HAMMOND

THIS new command will allow the user to reassign the default load address of a binary file.

This saves the user from having to remember the alternate load address of a binary file during a BLOAD or BRUN command or the bother of BSAVE, BLOAD, BSAVE again.

When the command is installed, via either an Exec or Basic loader, the DOS CHAIN command will no-longer exist.

Since the command is loaded between HIMEM and DOS's buffers, it is *not* saved as part of it during INIT.

The formal syntax of the command is:

**BADDR <filename>, A[$]nnnnn[, V nnn, Sn,Dn]**

To create the required program files:

Key the machine code and save it:

**BSAVE BADDR.OBJ0, A$95B1,L$43**

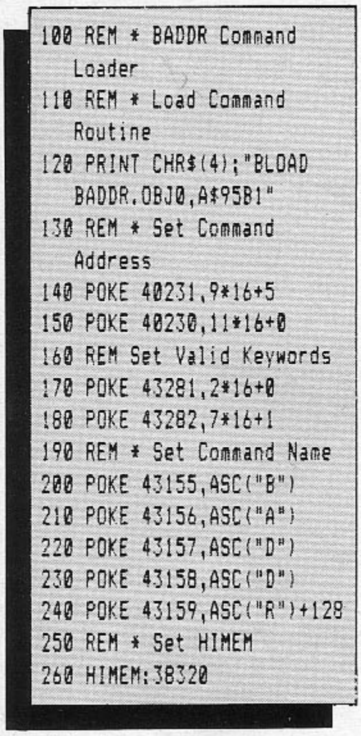
Key the Basic loader and save it.

To invoke the new command, run the Basic loader and then enter BADDR.

If required, the command name BADDR can be changed to one of your choice.

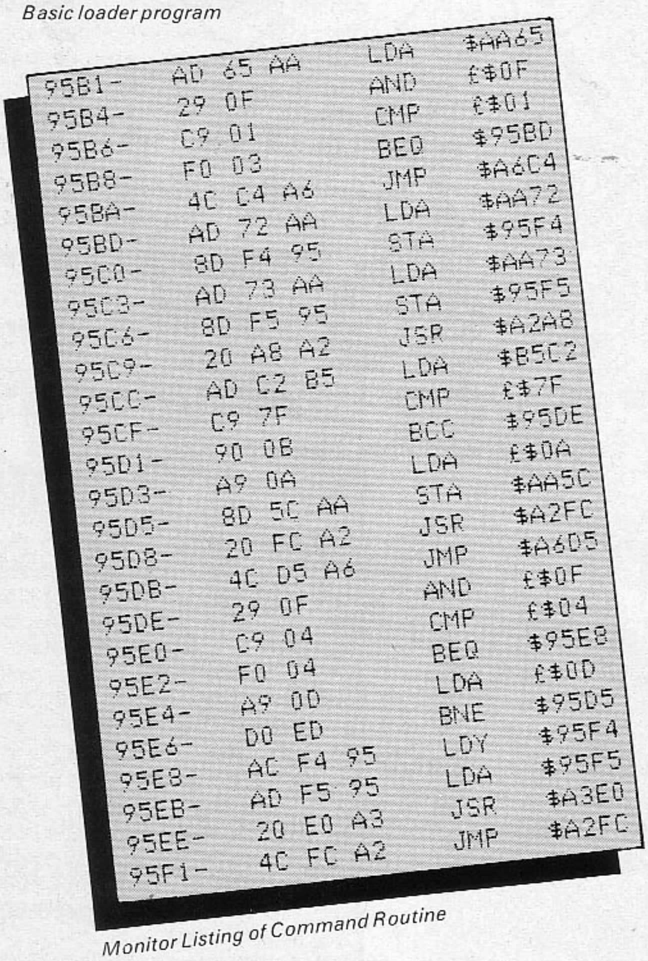
This is accomplished by changing lines 200 to 240 in the Basic loader.

Note: The command *must* be five characters long.



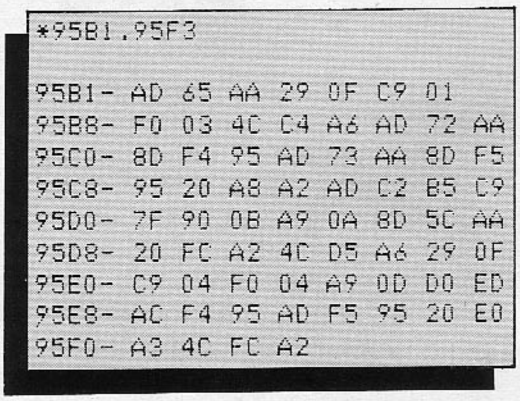
```
100 REM * BADDR Command
   Loader
110 REM * Load Command
   Routine
120 PRINT CHR$(4);"BLOAD
   BADDR.OBJ0,A$95B1"
130 REM * Set Command
   Address
140 POKE 40231,9*16+5
150 POKE 40230,11*16+0
160 REM Set Valid Keywords
170 POKE 43281,2*16+0
180 POKE 43282,7*16+1
190 REM * Set Command Name
200 POKE 43155,ASC("B")
210 POKE 43156,ASC("A")
220 POKE 43157,ASC("D")
230 POKE 43158,ASC("D")
240 POKE 43159,ASC("R")+128
250 REM * Set HIMEM
260 HIMEM:38320
```

*Basic loader program*



```
95B1-   AD 65 AA      LDA   $AA65
95B4-   29 0F         AND   £$0F
95B6-   C9 01         CMP   £$01
95B8-   F0 03         BEQ   $95BD
95BA-   4C C4 A6      JMP   $A6C4
95BD-   AD 72 AA      LDA   $AA72
95C0-   8D F4 95      STA   $95F4
95C3-   AD 73 AA      LDA   $AA73
95C6-   8D F5 95      STA   $95F5
95C9-   20 A8 A2      JSR   $A2A8
95CC-   AD C2 B5      LDA   $B5C2
95CF-   C9 7F         CMP   £$7F
95D1-   90 0B         BCC   $95DE
95D3-   A9 0A         LDA   £$0A
95D5-   8D 5C AA      STA   $AA5C
95D8-   20 FC A2      JSR   $A2FC
95DB-   4C D5 A6      JMP   $A6D5
95DE-   29 0F         AND   £$0F
95E0-   C9 04         CMP   £$04
95E2-   F0 04         BEQ   $95E8
95E4-   A9 0D         LDA   £$0D
95E6-   D0 ED         BNE   $95D5
95E8-   AC F4 95      LDY   $95F4
95EB-   AD F5 95      LDA   $95F5
95EE-   20 E0 A3      JSR   $A3E0
95F1-   4C FC A2      JMP   $A2FC
```

*Monitor Listing of Command Routine*



```
*95B1.95F3

95B1- AD 65 AA 29 0F C9 01
95B8- F0 03 4C C4 A6 AD 72 AA
95C0- 8D F4 95 AD 73 AA 8D F5
95C8- 95 20 A8 A2 AD C2 B5 C9
95D0- 7F 90 0B A9 0A 8D 5C AA
95D8- 20 FC A2 4C D5 A6 29 0F
95E0- C9 04 F0 04 A9 0D D0 ED
95E8- AC F4 95 AD F5 95 20 E0
95F0- A3 4C FC A2
```

*Hexadecimal dump of Command Routine*

# Excuse the interruption...

**BILL ALLEN takes an in-depth look at two sophisticated systems, Thirdware's FingerPrint printer interface card and Dark Star's Printerrupt program for their Snapshot card**

TO compare Thirdware's FingerPrint printer interface card with Dark Star Systems' Printerrupt program for their Snapshot interrupt-and-resume card is like trying to compare Bruce Lee and Muhammed Ali.

Both were noteworthy exponents at their chosen art, even though the way they carried it out was quite different.

However, for those Apple users who want sophisticated print facilities, especially graphics, the systems do have some comparable features, and are roughly in the same price range.

What they both have is the facility to carry out text and graphics dumps to a more sophisticated level than the established Grappler-style cards, and they both achieve this by using a pushbutton to interrupt the Apple at any stage during the execution of a program.

They also allow specialised user routines to be executed when the interrupt occurs.

We'll start off by looking at Thirdware's FingerPrint printer interface card. The first thing to note is that this is in no way connected with a product of a similar name that fits inside an Epson printer and is used to select from the various fonts available.

The FingerPrint card can be considered first of all as a basic printer card and, as such, both parallel and serial versions are available.

These differ at present only in the firmware supplied on the card's Eprom and, of course, in the interface lead supplied.

The user manual supplied with the review board was actually for the parallel version, even though the card was obviously the Imagewriter serial version.

Since the features of both versions were identical, this didn't cause any problems once this confusion was cleared up.

The initial stage of installation is to insert the card into the appropriate slot of the Apple, usually slot 1, connect the cable to the Imagewriter, and boot up the supplied VDAP disc – Verify, Demo, And Preboot.

One little peculiarity to be noted at this point is that, if the Imagewriter is switched on before the Apple, the print head 'hunts' around the left-hand side of the carriage in a most disconcerting fashion.

This doesn't occur when one of the more conventional cards, for example Apple's Super Serial card, is used, and is probably attributable to one of the RS232 handshake lines needing to be linked in the interface lead.

When the VDAP disc was booted, the main menu was presented. Selecting option 1 – installation verify – we were then informed correctly that the FingerPrint card was in slot 1, the firmware date was 11/12/84, 12 November, and that – incorrectly – we had the parallel version of the card.

As was mentioned earlier, there was no doubt that the card was meant for the Imagewriter, so it must be assumed that somebody made a minor slip up in the firmware – not serious enough to stop the card working but enough to confuse any unwary novice.

Pressing any key, we were then presented with guidance as to how to set up the seven DIL switches on the card – this was done, of course, in conjunction with the nicely presented user manual.

Another key press carried out a check on the ROM – that is, eprom – and found no errors. A third key press carried out a RAM test for locations 51200 to 53247 – $C800 to $CFFF – of the on-board static RAM.

Like many RAM tests, the majority of time is spent displaying the address of the current RAM location, rather than testing it. Consequently, this test takes almost $1\frac{1}{2}$ minutes to complete.

Mind you, the test did correctly detect a RAM "fault" that I deliberately induced and, in any case, the installation verify would only normally be done on initial installation.

The final part of the Installation verify is the Print Test. This prints "This is the FingerPrint test" 10 times, then the printable Ascii character set, $21 to $7E.

Now that the card is correctly installed, we can get down to using it.

The first thing to try out is that the card will behave as an ordinary dumb printer card.

This was done by loading an Applesoft program, entering PR #1, then LIST. As expected, this produced the usual program listing.

The next thing tried was option 2 from the main menu – Demo Program. This showed the effect of various control codes.

It then followed with a normal hi-res dump of page 2, containing, the Thirdware logo, followed by the same dump rotated 90 degrees clockwise using Ctrl-I R.

According to the user manual, we should then have been "shown how also to use the FingerPrint button".

For some reason this wasn't included in the review copy of the software. This was disappointing, since this feature is the raison d'être of the card.

At this stage, two things should be done – firstly, read the user manual complete with its revisions and, secondly, select item 6 from the main menu for a clear, concise, overview of the FingerPrint's features.

It was decided to dive in at the deep end and try out the FingerPrint button. This is a one inch square, flat, self adhesive button connected by a flexible flat cable to the card and intended to be fixed in any convenient position on the outside of the Apple's case.

When the FingerPrint button is pressed the active application program stops execution and there is a long bleep from the Apple's speaker.

We then have a choice of pressing any of the ten main key options. A handy stick-on reference strip is also provided with these listed.

Of course, even a straightforward interrupt and resume option is useful. If the phone rings just at the vital moment when you are about to zap some alien invaders you can effectively freeze the entire game and then resume at a more convenient time.

Options 1 and 2 print the 40 and 80 column text area on an Apple IIe, but it is worth noting that the only 80 column card supported on the II+ is the Videx, or exact copies, and that option 2 would not work with my Vision-80.

This clearly restricts the usefulness of the card for a large number of II+ owners.

Option 4 produces a dump of the hi-res page 2, and option 3 in its default mode of operation dumps page 1.

It is also possible to dump the double hi-res screen, available only on the IIe. Option 8 can be used, if required, to revert to the normal default setting of option 3.

Although, as we've already seen, it is possible to interrupt and later resume operation of any program executing in the Apple, one of the most exciting possibilities is to examine the program, for example, disassemble it, and any relevant data.

Option 6 – jump to Apple monitor – provides such a facility, and examination of memory locations $CFF6 and $CFF7 provide the high and low order bytes of the address at the point of interruption.

This works all right as far as it goes but, as Thirdware points out in the user manual, "getting to the monitor is one thing, getting back is another".

It explains that some of the critical information about the state of the Apple at the point of interruption may be changed as a result of entering the monitor.

This is perfectly true, and the only program I was able to get to resume successfully after option 6 was a trivial loop.

It's a great pity that Thirdware didn't address itself to this problem – if it could have sorted it out this would have given the FingerPrint card some of the powerful facilities that are already offered by Dark Star Systems' Snapshot Copykit.

Leaving aside option 7 for the moment – jump to user routine in FingerPrint RAM – option 5, keyboard entry mode is where the FingerPrint card really comes into its own.

This permits the keyboard strokes to go directly to the FingerPrint card. These can be from the FingerPrint commands, data to be printed, or the large number of control codes.

Of the graphics commands, some four of these are standard, such as inversion and rotation, but a fairly unique feature of the FingerPrint is that it caters for colour printers.

This could be very useful for those users with these peripherals, and it was regrettable that I didn't have access to one for this review.

Nevertheless, the attention given by the manufacturer to colour features is commendable, and must be regarded as a very positive "plus" in their favour.

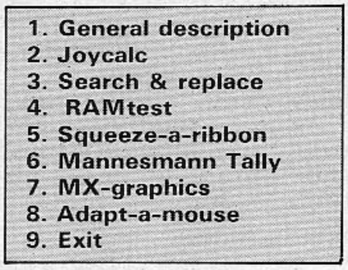
What must be regarded as a very innovative feature is the ability to invoke User RAM routines.

This feature is potentially very powerful, and it is worthwhile looking at this in some detail.

The attraction is that such routines are only limited by the imagination and skill of the programmer and, of course, the 2k memory limit.

It offers Thirdware the option of passing on any useful little routines that it or its customers develops at a minimal cost.

When we select option 5 of the main menu – RAM routine information – we are presented with the menu shown in Figure I.



**1. General description**
**2. Joycalc**
**3. Search & replace**
**4. RAMtest**
**5. Squeeze-a-ribbon**
**6. Mannesmann Tally**
**7. MX-graphics**
**8. Adapt-a-mouse**
**9. Exit**

*Figure I: RAM routine information*

To run any of the RAM applications, the Preboot option from the main menu can be used. This allows us to either load the appropriate routine from the VDAP disc, or create a custom preboot disc if the application is required to be used frequently.

Joycalc allows Visicalc to be used with a joystick emulating a mouse, in that it can move the cursor all over the speadsheet.

Up to this point, the routine is moderately useful. However, Thirdware then go on to provide the facility that, when one of the joystick buttons is pressed, a window is displayed in the upper right corner of the screen and its contents scroll through the character set until the other joystick button is pressed. In this way the selected character is entered just as if it was entered directly from the keyboard.

It must be commented that character selection from the scrolling window might well be novel, but it is very much slower than direct entry.

In fact, I doubt that anyone would want to use Joycalc in this way, because it actually seems to make Visicalc harder to use.

Search and replace allows us to search for a word and replace it by another at the moment of printing.

This seems to be of limited usefulness since, as stated earlier, this kind of facility is much more usefully handled by a word processor, or even a simple text editor.

Squeeze-a-ribbon allows darker printing from an old ribbon by printing every line twice before line feeding.

When I attempted this with an Applewriter file, I wasn't successful. Neither was I able to list an Applesoft program.

The instructions provided on how to invoke this routine were quite explicit, and so no explanation can be offered as to why this didn't work.

Adapt-a-mouse allows the Apple mouse to be used with the Apple II, and the usual range of printers, via the print option of the Mouse Paint program, but this wasn't attempted in the review.

Selecting Mannesmann Tally informed us that this was now available in ROM, and MX-graphics allows the use of all the graphics modes of the Epson MX series printers.

Since the FingerPrint card supplied was intended for an Imagewriter printer, it wasn't clear why the last two options were provided. It can only be assumed that it is related to the confusion, mentioned earlier, as to whether the card was intended for use with an Imagewriter or a printer with a parallel interface.

The RAMtest option was found to be quite useful. It provided a quick test of the 48k motherboard RAM.

This was demonstrated by setting up a loop that printed out the numbers from 1 to 1000. At an arbitary moment during execution, the FingerPrint button was pressed, and this froze the execution program.

Then the key 7 was pressed, which invoked the RAM test. After six seconds a beep from the Apple announced presumably that the test was completed.

After the RAM test, pressing 0 continued execution of the interrupted program.

Since the test was all right, there was no other indication.

Had a RAM fault been detected, we are told that an E would have been displayed in the upper right corner of the screen.

It's a great pity that a simple message, such as "OK" wasn't printed when the test was successful, as it would have been a lot more reassuring.

Also, it surely should be possible to provide an indication of the actual address of any faulty RAM locations.

A little more thought would certainly have made a lot of difference to the usefulness of the RAMtest program. Maybe another option could have carried out a full check of all the Apple's ROMs as well?

The FingerPrint card has been described at great length, but it has to be said that it isn't the easiest of cards to get to grips with.

Thirdware must try to make the card a lot more user-friendly,

〉

and they could do a lot worse than having a look at Dark Star Systems' Printerrupt package.

This is a software package – one of many – that is intended for use with their well-established Snapshot card. It must surely be one of the simplest packages to use anywhere. The documentation is comprehensive, but rarely has to be consulted after an initial read-through.

The Snapshot Printerrupt package comprises an unprotected disc and a 27 page manual.

The first thing to do after installing the Snapshot card is, of course, copy the disc. This backup disc is then booted for configuration of the system. The first step involves selection of the appropriate 80 column card type from IIe 80 column card, Videx Videoterm, Vision-80 or M&R Sup'R'Term.

The first option includes either the standard IIe card, or that with additional RAM. For II+ owners, the other options will probably be sufficient, as many of the available cards are fully compatible with one or the other.

On the other hand, Dark Star Systems' Technical Department offers assistance if your particular card is of an unusual variety.

The second part of the configuration is the selection of the printer card. Of the 23 types covered in the menu, most of the review was carried out using an Apple Super Serial Card.

The final selection was of the printer type and, naturally, we selected the Apple Imagewriter from the 27 options.

The system was tried with other combinations, such as the Practical Peripherals Microbuffer II with an Epson MX100, and worked faultlessly in all cases.

Again, for those very rare interface cards and "bit-image" printers Dark Star Systems offer assistance, either using their ROMgetter program as a means of finding out the characteristics of your printer interface card, or by sending them a photocopy of the command summary in the case of the printer.

Having completed the configuration, a list of the Printerrupt's default settings appeared.

Although these can be altered at any time they would be suitable, as they stand, for a vast majority of applications.

After the configuration, the Printerrupt asked us which slot the Snapshot is in. The appropriate number was entered – 4 in our case – and we were asked to press the Snapshot trigger. A menu then appeared on the screen.

The top half of the screen is termed the main menu and the bottom half is the default menu. At the bottom of the default menu is a solid cursor taking up approximately half the line. This is the form position indicator option, and will be dealt with later.

It may be noted that a Normal/Overstrike item has recently been added to the default menu which eliminates the pinstripe effect that occurs with screendumps on an Epson printer.

In order to move around the main menu we only have to press the right arrow key to move down the menu, and the left arrow key to move up. The current item is shown in inverse video, and this may then be selected by pressing the Return key.

If any of the items in the default menu require changing, it is necessary to first select change defaults from the main menu, then use the arrow keys to move up and down the default menu.

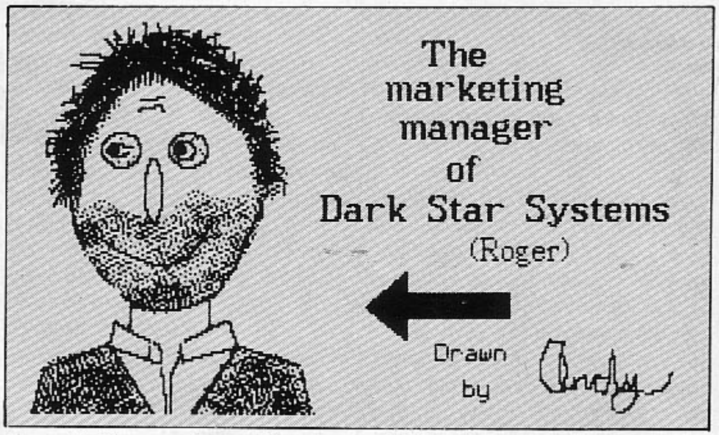


*Figure II: The sample graphics screen*

Rather than inverse video, which is used in the default menu to indicate the current individual item, the cursor now become a pair of square brackets [ ] which enclose the selected line.

The particular option required on the line can then be selected by pressing the spacebar. Pressing the Esc key takes us back again to the main menu.

In order to try out some of the facilities, it was decided to use a sampler of graphics screens provided on the rear of the Printerrupt disc. The Boot-a-disc option was selected from the main menu and we had a selection of six pictures from which to choose.

Five of these were supplied by Norah Arnold. The sixth was a MousePaint picture of another of the Dark Star team – drawn by Andy Beveridge, one of the authors of Printerrupt. Since this was the only picture not copyright, it was used to illustrate the initial graphics dump, as shown in Figure II.

It may not be a particularly pretty sight – I hope "Roger" really doesn't look like this – but it's a convenient example with which to illustrate several of the Printerrupt's features.

When this picture was displayed, the Snapshot button was pressed with the accompanying beep from the Apple's speaker and the Printerrupt menu was displayed.

At this stage, we chose to view the various video modes, one of which was the hi-res page 1 containing Figure II.

To move between the various screens, it was only necessary to press a key – other than Esc, which returned us to the main menu – until the required display appeared.

One very minor point I would raise is whether or not it would be possible to indicate which screen it was that was currently displayed. It this could be done, it would help novices to better understand the various Apple screens.

Returning to the main menu, it was decided to dump the picture to the printer using the default selections except inverse and density 1, then try out the very novel feature of being able to crop the selected screen.

Screen cropping consists of aligning four L-shaped cursors in any of the possible screens so that they define an imaginary box around the required part of the screen.

What I decided to do was select the Andy signature in the bottom right corner of the picture, and crop the rest of the picture out.

This would have left us with rather a small display, so the selections in the default menu were changed to Inverse, Hor.Scale 6, Vert Scale 4, Density 2.

Clearly, as we increase the density, the corresponding size of the picture decreases, and so we can increase the horizontal and vertical scales accordingly.

Although the manual gives some guidance as to how this operates, it is far simpler to just use trial and error. By sensible use of the scaling and density settings, it is easy to get nice dark images on the printer of the required size, and still preserve the correct aspect ratio.

A particularly useful feature, which was invoked when carrying out the screen dumps, was the form position indicator.

This is the facility, in the default menu, which allows us to determine where the image is going to appear on the printed page.

It takes account of the carriage width of the printer, cropping, scaling and density, and allows the cursor to be positioned manually anywhere across the page, in increments of one text character.

This is done by simply using the < and > keys to move the cursor to the left or right, or C to centre it.

A very large number of graphics and text dumps was carried out – I must have got through half a box of printer paper and a couple of ribbons. In all cases, the Printerrupt performed exactly as expected.

It must be commented that it was found useful to be able to rotate both left and right – 90 degrees anticlockwise and clockwise, respectively.

The Set up printer option was used in conjunction with the cropping box to try out some of the more exotic fonts and styles for isolated paragraphs of text, using the printer's command codes.

The set Resume video mode option on the main menu was also tried, since the II+, unlike the IIe, doesn't automatically resume in the correct mode. That left only the shading and the mixing facilities still to try.

The shading facilities allow areas of black or white to be replaced by fine horizontal lines.

It takes a few minutes to get used to, but it can help to give some quite pleasing effects.

The final facility that was tried was that of mixing hi-res pages 1 and 2.

Obviously, each page can be printed separately, and also side by side, by selecting Double. What isn't so obvious, however, is the way in which the two screens can be superimposed.

Anyone familiar with digital logic circuits would find the terms XOR, OR and AND self-explanatory, but to others an illustrative example may be in order.

For this, a checker-board was selected for hi-res page 1. For page 2, a three-dimensional plot (the "trilby hat") was used.

The XOR function causes a black dot to appear on the printout when either one or the other of the dots being superimposed is black.

However, when both are black, a white dot appears. The overall effect is that no detail is lost from either screen.

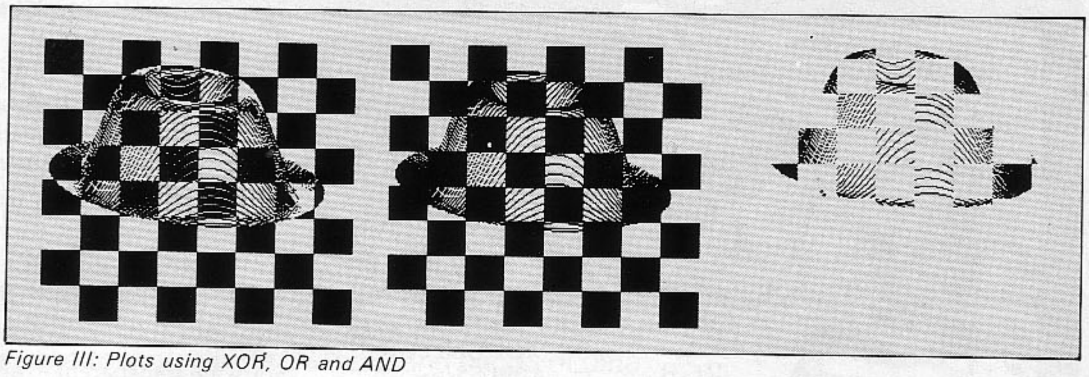


*Figure III: Plots using XOR, OR and AND*

On the other hand, for the OR function, a black dot appears when either or both of the super-imposed dots is black, and results in the effect of the black squares of the checkerboard concealing the pattern behind.

Finally, for the AND function, a rather strange effect occurs, since a black dot only appears on the printer when there is a black dot provided by both page 1 and page 2.

These plots are all shown in Figure III.

Now comes the crunch – which is best, the FingerPrint or the Printerrupt?

The question can be answered in a rather oblique fashion by first of all asking the rather obvious questions. What sort of printer card does the user have already, and does the user already have the Dark Star Systems' Snapshot card?

To make the comparison easier, let's first assume that we have neither. The choice then comes down to either buying the FingerPrint at around £145, or investing in the Snapshot card with the Printerrupt software at £115, and *then* having to buy a printer card on top.

It must be pointed out that the printer card only needs to be very basic, and that, depending on what printer you have, an additional £30-£70 would be involved.

Obviously, if a printer card is already available, there's quite a price advantage in going for the Snapshot.

If not, it's worth looking at what additional features the Snapshot provides. The ability to interrupt, copy, and resume any memory-resident program surely has to be worth the cost of that printer card?

Furthermore, by going the Printerrupt route, you are tapping into the vast resources of the Snapshot system. That means being able to buy additional software packages, such as the multi-tasking Shuttle and the Shell at £20 each. In fact, if the packages are offered together when the Snapshot is purchased, the cost is even less.

Another point worth considering is that Dark Star Systems' technical support is excellent, as I've found on several occasions.

So far, then, we see that the possible extra cost of opting for the Printerrupt, rather than the FingerPrint, is greatly outweighed by the availability of some really exciting software packages.

What we must now look at are the facilities offered.

Both systems provide comprehensive dump facilities for text and black-and-white graphics, with the Printerrupt having the edge by virtue of its independent scaling facilities and the ability to carry out screen cropping.

The FingerPrint supplies, as standard, a very comprehensive set of facilities for colour printers. Although these facilities were not used in the present review, the promotional literature in the American magazines shows that the results can be very impressive.

The Printerrupt package does not offer colour support. Instead, some degree of support for colour is provided by the Screen Snapper package at an additional cost.

Another facility provided by the FingerPrint is a set of demonstration RAM routines.

However, these weren't particularly well chosen in the review copy of the software and, in any case, the limitation of the 2k RAM could prove restrictive for the more exotic applications.

It would have been a lot more useful if Thirdware had provided some software for correctly interrupting, dumping to disc, and then resuming the execution of programs similar to the Snapshot Copykit.

To be fair, though, the RAM routines were included at no additional cost, and it will be interesting to see what Thirdware's charges will be for software updates.

Dark Star Systems' approach is not to provide a potpourri of applications, but to break them up into well-defined areas, and then offer a really comprehensive package for each.

Remember, too, that the Snapshot card has a full 8k of RAM to house this software.

At this stage, we still appear to be in a situation of "pay your money and take your choice".

What clinched the choice for me was the user-friendliness or otherwise of the software provided with the two systems.

When the present bewildering assortment of keypresses in the FingerPrint is compared to the menu-driven and apparently crash-proof approach in the Printerrupt, then I'm afraid I have to come down clearly in favour of the latter.

If Thirdware revises its product so that the whole thing is menu-driven, which I believe it is currently doing, and also thinks a little more carefully about whom the product is aimed at instead of at present trying to offer a little for everyone, then my choice might not have been as clear.

For the time being, however, my cheque goes off to Dark Star Systems.

# Visual data expression – the Mac philosophy

## DUNCAN LANGFORD puts it into practice with easy-to-follow flow charts

THERE are many uses for the MacPaint program which involve drawing pictures, but there is one use which actually takes as its starting-point the Mac philosophy – visual expression of data.

Displaying information in a visual form makes it far easier to understand, which is why programs for the Mac like Microsoft's Chart are so popular.

The same concept is behind the Mac's use of icons to represent its operating systems, rather than the traditional, hard-to-understand DOS or MS-DOS instructions of other computers.

Visual representation of data is fundamental to many other activities. The reason we write lists is because this automatically makes accurate and successful results more likely.

When a large number of related actions are involved, a written check-list is virtually essential.

Involved lists of actions may be helpfully listed visually in a flowchart, commonly used by computer programmers.

When writing a computer program, it is usual to first draw the "flow" of program logic, using a stencil of special shape boxes.

It may look elaborate, but the consequent clear understanding of a program's flow greatly simplifies the actual coding.

The same approach is used for many other activities – instrument checks, emergency procedures, auto servicing – virtually any activity important enough to justify the time needed to construct a flowchart.

Charts in this case are used as guides to follow, enabling complex procedures to be carried out with minimum difficulty.

The reason that there are not far more flowcharts used in home, office and industry is basically that a good chart is hard to draw, and a professionally-drawn flowchart expensive to produce.

Using the Mac, and MacPaint, the process of producing a professional-looking chart is very easy – so easy that you will probably find productive uses for flowcharts that no one else has thought of yet!

The tools you will need are almost all in MacPaint already – the open boxes, both rounded and square, lines, text in various sizes, perhaps a pattern or two, and of course Fatbits.

There are additionally a few drawn objects which I have found by experience to be useful: the words "Yes" and "No" typed inside small boxes, the arrow character from the Cairo font, a double-ended arrow. These may be drawn separately, and saved in the Scrapbook until needed.

Remember, there's no need to actually draw a double-ended arrow; enclose your single arrow in the marquee, press the function key, and click/drag the mouse to clone a copy arrow.

Choose "Flip Horizontal" from the Edit menu to produce a reversed image, then loop around it with the lasso and move it over, joining the two arrows together.

The actual process of drawing a flowchart falls into three stages – identification of the task, detailing of the various steps involved in completing it, and finally actually drawing the chart.

First, decide which tasks the chart is intended to illustrate. Ideally, the job should be a sequential one, involving periodic selection between various options.

The first time I drew a chart using the Mac, my brief was just to show new staff a flowchart, so the subject I selected wasn't exactly a high-level one: "How to get Steve to make the coffee".

Once the task has been identified, the steps necessary to accomplish it should be carefully worked out in detail.

Remember to cater not only for the usual flow of events, but also for the unexpected response, so that the chart is able to handle all possible alternatives.

The intention is that someone totally unfamiliar with the task you are illustrating would be able to understand what to

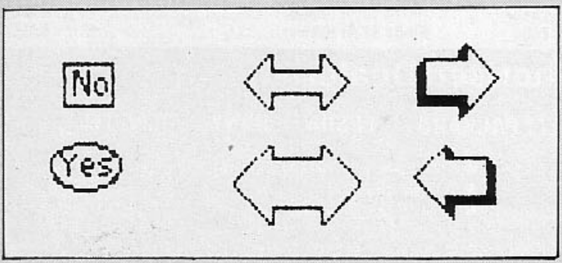


*Figure I: Some useful symbols*

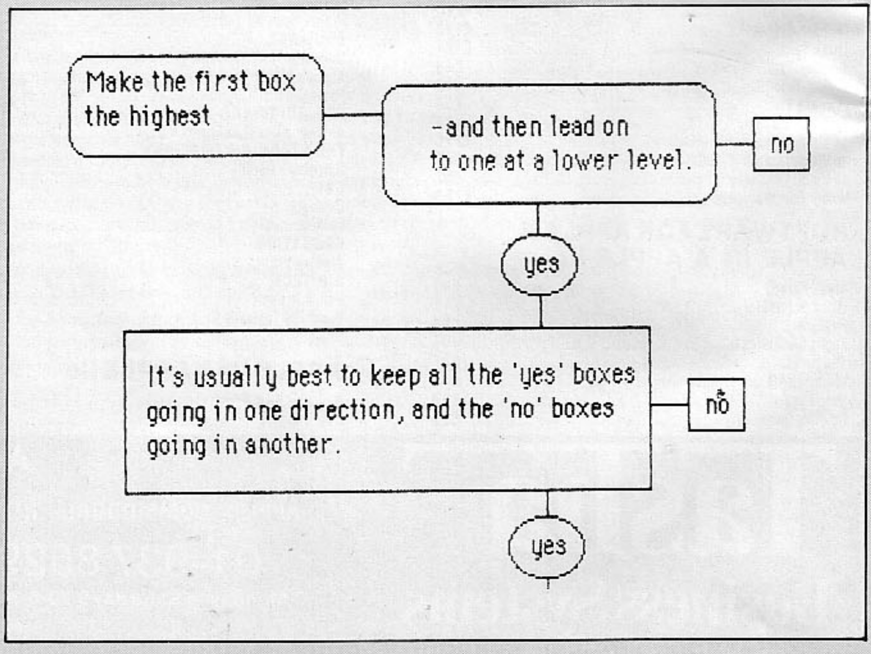
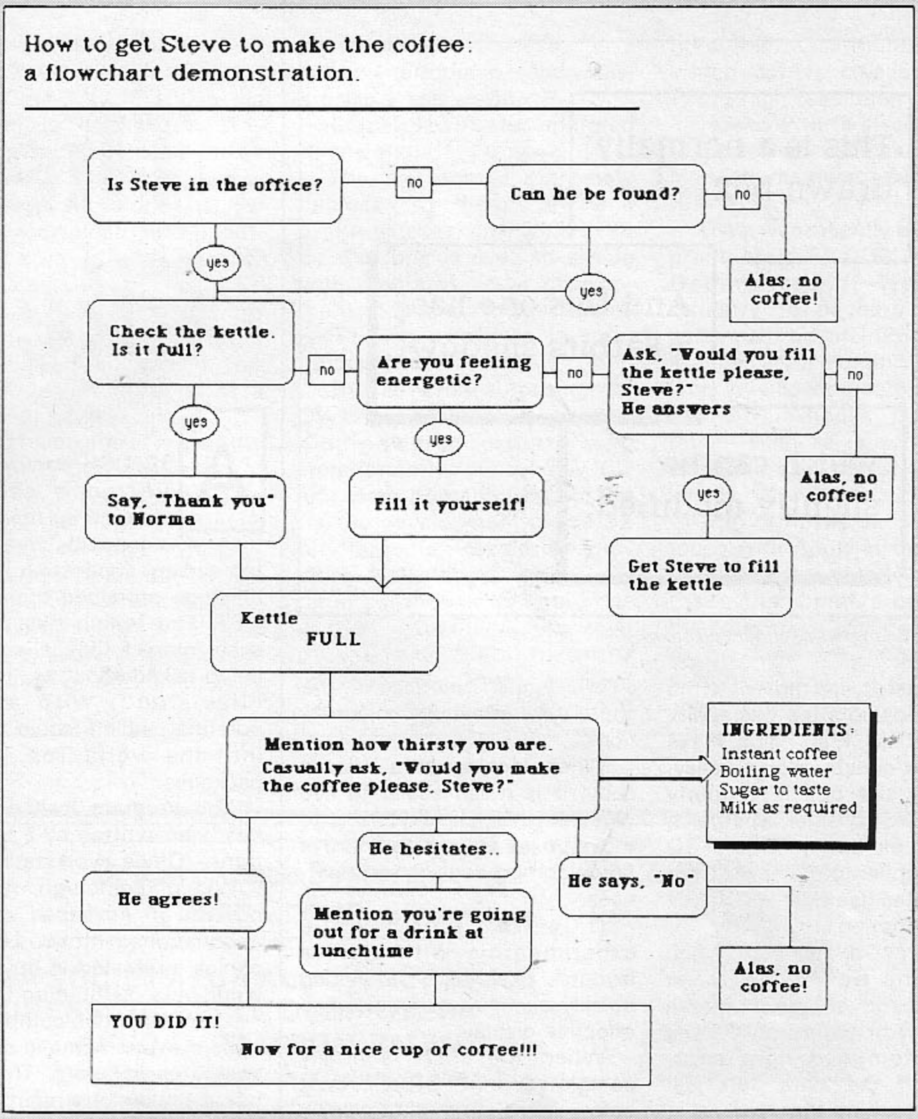

Make the first box the highest
-and then lead on to one at a lower level.
no
yes
It's usually best to keep all the 'yes' boxes going in one direction, and the 'no' boxes going in another.
no
yes

*Figure II: Box arrangement and consistent yes/no flow*


How to get Steve to make the coffee:
a flowchart demonstration.
Is Steve in the office?
no
Can he be found?
no
Alas, no coffee!
yes
yes
Check the kettle. Is it full?
no
Are you feeling energetic?
no
Ask, "Would you fill the kettle please, Steve?" He answers
no
Alas, no coffee!
yes
yes
yes
Say, "Thank you" to Norma
Fill it yourself!
Get Steve to fill the kettle
Kettle FULL
Mention how thirsty you are. Casually ask, "Would you make the coffee please, Steve?"
INGREDIENTS: Instant coffee Boiling water Sugar to taste Milk as required
He hesitates..
He says, "No"
He agrees!
Mention you're going out for a drink at lunchtime
Alas, no coffee!
YOU DID IT! Now for a nice cup of coffee!!!

do by following your flowchart.

Make a list of all these options, perhaps in the form of a very rough hand-drawn flowchart. For my coffee example, I had a list which looked like this:

- Is Steve in the office? If not, can you find him?
- If you can't find him, no coffee.
- If he is around, is the kettle full? If it is, Norma must have filled it – say "Thank you" to her.
- If it isn't full, and you're feeling energetic, fill it – otherwise, ask Steve to fill it. If he won't, no coffee.
- When the kettle is full, say how thirsty you are, and ask Steve to make coffee. If he hesitates, mention you're going out for a drink at lunchtime (That should do it!)
- If it doesn't, no coffee.

As you can see, even an apparently simple task may well produce a confusing list of options.

When the list is complete, draw out in rough the way the chart will probably look. Don't worry if the draft is untidy or covered in loops and arrows – correcting it easily is what the Mac is for.

Choose whichever format you find easiest to follow. Generally, the flow should start at the top of your chart, and move downwards. It's sensible to make "yes" and "no" choices always flow in a consistent direction:

Draw out the chart on these lines by hand, revising it as you go. There's no need for your boxes to be square, or the lines to be straight – provided you are able to understand the results, neatness at this stage doesn't matter at all. Concentrate on fitting in all the information!

If you are not totally sure of all the relevant data you need to enter, check with colleagues or a specialist in that area to make quite certain you have covered all the angles.

Remember that the more complex an operation the more a flowchart is needed, and the less likely it is that any one person will have all the necessary knowledge.

There may perhaps be a need for additional information. For example, a "Does the applicant qualify?" question may need a list of qualification criteria. Or a chart illustrating a cookery process may need a list of ingredients.

This data is not strictly a part of the flow, and if you try to include it inside a "flow box" the result could well be confusing. The answer is to simply mark with an arrow a special "data box" at the appropriate part of the chart. When your rough chart is completed, the time has come for you to turn on your Mac, and actually start to construct the real flowchart.

Go to "Show Page", and set up the visible screen to the top of the page, before returning to the program.

Text is best entered in either a small font (Andover, for example) or in 9-point. I normally use New York 9-point, but experiment to find which suits your current chart best, and set up that font and size.

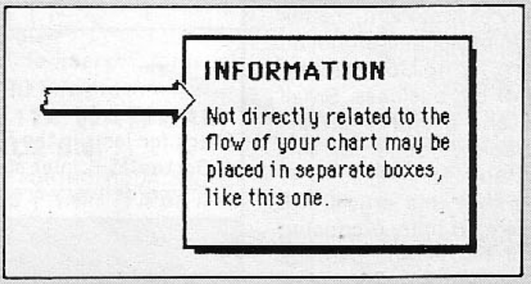

INFORMATION
Not directly related to the flow of your chart may be placed in separate boxes, like this one.

*Figure III: Example of an information box*

The next stage is the one I find most enjoyable.

Type in the text for each of your boxes in a clear area of screen, and then draw a box around it.

Generally, it's easier to type text first, box second, and then lasso the text to centre it in the box, rather than drawing a box and trying to fit text inside it.

The boxes should be connected by straight lines, preferably from/to the centre of a side. If a box isn't in exactly the right

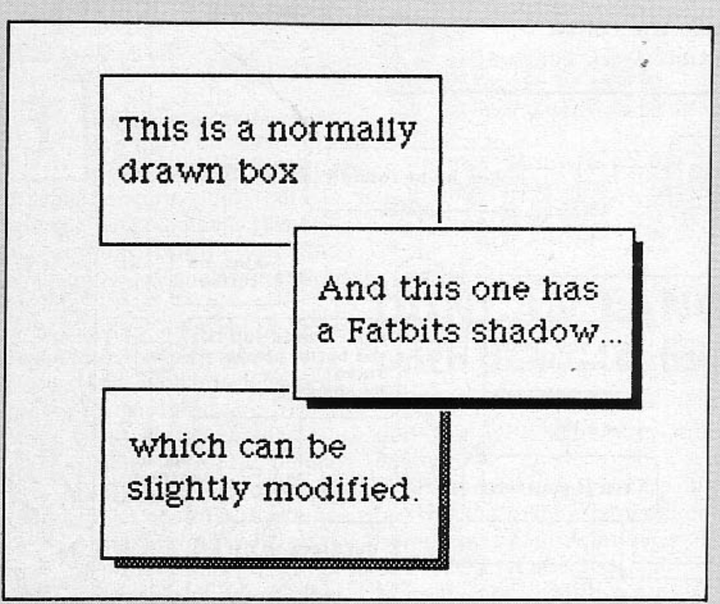

This is a normally drawn box
And this one has a Fatbits shadow...
which can be slightly modified.

*Figure IV: Examples of enhanced boxes*

place, lasso it, and move it to the correct position. It's this ability of MacPaint that helps make flowchart construction so easy.

When the box is properly positioned, consider whether it would look better with a 3D effect, similar to the boxes used by the Mac itself when displaying information.

To draw a "shadow", first make the box. Then enter Fatbits, and simply draw an extra row or two of dots along the bottom and right edge. Taper off the extra dots just before reaching the corners.

The modified box in Figure IV had three extra rows of dots added, and then the middle two rows erased. The gap was filled with a pattern using the paint bucket.

To make sure new text exactly lines up with the old – difficult in MacPaint, particularly to anyone who has used MacWrite – type in a clear part of the screen the last few characters of the old text, followed by the new.

Lasso this, and move it to the required spot.

When overlapped with the existing old text, duplicate letters will disappear, leaving the new text perfectly aligned.

As the full page is used by a chart, it is of course necessary to make sure the layout fits properly on the page – and here, although frequent use of "Show Page" helps, nothing can really equal regular printing of the chart throughout its construction.

When looking at a printed copy, it is much easier to see whether there is any need to move boxes around, to improve flow, comprehension or neatness.

It is well worth while experimenting with various layouts. Experience is gained quickly on what constitutes an effective display.

When the chart is finally completed, print "DRAFT" on it before circulating photocopies around – it is at this stage that the inevitable errors and omissions show up!

Fortunately, revision is so easy that it is worth returning a chart for only minor polishing, after which the definitive edition may be finally printed.

Flowcharts have been used as valuable tools in industry and commerce for years, but the difficulty of producing them has prevented much use elsewhere.

The advent of the Mac, and particularly of MacPaint, means that a professionally-produced flowchart is no longer the preserve of big business. Small business, and even home uses will be rapidly found for this powerful tool.

I hope that this article will encourage yout to try producing a chart or two – and perhaps even encourage Steve to make the coffee!

# The next Filevision?

**BILL HILL previews Guide, which can integrate MacPaint graphics with text to simplify the manual writer's job**

A BRITISH company has written a software package for the Macintosh which could be the first top-selling application for the machine produced outside the USA and which has created such interest that it is already being talked about as "the next Filevision", with a sales potential which could take it into the world Top Twenty packages.

The program, called Guide, has been written by a Scottish firm – Office Workstations Ltd (OWL) of Edinburgh, which is now at an advanced stage of negotiations with two US companies interested in buying the worldwide distribution rights.

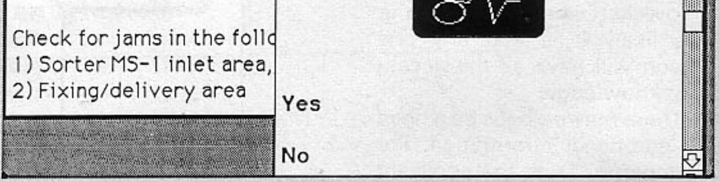
One of the two companies is a major world name in personal computer software. The other has a prestigious reputation in high-resolution computer graphics.

Although the package is not expected to be on sale before September or October of this year, development is complete apart from a few final details which are being left open at this stage until the deal is concluded, and *Apple User* has had a preview of the system.

OWL describes Guide as a "dynamic documentation" package, and it is capable of storing both text and graphics in a hierarchical structure, in some ways reminiscent of Filevision.

However, while Filevision insists that graphics must be drawn within the program itself, Guide allows graphics created in either MacPaint or MacDraw to be integrated with text in an extremely usable manner.

One likely application for the package – almost enough in



This is the JAM indicator. It lights up if copy paper jams in the copier. Use the following procedure to remove jams.

Is this indicator lit up?

Check for jams in the foll

1) Sorter MS-1 inlet area,

2) Fixing/delivery area

*Faultfinding with Guide*

itself to ensure Guide of a large potential market – is in the production of manuals and troubleshooting guides for complex equipment.

Instead of having to refer to a manual, the user would have the document stored on disc. It is capable of interaction – for example, taking him by a series of Yes/No questions through a "faultfinding tree" to find out what is wrong with the equipment and how to repair it, almost like an expert system.

OWL has already written two such examples – a troubleshooting guide for photocopiers, and a manual for the correct uses and handling of computer floppy discs – to illustrate the program's uses. Both contain illustrations, for example showing the various parts of a photocopier to explain its operation in visual terms.

The key to Guide's usefulness is the way in which the program is configured. Any pieces of text or graphics can be linked, using a facility known as an "expansion button" (just like having Macintosh dialog buttons built into key words or illustrations).

The best way to illustate this is to give another example which I have seen running. This is an index of editions of a number of newspapers.

On opening, the user sees the "mastheads" of a number of papers – for example The Guardian, the New York Times, and the Washington Post.

If he uses the mouse to click on any of these mastheads, Guide then zooms in to a scrollable list of the headlines in that edition of the newspaper. By clicking on these, it is possible to zoom in to individual articles – and so on.

Graphics can be incorporated at any stage.

The package is extremely impressive, and there are none of the glitches, crashes and so on that one is used to seeing with programs under development. In fact, the most impressive thing about it is how professional it looks.

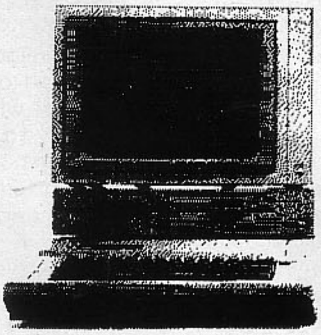
This is hardly surprising, for OWL was set up by a group of high-powered experts who were formerly employed by ICL, and were primarily responsible for the development of the 32 bit Perq high-resolution graphics workstation, which was far in advance of its time.

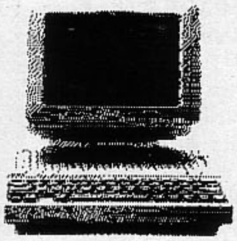
OWL executives say they picked the Macintosh for software development because it is the mass-market micro which comes closest to the Perq's high-resolution graphics.

However, it is also intended to port Guide across to other high-volume micros such as the IBM PC.

OWL was recently awarded a grant of £250,000 from the Department of Trade and Industry to develop another software product, InPrint, which is aimed at revolutionising office work by allowing companies to develop in-house documentation, such as reports, faster, cheaper and more professionally.

The company is closely in touch with Apple in the UK. In fact, it has one of the first Apple Laserwriter printers outside of the company – which was used to produce the screendump seen here.





# Beating the bug in ProDOS CHAIN

While developing a large Applesoft/ProDOS application recently (using ProDOS Version 1.0.2, released Feb. 15, '84), I came across a bug in the ProDOS CHAIN command.

This bug occurs, on average, only once every 256 times during the course of program development. It is thus even worse than one of those now-you-see-it-now-you-don't bugs.

The following two Applesoft programs demonstrate the bug. Save this to disc as PART.A:

```
10 X=1
20 D$=CHR$(4)
30 DIM A$(165)
40 PRINT D$"CHAIN PART.B"
```

Save this as PART.B:

```
10 IF NOT (D$=CHR$(4)) THEN
   LIST: STOP
20 PRINT "ALL OK"
```

Run PART.A and the message "ALL OK" will appear, showing that PART.B has been chained with variables intact.

Now load PART.A, delete line 10, and run it. Part B will be chained, but the variable D$ no longer has its original value, and PART.B will stop at line 10.

What's going on here? The obvious place to start to track down the bug is to check the values of the Applesoft variable pointers before and after the CHAIN command is executed. The pointers themselves will have changed, since they depend on the length of the program being chained, but if the CHAIN command preserves the variables intact then the bytes which are pointed to should be the same.

The relevant Applesoft pointers are:

| | | |
|---|---|---|
| VARTAB | $69,6A | Pointer to the start of the simple variables. |
| ARYTAB | $6B,6C | Pointer to the start of the array variables. |
| STREND | $6D,6E | Pointer to the end of the array variables. |

The bug manifests itself only when line 10 of PART.A is removed, so run the following program:

```
20 D$=CHR$(4)
30 DIM A$(165)
40 CALL -151
```

This will leave us in the Apple Monitor, with the Applesoft variable pointers as they are at the time the CHAIN command is executed. If you type "69.6E" the Monitor will respond with:

```
0069- 27 08 2E 08 27 0A
```

This gives us the following values for the Applesoft variable pointers: VARTAB = $827, ARYTAB = $82E and STREND = $A27.

The logical way to proceed in our bug-tracking is now as follows:

Take a note of the first few bytes which are pointed to by VARTAB and ARYTAB. Replace line 10 in PART.B with CALL –151 and save this to disc as PART.B. Load PART.A, delete line 10 and run it.

When PART.B drops into the monitor after the chain, again check the first few bytes pointed to by VARTAB and ARYTAB.

This would show us that the bytes pointed to by VARTAB and ARYTAB are different after the chain than they were before.

But this simply confirms that the variables are not being preserved by the CHAIN command, which we knew anyway from line 10 of PART.B. It does not tell us what is causing it.

The clue lies in noting the unusual. In this case it is that the low byte of STREND is the same as the low byte of VARTAB (both before and after the chain).

VARTAB and STREND are independent, so this generally occurs only once every 256 times. Could it be that the bug in the ProDOS chain command occurs when and only when the low bytes of these two pointers are the same? The answer seems to be Yes.

This is confirmed by adding a variable (as is the case in our original PART.A, where we let X = 1) in order to cause the low byte of STREND to be different from the low byte of VARTAB.

Thus it seems that the ProDOS CHAIN command fails to preserve the values of the variables when the combined length of the simple variables and the array variables (that is, STREND-VARTAB) is an exact multiple of 256.

Thus if you find your program crashing after a ProDOS CHAIN you can probably fix things simply by adding a dummy variable to the program using the CHAIN command.

For program development to continue it is enough to know the conditions under which a bug manifests itself, so as to avoid those conditions. We do not have to know exactly why the bug manifests under those conditions.

But in this case we can hazard a guess.

The ProDOS CHAIN command first moves all the simple and array variables up in memory to just below the strings.

It then loads in the new program and moves the variables back down to the end of the new program.

Presumably the actual machine code in ProDOS which does this does not properly handle the case where the length of the bytes being moved is an exact multiple of $100 (decimal 256).

No doubt the bug can easily be eliminated, and one hopes that Apple will do so in a future release of ProDOS, which is generally a very fine operating system.

**Peter Meyer**

# Changing line numbers

Here's a Basic programming trick which other readers may care to share. Enter this program:

```
10REM OTE PORT 163 [RETURN]
```

Now enter the machine monitor with a CALL–151 and type the following:

```
803:FF FF [RETURN]
806:8 [RETURN]
```

Exit the monitor by typing Ctrl-C, Return and LIST the program. You will see:

```
65535 REMOTE PORT 163
```

The line numbers for the first line of Basic are stored at $803 and $804. The first character after the REM token is stored at $806.

We have changed it to H (the backspace character) so that the space put out by LIST is lost.

Note that line 65535 cannot now be deleted, but it can if we change it back to 10 by entering the monitor typing:

```
803:A 0 [RETURN]
```

So we have changed program line numbers and this can be used to add to your subroutine library, perhaps like this:

```
65535 REMOTE PORT 5,56,A$300
65534 REM — COPYRIGHT 1985
65533 REM — BY X.Y.Z.
65532 REM — FOR EVERMORE
```

Note that a GOTO or GOSUB after lines such as these will not be accessible to Basic so only use them at the end of programs. My REM subroutine includes:

```
65535 REM IBM — I Buy Macintosh
```

**Seija Teromoto**

# Businesslike way to better Apple III programs

IN the 3½ years since the Apple III Business Computer has been available in this country, very few books have been written about the Business Basic programming language for this machine.

Now a book has appeared on the United Kingdom market which deals with Apple III Business Basic and assumes the person reading it has no previous programming knowledge.

The first few chapters explain the various types of numbers, numeric variables and arithmetic functions found in the world of computing and mathematics in terms which, for the layman, are easy to understand.

*Business Basic for the Apple III* by Eddie Adamis *(John Wiley, £14.40).*

The book then moves on to deal with an equally important function used in Business Basic, namely the "Type Conversion Function".

Here I feel the author should have left this chapter until he had dealt with string functions and then included the method of converting a numeric string containing a decimal portion (i.e. pence) into a long integer, as in Business Basic the only way to store an actual number with more than six figures is to use long integers.

After these chapters on logic and operator functions, the book turns to the writing of programs and setting up of files.

Again, the author assumes that the reader has no previous programming knowledge, and explains the techniques involved in terms which the reader can easily understand.

He guides him with ease through the methods of writing program statements, subroutines and the use of screen display and formatting output for a hard copy of the results.

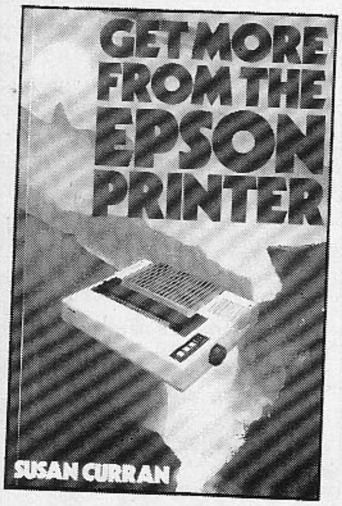
The book finishes with some excellent chapters on setting up files and how to handle them, again explained in terms which are easy for the beginner to understand.

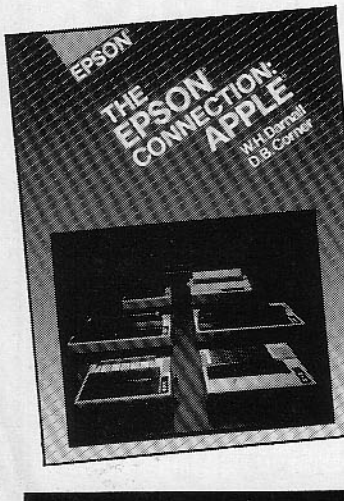
This book is in general well written, explaining the Business Basic language in very understandable terms both for the beginner and the experienced programmer.

These explanations probably clear up any misunderstandings that the Business Basic instruction book supplied with the Apple III could have created.

And it should help both types of people develop better programs for the Apple III.

**Ian Searle**





# POINTERS FOR PRINTERS

PROBABLY one of the most frustrating problems new computer users face is making the printer work as desired. Most computer magazines have letters from readers asking questions on how to use the features of their printers.

The problem is somewhat exacerbated because of the bewildering array of printers, printer interface cards, data formats, and application programs, etc.

Two new books have recently been published to help people better understand and use Epson printers, the market leader in the UK. One succeeds, the other doesn't.

The first book Get More From the Epson Printer, by Susan Curran, is only slightly more successful than the Epson Operator's Manual in explaining the Epson to Apple computer users.

*Get More From the Epson Printer,* by Susan Curran, *(Collins, £7.95).*
*The Epson Connection: Apple,* by W.H. Darnall and D.B. Corner, *(Reston/Prentice-Hall, £18.30).*

Curran uses the Basic dialects of Microsoft (language of the Epson manuals), BBC, Commodore and Sinclair QL to demonstrate Epson printer control codes and their effects.

Only seven Epson printers (P-40 and 80, RX-80 and 100, FX-80 and 100, and JX-80) are mentioned.

Curran puts most emphasis on the FX-80, the Epson she uses. The older models (like the MX-80 F/T that I use) are considered "superseded" and are not treated.

Curran says that much of the book is applicable to these older printers, but unfortunately never gives specific information about the differences.

The book has chapters on setting up the printer/computer combination, explanation of Ascii codes, and serial and parallel interfacing. Printer control codes are covered in three chapters.

There are more chapters on user-defined character sets, bit-image graphics and screen dumps, and general discussions of commercially available utility programs and buffers, and stationery.

One disappointing characteristic of the copy of Get More From the Epson Printer I reviewed was the tendency for the pages to fall out! This problem may be limited to my copy only, but is still unacceptable for a paperback costing £7.95.

The book is too limited to be of widespread interest, too complicated for the novice, and has insufficient details to satisfy the experienced Epson printer user.

In conclusion, Get More From the Epson Printer probably would not be of interest to most Apple users.

On the other hand, The Epson Connection: Apple, by W.H. Darnall and D.B. Corner, succeeds where Curran's book fails.

The Epson Connection is well-organised, professionally designed, and is a very informative book.

The "blurbs" on the jacket say that with this book "... you'll have quality information that translates into building a productive understanding of the Epson printer". In my view, this is true.

The book has specific information on all Epson printers, including the earliest MX-80 Type I Firmware model, and is clearly directed to applications-orientated Apple users.

Each topic is first approached with the aim of getting immediate results, and then explained in detail for those who want to pursue the topic in greater detail.

There are three parts. The first part outlines the capabilities and limitations of the different Apple computers, Epson dot matrix printers, Comrex (division of Epson) daisywheel printers, and various printer controllers and cables.

A shopper, as well as an experienced Apple user, will learn much about Epsons and Apples and what they can do together.

Excellent specific and detailed information is included.

For example, the authors have included information about most of the most popular printer interface cards for the Apple/Epson combination.

Some readers may be disappointed that pin allocations for serial and parallel interfaces, unlike most other Epson books, are not discussed. But my view is that this subject is best left for technicians, not users.

The chapters on explaining the Epson and Comrex printing features and control codes is where Epson Connection shines over Get More From the Epson Printer.

All Epson/Comrex features are explained and demonstrated. There are also detailed tables showing whether or not each feature is implemented on each Epson, and how each feature is activated.

The format allows the reader to easily locate the desired feature and the control codes to activate it, for all Epson printers.

This is especially useful for people who have access to different Epson printers (home and office, perhaps).

Locating information is easy. The table of contents consumes seven pages, and the detailed index is nine pages long.

There is a chapter on Epson/Comrex graphics capabilities, including a few examples of printing metacharacters and bit image mode.

This section is not comprehensive (perhaps because the author's interests are not in programming) but for Apple users it is well presented and more comprehensible than the Epson manuals.

Epson Connection has a chapter on "Printing from Application Programs" in which specific information on using Epson/Comrex printers with WordStar, AppleWriter and VisiCalc is given.

This is useful since the manuals for these programs do not have specific advice for the Epson/Apple combination.

Also included is a section with perceptive advice about deciphering software documentation with respect to printer control codes for the Epson/Comrex.

The last part of Epson Connection discusses applications for the Apple and Epson/Comrex combination.

Applications ideas and advice for business, scientific, professional writing, artistic and personal uses are discussed.

This is a good book. Epson/Apple users, especially new applications-orientated users, should consider including The Epson Connection in their computer library.

**Rob Schneider**

By PETER GORRY

# Automatic production of graphs

## Part XIV of the Apple User Graphics Library

WE have finally arrived at the point where we can add automatic graph production routines to the *Apple User* graphics library. The job of these routines is to relieve you of all the messy tasks of deciding graph ranges, divisions, tick marks and annotation details.

The routines also provide borders, grids and a host of user defined options including the ability to overide all of the default values if you so wish.

Perhaps the best way to explain how the routines operate is to examine the sorts of questions you have to answer when plotting a graph. First of all you have to decide what ranges it must cover.

This isn't as simple as it may seem. For instance, you might not know what ranges the data actually covers till you've seen it.

In general there are three cases to be covered – no advance knowledge, a rough idea of the ranges and an exact predetermined format. All three cases are catered for.

Having determined the range to graph over, one must then decide on suitable divisions for each axis. This in itself is no simple task since the division size should be a sensible number – for example, 20 not 19.45 – and it should produce a reasonable number of tick marks along the axis, say 5-10 not 27 or 2.

With that out of the way you must then decide whether you want a grid imposing on the graph horizontally, vertically or both. In addition, if the axis values go from negative values to positive values you must decide if you want the tick marks to be along the zero line or the graph edge.

Finally comes the task of labelling the axes. How many decimal places do you need to show? What about very big numbers or very small numbers? What if there isn't room to label every tick mark? Should you label along the edges or along the zero lines?

Considering the complexity of the questions to be answered it's not suprising that the routines given this month are not particularly short, but by using several subroutines to perform the tasks it is fairly easy to see how the process works. In fact some of the subroutines can be used independently of the library if required.

One remaining question is what happens if one of the ranges isn't numbers but words – days, months or geographical areas? The problem of incorporating such symbolic data will be dealt with next month.

As in all previous cases, the routines presented here should be added to the rest of the library since they make extensive use of the other routines. Also, adopting the usual format, the routines are controlled by arrays beginning with the letter Z.

The X axis routine draws the X axis and any tick mark/grid/zero lines required. It is not normally called directly by the user although one could alter the controlling array ZU( ) directly if required.

**ZU(1)** 0=numbers, 1=symbols (option 1 available after next month).

**ZU(2)-(4)** XMIN, XMAX, division size all in user units.

**ZU(5)** Controls the zero line. 0=don't draw, 1=draw, −1=draw and annotate.

**ZU(6)** Controls the grid. 0=no, 1=yes.

**ZU(7)** If a grid is selected this holds the grid size (user units). If there is a 0 here it will use the value in ZU(4).

**ZU(8),ZU(9)** X,Y coordinates of the axis title.

**ZU(10)** The number of decimal places used for annotation.

The equivalent routine for the Y axis is controlled by the array ZV in the same manner.

The Range Finding routine is used to decide on sensible limits, division size, tick marks and decimal places given arbitrary minimum and maximum values.

On entry ZA and ZB should contain the minimum and maximum values. On exit ZA and ZB have been replaced by new values which start and stop on sensible numbers, ZD contains the division size, ZE the number of tick marks and ZZ the number of decimal places needed for labelling.

This routine uses no others in the library and could be used independently if desired. In normal use it is called internally and is not accessed by the user.

The Set Limits and Draw Axes routine is the heart of the graph layout section and it replaces the original border routine given in the very first article of the series. It decides how each axis is to be drawn and what grids, zero lines, etc are to be included.

Before calling the routine it is essential to ensure that the array ZM contains (at least) the screen size of the graph and also any values for which you don't want calculated values. In addition you must have set the page and colour before calling it.

The routine is controlled by the array ZG.

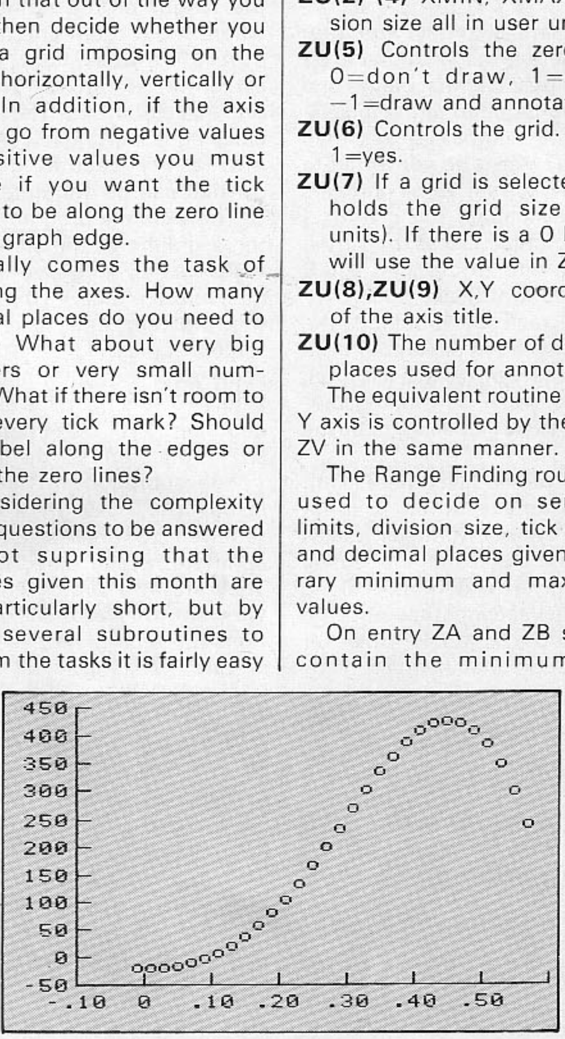
**ZG(0)** determines how the graph limits are to be found.

A value of 0 forces the routine to use the values currently set in ZM(1)-ZM(4). It will calculate the division size using the tick mark numbers held in ZG(1) and ZG(2). This option should only be used for a completely predetermined graph format and it is the user's job to ensure that the values in ZM(1)-ZM(4) and ZG(1) and ZG(2) are sensible. You should also ensure that ZU(10) and ZV(10) contain the appropriate number of decimal places needed for annotating.

A value of 1 will cause the limits currently held in ZM(1)–ZM(4) to be taken as a first guess. The routine will then calculate appropriate new limits, division sizes and decimal places based on these guesses. The original values are then replaced by the new ones, and ZG(1), ZG(2) and ZU( ) and ZV( ) will be set up accordingly.

A value of 2 will search the usual data arrays ZX( ),ZY( ) for their maximum and minimum vlaues and it will make these the first guess in deciding the graph ranges.

Before calling this option it is



*Sample output from the routines*

›

essential that the data be in the arrays and that ZN contains the number of data points. This option is the simplest of all to use since it requires no input from the user.

**ZG(1),ZG(2)** hold the number of X,Y axis tick marks. Only option 0 above needs these setting by the user.

**ZG(3)** determines whether a border is drawn round the plotting window or not: 0=no, 1=yes.

**ZG(4)** determines whether a grid is drawn on the graph: 0=none, 1=along X axis (vertical lines), 2=along Y axis (horizontal lines), 3=both.

**ZG(5)** determines whether a line is to be drawn through zero if an axis covers positive and negative values. 0=none, 1=X, 2=Y, 3=X and Y. If this parameter is set but the graph range does not include zero then no line will be drawn. It is thus safe to set this option even if you don't know the graph ranges in advance.

**ZG(6)** controls where the annotation will take place. A value of zero will cause the numbers to be written along the left hand size and along the bottom of the graph. Other values are used to force the annotation to be along the zero lines: 1=X, 2=Y, 3=X and Y.

The options which control the graph layout, ZG(3)-ZG(6), should be set before calling the routine. The default values of 0 will produce graphs with no border, no grid, no zero lines and annotated along the graph edges. This is the least cluttered type of display.

At first glance it might seem an easy task to produce a string corresponding to a given number since Basic offers just that facility in the STR$( ) function. Unfortunately, as a little experience soon shows, this is not good enough if neat layout is required.

> ‘... routines that can produce professional-looking graphics’

Basic outputs numbers with varying numbers of digits after the decimal point, and small errors in the way numbers are stored may result in values such as 2.0000001 being given instead of 2.0. Furthermore, although it happily writes 1000 as 1000, it converts 0.001 to 1E-03.

Conversion to scientific format is useful for very large or small numbers but .001 is hardly too long to fit on a graph axis!

The simple number formatter given here takes a number in Z1 requiring Z2 decimal places and puts the required string in ZS$. Only if a number is less than 0.00001 will it be converted to scientific format (large numbers use the standard Basic limit).

If a number is less than 1 the leading zero is dropped – 0.5 becomes .5, to save space.

The task of labelling the graph has been separated from that of drawing it since it is often quicker, when developing programs, to put labelling in at the end.

The routine given here is not yet complete since it must handle symbolic data as well as numbers. The rest of the routine will be given next month – along with the section that puts on titles and axis labels. However the section that handles numbers is complete and the routine will operate perfectly for numeric data.

The routine picks up the required information for each axis from the ZU( ) and ZV( ) arrays. It does not change any of the text parameters, so if you have selected inverse-underlined the graph will be labelled in this way.

It uses the machine code version of the text routines. If you use the Basic one you will need to change GOSUB 43460 to GOSUB 42800 in lines 48720 and 48840.

The routine keeps a tally of the length of the strings and where it put them and it won't plot a number that would overwrite part of the previous one. In this way the graph labelling is kept distinct, even if only every second or third division can be annotated.

The example program draws three graphs of the same data with varying degrees of user intervention. Line 110 loads in the plotting and hi-res text tables. Lines 120-130 puts some data into ZX( ) and ZY( ). Line 140 sets the page as usual and line 150 determines the size of the graph on the screen.

The subroutine at line 500 issues the three instructions required to draw the graph, annotate it, and plot the data. They are in a subroutine since we want to issue those instructions for each graph.

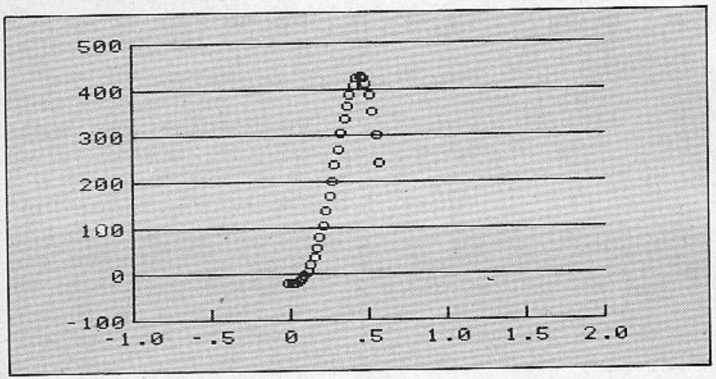
Line 170 shows all that is required to draw the first graph. No decisions are required by the user at all, neither is it necessary to know anything about the data beforehand. Pressing the space bar will take you to the next graph.

Lines 210,220 show how the user supplies initial values for the minimum and maximum graph values, a horizontal grid is also selected. No other information is required. Again pressing the space bar moves on to the next graph.

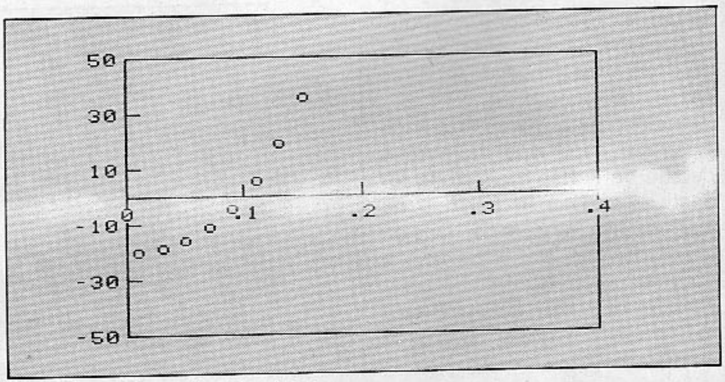
Lines 260,270 show how the user can determine the exact format of the graph. The values chosen cover only a small range of the total data and illustrate how a magnified image of a particular region can be selected using this technique. As well as setting the limits/divisions, a border and X axis annotation along the zero line is also selected.

The number of decimal places in ZU(10), left over from the previous graph, is correct for this one too. If that hadn't been the case ZU(10) would have to be set as well.

The graphics library is nearing completion, and those of you who have persevered with it now have a collection of routines that can produce professional-looking graphics with a minimum of effort. In the remaining couple of instalments I will deal with symbolic data and those "asking" routines I promised.



*Example of a horizontal grid*



*Negative and positive Y-values*

```
90  DIM ZX(30),ZY(30)
100  REM

EXAMPLE PROGRAM

110  GOSUB 42400: REM  SHAPE TABL
     E LOADER
120 ZN = 30: REM  DATA POINTS
130  FOR I = 1 TO ZN:ZX(I) =  - 3
     0 + 20 * I:ZX(I) = ZX(I) / 1
     000:ZY(I) =  - 20 + I * I *
      SIN (5 * ZX(I)): NEXT
140 ZC = 3:ZP = 1:ZF = 1: GOSUB 4
     0000: REM  SET PAGE
150 ZM(5) = 40:ZM(6) = 250:ZM(7) =
     160:ZM(8) = 10: REM  SCREEN
     VALUES
160  REM  SET GRAPH FROM DATA
170 ZG(0) = 2: GOSUB 500: REM  AL
     L DEFAULTS
180  GET A$: IF A$ <  > " " THEN
      GOTO 180
190  REM  NOW USE SUPPLIED VALUES
      AS FIRST GUESS
200  GOSUB 40000: REM  CLEAR PAGE

210 ZM(1) =  - 1:ZM(2) = 2:ZM(3) =
     - 100:ZM(4) = 500:
220 ZG(0) = 1:ZG(4) = 2: GOSUB 50
     0: REM  HORIZONTAL GRID
230  GET A$: IF A$ <  > " " THEN
      GOTO 230
240  GOSUB 40000
250  REM  NOW TOTALLY USER SUPPLI
     ED
260 ZM(1) = 0:ZM(2) = .4:ZM(3) =
     - 50:ZM(4) = 50
270 ZG(1) = 4:ZG(2) = 5:ZG(4) = 0
     :ZG(3) = 1:ZG(0) = 0:ZG(6) =
     1: GOSUB 500
280  END :

500  REM  DRAW,LABEL AND PLOT
510  GOSUB 47600: GOSUB 48600:ZT =
     1: GOSUB 42600
520  RETURN :

46700  REM

X AXIS ROUTINE

46710  REM  CONTROLLED BY ZU ARRA
     Y
46720  REM  1 NUMERIC OR SYMBOLIC
      0/1
46730  REM  2-4 XMIN,XMAX,DELTAX
     - USER UNITS
46740  REM  5 1=LINE THROUGH ZERO
               , 0=NONE
46750  REM  6 0= NO GRATICULE, 1=
     GRATICULE
46760  REM  7 GRATICULE DELTAX, 0
     =USE ZU(3) VALUE
46770  REM  8,9 X,Y FOR AXIS LABE
     L
46780  IF ZU(5) = 0 THEN  GOTO 46
     810
46790  IF ZM(3) * ZM(4) > 0 THEN
      GOTO 46810
46800 YP =  FN YCN(0): HPLOT ZM(5
     ),YP TO ZM(6),YP
46810 YP = ZM(7): IF ZU(5) =  - 1
      AND ZM(3) * ZM(4) < 0 THEN
     YP =  FN YCN(0)
46820 ZA = (ZM(6) - ZM(5)) / 25:Z
     B = (ZM(7) - ZM(8)) / 25: IF
     (ZB < ZA) THEN ZA = ZB: REM
      TICK MARK LENGTH
46830  HPLOT ZM(5),YP TO ZM(6),YP
     : REM   AXIS
46840  IF ZU(5) = 1 THEN YP = ZM(
     7)
46850  FOR ZI = 0 TO ZG(1):ZX = Z
     U(2) + ZU(4) * ZI:XP =  FN X
     CN(ZX): HPLOT XP,YP TO XP,YP
      - ZA: NEXT
46860  IF ZU(6) = 0 THEN  GOTO 46
     910
46870 ZA = ZU(4): IF ZU(7) > 0 THEN
     ZA = ZU(7)
46880  FOR ZX = ZM(1) TO ZM(2) STEP
     ZA
46890 XP =  FN XCN(ZX): HPLOT XP,
     ZM(7) TO XP,ZM(8)
46900  NEXT : REM  X GRID DONE
46910  RETURN :

47000  REM

Y AXIS ROUTINE

47010  IF ZV(5) = 0 THEN  GOTO 47
     040
47020  IF ZM(1) * ZM(2) > 0 THEN
      GOTO 47040
47030 XP =  FN XCN(0): HPLOT XP,Z
     M(7) TO XP,ZM(8)
47040 XP = ZM(5): IF ZV(5) =  - 1
      AND ZM(1) * ZM(2) < 0 THEN
     XP =  FN XCN(0)
47050 ZA = (ZM(6) - ZM(5)) / 25:Z
     B = (ZM(7) - ZM(8)) / 25: IF
     (ZB < ZA) THEN ZA = ZB: REM
      TICK MARK LENGTH
47060  HPLOT XP,ZM(8) TO XP,ZM(7)

47070  IF ZV(5) = 1 THEN XP = ZM(
     5)
47080  FOR ZI = 0 TO ZG(2):ZY = Z
     V(2) + ZV(4) * ZI:YP =  FN Y
     CN(ZY): HPLOT XP,YP TO XP +
     ZA,YP: NEXT
47090  IF ZV(6) = 0 THEN  GOTO 47
     140
47100 ZA = ZV(4): IF ZV(7) > 0 THEN
     ZA = ZV(7)
47110  FOR ZY = ZM(3) TO ZM(4) STEP
     ZA
47120 YP =  FN YCN(ZY): HPLOT ZM(
     5),YP TO ZM(6),YP
47130  NEXT
47140  RETURN :

47300  REM

RANGE FINDING ROUTINE

47310  REM  ON ENTRY ZA=MIN,ZB=MA
     X
47320  REM  ON EXIT ZA=NEW MIN,ZB
     =NEW MAX,ZD=DELTA,ZE=NO. TIC
     KS, ZZ=N.D.P.
47330 Z3 = ZB - ZA:Z2 =  INT ( LOG
     (Z3) /  LOG (10))
47340 Z1 = Z3 / (10 ^ Z2)
47350  IF Z1 < 2 THEN ZD = 2 * (1
     0 ^ (Z2 - 1)): GOTO 47380
47360  IF Z1 < 7 THEN ZD = 5 * (1
     0 ^ (Z2 - 1)): GOTO 47380
47370 ZD = 10 ^ Z2
47380 ZZ = 0: IF ZD < 1 THEN ZZ =
      ABS ( INT ( LOG (ZD) /  LOG
     (10)))
47390 ZE = Z3 / ZD: IF  INT (ZE) -
     ZE < 0 THEN ZE = ZE + 1
47400 ZE =  INT (ZE)
47410  IF ZE > 10 THEN ZE =  INT
     (ZE / 2 + .5):ZD = ZD * 2
47420 Z4 =  INT (ZA / ZD) * ZD:Z5
      = ZE * ZD + Z4
47430  IF Z5 < ZB THEN ZE = ZE +
     1: GOTO 47420
47440 ZA = Z4:ZB = Z5: REM  NEW L
     IMITS
47450  RETURN :

47600  REM

SET LIMITS AND DRAW AX
     ES

47610  REM  CONTROLLED BY ZG ARRA
     Y
47620  REM  OPTIONS 0,1 REQUIRE Z
     M ARRAY 1-8 SET, 2 REQUIRES
     ONLY 5-8
47630  REM  ZG(0) = 0 KEEP MAPPIN
     GS - USE ZG(1),ZG(2) AS SET
47640  REM            1 CALCULATE S
     UITABLE RANGE
47650  REM            2 SEARCH ZX()
     ,ZY() FOR LIMITS
47660  REM  ZG(1) NUMBER OF X TIC
     KS
47670  REM  ZG(2) NUMBER OF Y TIC
     KS
47680  REM ZG(3) 0= NO BORDER, 1=
     BORDER
47690  REM  ZG(4) SETS GRID; 0=NO
     NE, 1=X, 2=Y, 3=X AND Y
47700  REM  ZG(5) SETS ZERO LINE;
      0=NO, 1=X, 2=Y, 3=X AND Y
47710  REM  ZG(6) ANNOTATE ON ZER
     O; 0=NO, 1=X, 2=Y, 3=X AND Y
47720  IF ZG(0) = 0 THEN  GOTO 47
     950
47730  IF ZG(0) = 1 THEN  GOTO 47
     890
47740  REM  SEARCH ARRAYS
47750 ZA = ZX(1):ZB = ZX(1)
47760  FOR ZI = 1 TO ZN
47770  IF (ZX(ZI) < ZA) THEN ZA =
     ZX(ZI)
47780  IF ZX(ZI) > ZB THEN ZB = Z
     X(ZI)
47790  NEXT : GOSUB 47300: REM  S
     ET X RANGE
47800 ZM(1) = ZA:ZM(2) = ZB:ZG(1)
      = ZE
47810 ZU(2) = ZA:ZU(3) = ZB:ZU(4)
      = ZD:ZU(10) = ZZ
47820 ZA = ZX(1):ZB = ZX(1): FOR
     ZI = 1 TO ZN
47830  IF (ZY(ZI) < ZA) THEN ZA =
     ZY(ZI)
47840  IF (ZY(ZI) > ZB) THEN ZB =
     ZY(ZI)
47850  NEXT : GOSUB 47300
47860 ZM(3) = ZA:ZM(4) = ZB:ZG(2)
      = ZE
47870 ZV(2) = ZA:ZV(3) = ZB:ZV(4)
      = ZD:ZV(10) = ZZ
47880  GOTO 47980
47890  REM  USE ZM() VALUES AS FI
     RST GUESS
47900 ZA = ZM(1):ZB = ZM(2): GOSUB
     47300
47910 ZM(1) = ZA:ZM(2) = ZB:ZG(1)
      = ZE:ZU(2) = ZA:ZU(3) = ZB:
     ZU(4) = ZD:ZU(10) = ZZ
47920 ZA = ZM(3):ZB = ZM(4): GOSUB
     47300
47930 ZM(3) = ZA:ZM(4) = ZB:ZG(2)
      = ZE:ZV(2) = ZA:ZV(3) = ZB:
     ZV(4) = ZD:ZV(10) = ZZ
47940  GOTO 47980
47950  REM  USE GIVEN VALUES
47960 ZU(2) = ZM(1):ZU(3) = ZM(2)
     :ZU(4) = (ZM(2) - ZM(1)) / Z
     G(1): REM  X AXIS
47970 ZV(2) = ZM(3):ZV(3) = ZM(4)
     :ZV(4) = (ZM(4) - ZM(3)) / Z
     G(2): REM  Y AXIS
47980  REM  NOW SET GRID
47990 ZU(6) = 0:ZV(6) = 0
48000  IF ZG(4) = 1 OR ZG(4) = 3
      THEN ZU(6) = 1
48010  IF ZG(4) > 1 THEN ZV(6) =
      1
48020  REM  NOW ZERO LINE
48030 ZU(5) = 0:ZV(5) = 0
48040  IF ZG(5) = 1 OR ZG(5) = 3
      THEN ZU(5) = 1
```

❯

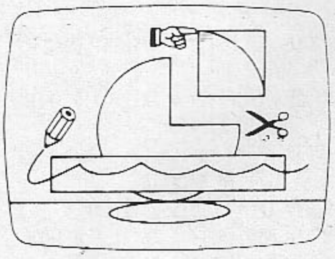
```
48050  IF ZG(5) > 1 THEN ZV(5) =
     1
48060  REM  NOW ANNOTATION
48070  IF ZG(6) = 1 OR ZG(6) = 3
      THEN ZU(5) =  - 1
48080  IF ZG(6) > 1 THEN ZV(5) =
      - 1
48090  REM  NOW THE BORDER/MAPPIN
     GS
48100  IF ZG(3) = 0 THEN  GOTO 48
     120
48110  HPLOT ZM(5),ZM(7) TO ZM(5)
     ,ZM(8) TO ZM(6),ZM(8) TO ZM(
     6),ZM(7) TO ZM(5),ZM(7)
48120  GOSUB 40200: REM  SET MAPP
     INGS
48130  GOSUB 46700: REM  X AXIS
48140  GOSUB 47000: REM  Y AXIS
48150  RETURN :

48300  REM
NUMBER FORMATTER

48310  REM  NUMBER IN Z1, NUMBER
     OF DECIMAL PLACE IN Z2
48320  REM  RETURNS WITH STRING I
     N ZS$
48330 ZS$ =  STR$ ( INT ( ABS (Z1
     )))
48340  IF Z1 < 0 THEN ZS$ = "-" +
     ZS$
48350  IF Z2 = 0 OR Z1 = 0 THEN
      RETURN
48360  IF Z2 > 5 THEN ZS$ =  STR$
     (Z1): RETURN
48370 Z3 =  ABS (Z1) -  INT ( ABS
     (Z1))
48380 Z3 =  INT (Z3 * (10 ^ Z2) +
     .5):Z3$ =  STR$ (Z3)
48390  IF  LEN (Z3$) < Z2 THEN Z3
     $ = "0" + Z3$: GOTO 48390
48400  IF ZS$ = "0" THEN ZS$ = ""

48410  IF ZS$ = "-0" THEN ZS$ = "
     -"
48420 ZS$ = ZS$ + "." + Z3$
48430  RETURN :

48600  REM
AXIS LABELLING
48610  REM  LABELS THE X AND Y AX
     IS AND TITLE
48620  REM  X AXIS FIRST
48630  IF ZU(1) = 1 THEN  GOTO 48
     860
48640 YP = ZM(7) + 9: IF ZU(5) <
      > - 1 THEN  GOTO 48660
48650  IF ZM(3) * ZM(4) < 0 THEN
     YP =  FN YCN(0) + 9
48660 ZY =  FN UYCN(YP):Z2 = ZU(1
     0)
48670 Z5 = 0: REM  END OF LAST ST
     RING
48680  FOR Z4 = ZU(2) TO ZU(3) STEP
     ZU(4)
48690 Z1 = Z4: GOSUB 48300: REM
     NUMBER
48700 Z6 =  LEN (ZS$) * 3.5:XP =
      FN XCN(Z4) - Z6 + 3.5
48710  IF XP < Z5 + 4 THEN  GOTO
     48730
48720 ZX =  FN UXCN(XP): GOSUB 43
     460:Z5 =  FN XCN(ZX) + Z6 +
     4
48730  NEXT
48740  REM  Y AXIS
48750  IF ZV(1) = 1 THEN  GOTO 48
     870
48760 XP = ZM(5): IF ZV(5) <  > -
     1 THEN  GOTO 48780
48770  IF ZM(1) * ZM(2) < 0 THEN
     XP =  FN YCN(0)
48780 Z7 = XP:Z2 = ZV(1): REM  RI
     GHT HAND SIDE
48790 Z5 = 191:Z9 = 279: REM  MAX
      Y,X
48800  FOR Z4 = ZV(2) TO ZV(3) STEP
     ZV(4)
48810 Z1 = Z4: GOSUB 48300:Z6 =  LEN
     (ZS$) * 7:ZY = Z4:Z8 =  FN Y
     CN(ZY)
48820  IF Z5 - Z8 < 10 THEN  GOTO
     48850
48830 XP = Z7 - Z6: IF XP < Z9 THEN
     Z9 = XP
48840 ZX =  FN UXCN(XP): GOSUB 43
     460:Z5 = Z8
48850  NEXT
48860  REM  TO BE CONTINUED
48870  RETURN :
```



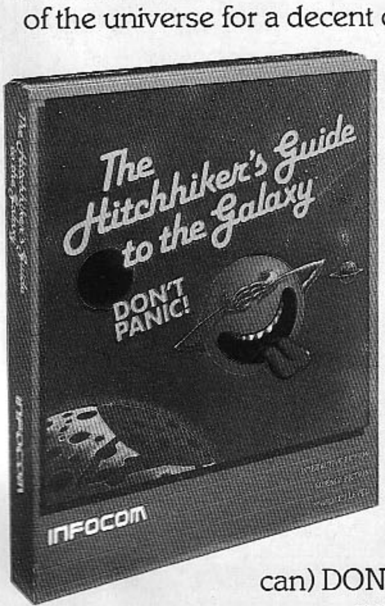
"THE PHOTO SOURCE"

# To boldly drink where no man has drunk before.

If you've ever been served with a cup of tea that resembles canal water. Or begrudgingly sipped a brew with as much taste as a gold lamé string vest, you should be able to sympathize with Arthur Dent.

Star of Infocoms'™ hilarious new software extravaganza. The Hitchhiker's Guide to the Galaxy.™

Now programmed for use on the Apricot™ and a host of other machines "Hitchhikers" gives you the chance to slip into Arthur Dent's shoes and scour the length and breadth of the universe for a decent cuppa.



### DON'T PANIC

Gulp with excitement when your planet is demolished by ruthless galactic hyper-space planners.

And split your sides as your home is bulldozed by the local council. You'll laugh so much, it'll bring the house down.

Whatever happens (and just about anything can) DON'T PANIC! For all this will prepare you for a few dangers that are a little harder to swallow.

### ONE LUMP OR TWO

A journey throughout the universe can be a pretty nervewracking business. Especially when there's a whole host of perilous flying saucers, black holes and burning stars to deal with.

And when the Ravenous Buggblatter Beast of Traal asks "Would you like one lump or two?" He's not referring to sugar!

If you sense trouble is brewing you can just zap into another solar system and continue your voyage. But don't forget your towel.

The Hitchhiker's Guide to the Galaxy is available on Earth from your local Softsel dealer. Along with over 2,600 other titles from over 250 publishers.

So don't just sit there stewing – Go and cause a stir in the galaxy.

The Hitchhiker's Guide to the Galaxy – It's out of this world.

SOFTSEL®

**The number one distributor of software. In the world.**

Softsel Computer Products Ltd, Softsel House, Syon Gate Way,
Great West Road, Brentford, Middlesex TW8 9DD.

Hitchikers Guide to the Galaxy is a trademark of Douglas Adams. Infocom is a trademark of Infocom Inc. Apricot is a registered trademark of A.C.T. Ltd.

# Hey, Mac – here's how to strike Gold!

*IN the March issue, under the heading Window on the World, Bill Hill very ably describes the ease with which a modem, MacTerminal and a Macintosh could readily access Telecom Gold.*

*Regettably it is not easy – as many like myself, after spending our money, have found out.*

*Would you kindly send me a listing or details of the terminal settings and compatibility setting you used? Perhaps, better, a blow-by-blow account of constructing your Telecom Gold file would be more acceptable.*

*Try as I have I get little response from either MacTerminal the program and manuals or from Telecom Gold.* – **C. Clinton-Carter, West Wellow, Hants.**

● I'm sorry to hear that you are having problems accessing Telecom Gold with your Macintosh and a modem. I'm using a standard 128k Mac, with a Minor Miracles WS2000 modem and MacTerminal, and I must say that even though I access the system daily neither hardware nor software has ever given me any trouble.

The best thing I can do, I think, is to give you the step-by-step account of setting up a MacTerminal File for Telecom Gold. Forgive me if I restate the obvious.

First, boot up the MacTerminal disc. Once on the desktop, open MacTerminal with the mouse. Go to the Settings menu and pull it down. Highlight Terminal first, and enter the following settings:

- Terminal: TTY.
- Mode: This line goes inactive after choosing TTY on Line 1.
- Cursor shape: This is a personal preference but I like a block cursor.
- Line width: 80 columns.
- Protocol converter: Line is inactive when TTY chosen.

Of all the boxes at the bottom of the screen, only two should be selected (selected ones are marked with crosses) – ON Line and Auto repeat.

Click the OK box with the mouse. This window closes. Pull down Settings menu again. Highlight compatibility and release mouse. Settings should be as follows:

- Baud rate: 300.
- Bits per character: 8.
- Parity: None.
- Handshake: None.
- Connection: Modem.
- Connection port: Telephone icon.
- Click OK with the mouse.
- Pull down the Settings menu one last time and highlight character sets. Check that UK Ascii is selected for both G0 and G1 options. In fact, if you are running a UK Macintosh with a UK operating system and a UK version of MacTerminal, this selection should be automatic.

Finally, once all the settings have been entered (you can ignore file transfer and answerback message meantime) go to the File menu, highlight SAVE AS..., and, when prompted, type the title of your new file – I called mine, obviously enough, Telecom Gold.

From now on, all you have to do when accessing Telecom Gold is open the Telecom Gold icon on the desktop, dial your local node, and when you receive the tone, switch your modem to On Line.

Type in two Returns, then type A2, then another two Returns. If you don't get a system response right away, do it again another couple of times until you do.

It goes without saying, of course, that your modem should be correctly connected. It should be set to 300 baud full duplex, and Originate if you have that option.

I agree with you that the manuals you receive from Telecom Gold are none too helpful. Since I've never had the chance to see the MacTerminal manual, I can't speak for its clarity or lack of it. – **Bill Hill.**

## You need a chain...

*I AM a student and a newcomer to Apple computing. I am using Apple IIe with 64k extended 80 column card for my work and with the help of Applesoft Programmer's Assistant version 1.0, 1979, I can do all renumbering and merging very easily.*

*Recently I tried to merge one part of the program which is about 29k with another part of about 13k. By loading the first part (29k) and &Hold, I then load the second part (13k).*

*Unfortunately, the answer is "Program too long". I suppose that my computer can handle up to 128k, but why not with the program of less than 50k?*

*It would be very much appreciated if any reader could advise how to merge such program, and is there any other method that could merge and handle up to 128k as claimed?* – **M.B.A. Ghani, Glasgow.**

● As things stand, in Basic, you are using a 64k computer. Your extra 64k is just not used. You will have to "chain" from one part to another in some way.

Pascal 2 *can* use the full 128k of memory. In Basic you will have to write your own routines to handle the bank switching and storage problems.

The article by Paul Overaa on page 31 might get you started on this road. – **Max Parrott**

## Pascal file bug

*I HAVE quite frequently used Paul Smith's Dosinterface program (Windfall, February-April, 1982) to access DOS text files and have uncovered a bug in the program when copying from DOS to a Pascal file.*

*In the function CPYTEXTP, the statement:*

```
WHILE (TRACK<>0) AND
    (SECTOR<>0) DO ...
```

*is intended to signal the end of the textfile.*

*However, any data in the zeroth sector on any track is caught by this statement and is accordingly not transferred.*

*However, it does not end the transfer of data and the other sectors' data are transferred correctly.*

*This bug became apparent when I discovered that, on transferring a text file to the printer, a few records in the (randomaccess) text file appeared shorter than the others!*

*To overcome this problem, I have replaced the condition with the following:*

```
WHILE NOT ((TRACK=0)
AND (SECTOR=0)) DO ...
```

*I hope others have not suffered like me.* – **Harold Binley, Bristol.**

## Two more creep in

*THE following bugs crept into the April issue:*

*In Duncan Langford's Appletip on page 11, the location to peek and poke in line 70 should be 43624.*

*The correct price for the Machine Level Programming book reviewed on page 53 is £17.95.*

*Prentice-Hall inform us that the book without the disc can be bought for £7.95 and the disc can be obtained separately for £10 plus VAT.*

*We offer our apologies for the first mistake and Prentice-Hall offer theirs for the second.*

〉

# RWTS – not as simple as it looked ?

*I HAVE just read Michael King's article on using the DOS 3.3 RWTS directly from a Basic program published in the May issue with great interest.*

*I broached this subject a few years ago when I wrote a small program to find and print out the start address and length parameters of a binary file (see Practical Computing 1982, Vol. 5 No. 6).*

*I hope these few notes are of use both to users of Mr King's programs and Mr King himself and suggest he obtains a copy of Beneath Apple DOS rather than relying on the scant information provided in The DOS Manual if he wants to use the RWTS to its best advantage.*

*Firstly I would like to point out that Mr King is wrong in stating that the DELETE command simply marks the file's directory entry. It also releases any sectors used by the file for future use.*

*It does this by deallocating the sector bit masks in the volume table of contents, or VTOC sector. The structure and location of the VTOC sector can be found on page 132 of The DOS Manual.*

*What this means is that simply fixing the file's directory entry will not ensure that any future programs or files saved to the disc will not overwrite the sectors used by the undeleted file.*

*There is, however, a simple solution to the problem.*

*Proceed with Mr King's OOPS program and undelete your file as before, but as soon as possible thereafter load, bload or read the file's contents into memory, delete the file, intentionally this time, and resave it back to disc.*

*This will ensure that DOS creates a complete record of the file's usage of the disc as well as restoring the directory entry.*

*Another tip of relevence to Mr King's article is that you do not have to create your own IOB or DCT for use with the RWTS – as DOS, not surprisingly, already has both set up in its memory and they are easier to use by the programmer as the IOB already contains information relating to the last disc operation so only the minimum will have to be changed.*

*To use this internal IOB and DCT you only need to know the start address of the IOB and calculate any necessary byte offsets from this with reference to either Mr King's IOB table description or the one documented on pages 96-98 of the DOS manual. The start address of the IOB is:*

**IOB = 47080**

*If you use DOS's own IOB the RWTS calling routine may also be shortened as DOS already has a routine to point to its own IOB. The new calling routine bytes are now:*

**32,227,3,32,217,3,96.**

*Furthermore, if the calling routine is poked into some free memory in page 3 (from 768 to 975 is normally free), then you only need one page of memory or 256 bytes that you need to allocate as a sector buffer and protect from Basic. Lowering HIMEM by one page can be achieved simply by two statements:*

**POKE 116,PEEK(116) – 1**
**POKE 112,PEEK(112) – 1**

*This also means that the sector buffer start can be easily found by the following statement:*

**BUFADR = PEEK(115) + PEEK(116) * 256**

*Using all of these points, an initialising subroutine to set up access to the RWTS would be as follows:*

```
60000 RWTS=768: FOR I=RWTS
   TO RWTS+6: READ B: POKE
   I,B: NEXT: REM Set up
   RWTS calling routine
   utilised by CALL RWTS
   from Applesoft
60010 DATA
   32,227,3,32,217,3,96,:
   REM RWTS calling routine
   values
60020 IOB=47080: REM Locate
   start of IOB table
60030 POKE 116,PEEK(116)-1:
   POKE 112,PEEK(112)-1: REM
   Lower HIMEM to reserve
   space for sector buffer
60040
   BUFADR=PEEK(115)+PEEK(116)
   *256: REM Locate start of
   sector buffer
60050 RETURN
```

*Mr King's other point that DOS recognises a value of 0 as the end of a Text file only applies to sequential access text files. The end of a random access text file cannot be determined in this way.*

*Compounding the problem is the fact that the file's record length is not stored in any way with the file, so the whole file can only be read by reading in every sector allocated to it.*

*Also appearing in Mr King's IOB table description is the fact that command code 0 at byte 13 of the IOB is Initialise for RWTS.*

*This command code is in fact used to move the disc read/write arm to the correct track before attempting to read or write the sector and is more commonly described as a SEEK command, although it can be*

# 'PRETTY' ADVICE

*THANK you for sending me the RWTS program on my disc.*

*However . . .*

*The program Pretty Print contains a "bug" virulent enough to prevent it from working. The offending bacillus resides in line 750, where the variable 'FTYPE' is tested for validity.*

*The values shown are 1,2,81 and 82, ie INT and FP unlocked files, and INT and FP locked files.*

*When the program finds an unlocked file it operates correctly, but a locked file results in a report "Not a Basic file", and the program exits.*

*The reason for this can be determined by reference to Professor Luebbert's What's Where in the Apple, page 129, fig. 17.3E, which reveals that the file type codes – 1,2,81 and 82 are HEX numbers!*

*To make the program work, it is necessary to change the values 81 and 82, to 129 and 130, the decimal equivalents, as required by Basic.*

*As one who only ventures to dip a toe in the edge of the pool, I am only too well aware how errors of this type arise when programming, but I must say that it seems somewhat remarkable that one should evade detection to such a late stage.*

*May I take this opportunity to offer my congratulations on your magazine, to which I have subscribed since its inception. –* **E.H. Phillips, Derby.**

★ ★ ★

*THANK you for sending the programs for Dodge DOS with RWTS.*

*Below is a small update to the Pretty Print program that will ensure that colons are only treated as statement separators when they are not enclosed in quotes. –* **Lee Hammond, Chatham, Kent.**

```
920 A$=STR$(K) + "      ":
   A$=LEFT$(A$,6): J=J+2:
   FL=1: QF=0: BP=PAPERWTH:
   REM START THE LINE WITH
   THE LINE NR AND GO TO
   START OF TEXT
927 IF PEEK(BUFFERST + J)=34
   THEN QF=NOT(QF)
940 IF
   CHAR$(PEEK(BUFFERST+J)) =
   ":" AND QF=0 THEN PRINT
   A$: A$="      ": LC=LC+1:
   REM COLON IN TEXT, SO
   PRINT LINE SO FAR AND
   INDENT READY FOR NEXT
```

*used to initialise an external IOB as done by Mr King.*

*Byte 12 of the IOB is not always 0 as described, it is in fact a byte count used to read or write a partial sector, although it is normally 0 to enable the full 256 bytes to be transferred.*

*This enables very flexible use of the RWTS to read or write only up to the bytes required in the sector.* – **Allan Ogg, Glasgow.**

## Easy way

*READERS who intend to use D. Poirier's Verify program from the May issue (Page 32) might find it helpful to substitute the following re-writing of line 36 for the one given in the listing. This makes getting the screen layout correct a lot easier than trying to count the spaces involved in the listing.* – **W.J. Davis, Dublin.**

```
36 HTAB 19: VTAB 3:
   PRINT"1111111111111111222
   ": HTAB 3:
   PRINT"0123456789ABCDEF0123
   456789ABCDEF012"
```

## Magician runs into snags

*I HAVE come across several problems while using my Apple IIe.*

*First of all, I have an adventure program, written in Applesoft, which I want to turn into a graphics adventure, using pictures created with Graphics Magician.*

*The program is quite large (it takes up 53 sectors on disc) and consequently it runs straight into BOTH hi-res pages.*

*I tried to relocate the program above hi-res page 2, by using the command LOMEM:24576 as the first line in the program, but it didn't work, since every time I use the command HGR or HGR2, part of the program is scrubbed from memory.*

*I tried changing LOMEM first, and then loading the program again, but that didn't work either. Can anyone help?*

*My second problem concerns the use of Apple Pascal (1.1). I only have one disc drive, so I followed the single-drive boot procedure as recommended – that is, boot the APPLE3 disc, then insert the APPLE0 disc in the drive and press the Reset key.*

*Pressing Reset by itself does nothing, so I used Control Reset. This has exactly the same effect as just booting APPLE0 on its own, which results in the usual message "No File System. Apple".*

*Until recently, I thought that something was wrong with my Apple, and that I would have to have two drives before I could use Pascal properly.*

*I could boot APPLE1 with no problems, but all I could do is make text files ... I can't compile programs because the compiler is on APPLE2, which is supposedly in drive 2.*

*Then I read an article on Pascal, on page 34 of the November 1984 issue of* Apple User. *It mentioned the boot procedure for an Apple II+ as I described before.*

*It then goes on to say: "With the Apple IIe things are a little more difficult, because the Reset key operates in a different way. To get round this requires a modified procedure with a modified set of discs".*

*Could someone please explain to me what the article means about the Reset key being different, and also how to make the modification that it mentions?*

*I have been searching for an answer to this problem for ages now, and if anyone could help me, I would be eternally grateful.*

*Another programming problem I have recently encountered is about shape tables.*

*Does anyone know of any registers which change when two shape tables "collide" or when they mix with the background?*

*I have just typed in the VBLANK demo program in the April 1985 issue of* Apple User, *and I was very impressed.*

*I then had the idea of using this technique of mixing text and graphics to create window effects and pull-down menus.*

*I tried changing the text window, but it made no difference to the result.*

*Does anyone know if it is possible to make text windows on the hi-res screen?* – **Leon Seltsikas, Hendon, London.**

**Want to write to Feedback? The address is: Feedback, Apple User, Europa House, 68 Chester Road, Hazel Grove, Stockport SK7 5NY.**

● LOMEM:24576 starts the variable storage space after the two hi-res pages but does *not* change the area where the program resides. In order to move it above the hi-res pages you have to issue the commands POKE 24576,0:POKE 103,0:POKE 104,96 before loading the program.

To set everything back to normal, issue the command 'FP'. However, I suspect that your adventure game will not run at this position because it will not have enough memory left for storing the variables. Try it and see!

If it does not run, then you will have to split the program around the hi-res screens. If you want more details on how to do this, send me a blank disc.

The Reset key on the IIe does nothing by itself and when people refer to it they really mean Ctrl-Reset – it's a safety device to save accidentally pressing Reset.

Boot your drive with APPLE1:, press X, take out APPLE1: and insert APPLE3:, type APPLE3:FORMATTER, enter the program and format some spare discs. Before finishing, put APPLE1: back in the drive and exit.

Now press F to enter the filer, transfer the files SYSTEM.APPLE, SYSTEM.PASCAL, SYSTEM.MISCINFO, SYSTEM.EDITOR and SYSTEM.FILER to one of the new discs from APPLE1:, and name the new disc EDIT:.

You now have a bootable disc with 125 blocks on for creating text files.

While in the filer, transfer APPLE2:SYSTEM . COMPILER to another of your new discs. This will be your COMP: disc.

Put APPLE1: back in the drive and exit the filer. Re-boot the system with your new disc EDIT: and create a simple program.

When finished, enter the filer and transfer it to COMP:. Put EDIT: in the drive and exit the filer, put COMP: in the drive, press C and follow the instructions to compile.

Later on you will want other files on this disc and life will become more complicated – there's enough disc-swapping as it is! If you grow to like Pascal, buy another drive.

There is a collision counter at location 234 and its use is well illustrated by the following program. – **Max Parrott**

```
10 HOME
20 GOSUB 500
30 HCOLOR=3
40 ROT=1: SCALE=1
50 HGR
60 POKE 232,0: POKE 233,96:
   REM POINT TO SHAPE TABLE
70 DRAW 1 AT 100,100: REM
   DRAW STATIC SHAPE
80 DRAW 1 AT PDL(0), PDL(1):
   REM DRAW MOVEABLE SHAPE
90 VTAB 22
100 CC=PEEK(234): REM LOOK
   AT COLLISION COUNTER
105 PRINT CC: REM DISPLAY
   COLLISION COUNTER
110 GOTO 50
500 REM SQUARE GENERATE
510 A=24576
520 READ B
530 IF B=999 THEN RETURN
540 POKE A,B
550 A=A+1
560 GOTO 520
570 DATA
   1,0,4,0,32,36,36,44,54,54,
   54,37,36,36,44,54
580 DATA
   54,54,37,36,36,44,54,54,54
   ,0,0,120,12,0,120,120,120
590 DATA 120,999
```