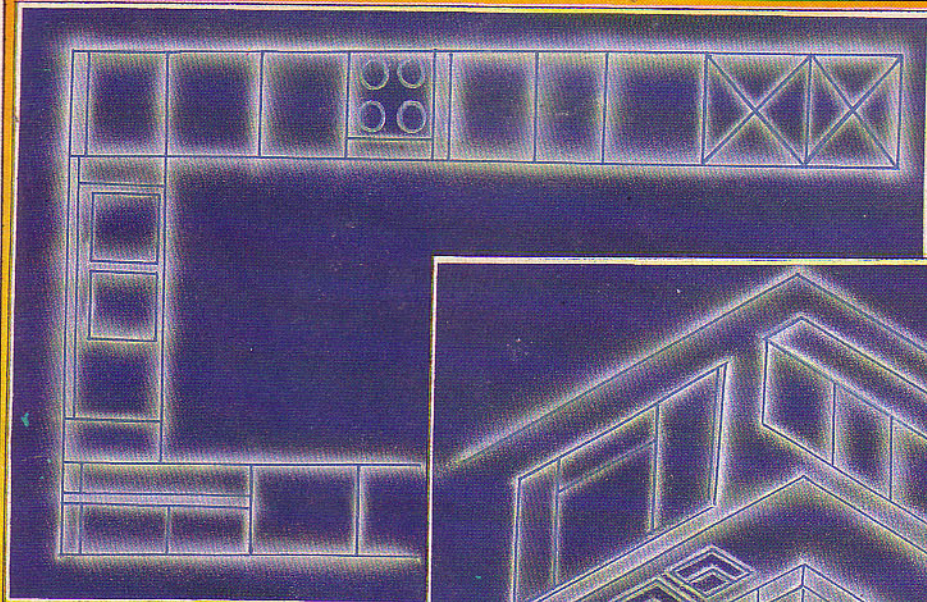




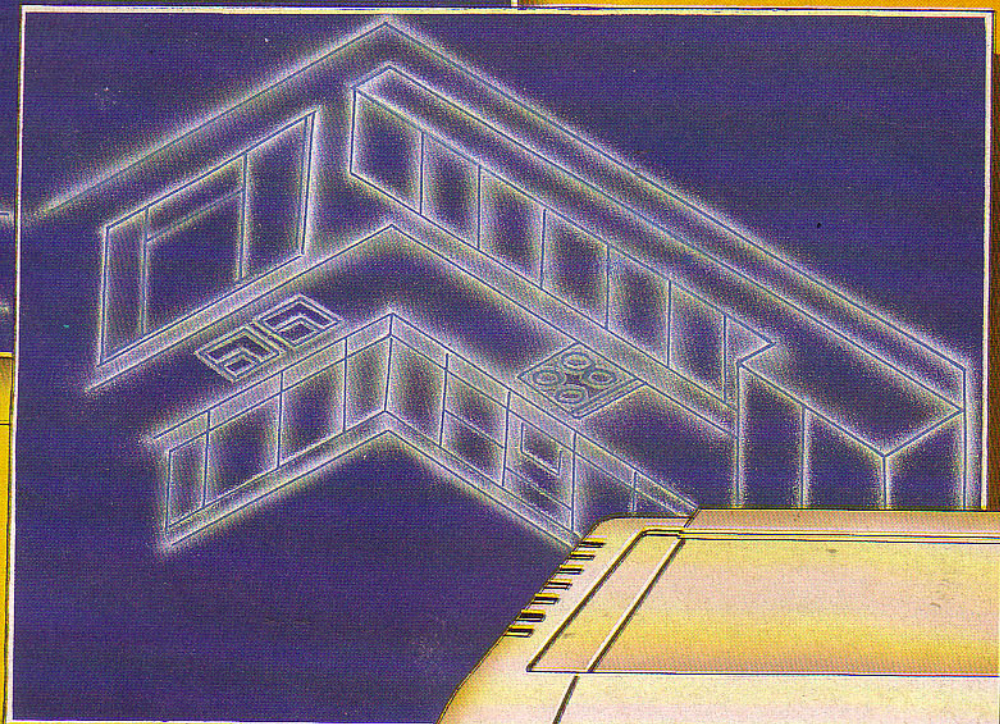
A Database Publication

apple user

Vol. 5 No. 9 September 1985 £1



**Apple cuts
kitchen costs**



**Home buying
made easy
— with
Appleworks**

**Dumping hi-res graphics
with a serial card**

**MacPublishing:
three software
packages reviewed**



Is Jazz what business has been waiting for?

News

● All the latest events in the world of Apple computing.

5

MicroLink

● This month's update on news from Britain's national electronic mail service.

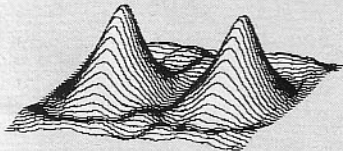
9

Pascal

● Stuart Bell continues the tutorial series with an explanation of units.

10

Graphics



● While Peter Gorry takes a well-earned rest, Max Parrott shows how to dump graphics to an Imagewriter.

17

Macintosh

● Will Jazz be music to your ears? Bill Hill gives his opinion on this important package.

21

Utility

● If you've ever tried to compare two versions of a program, you'll appreciate the usefulness of this routine from Richard Shann.

24



Volume 5
Number 9
September 1985

Spreadsheet

```

File: HOUSECALC          REVIEW/ADD/CHANGE          Escape: Main Menu
-----A-----B-----C-----D-----E-----F-----G-----
1:      HOUSE BUYING AND SELLING CALCULATOR
2:
3:
4:
5:SELLING PRICE FOR EXISTING HOME          45,000
6:BUYING PRICE                             68,000
7:
8:
9:DEDUCT REPAYMENTS ON EXISTING HOME
10:
11:      First mortgage
12:      Home improvement 1
13:      TOTAL REPAYMENTS
14:
15:GROSS SURPLUS FROM HOME
16:
17:ADD:
18:      First
19:      Savin
20:      Loan
21:      TOTAL
22:DEDUCT SELLING
23:
24:      Estat
25:      Stamp
26:      Other
27:      Other legal costs - selling      450
28:      Survey - new home                125
29:      Removal & sundry costs          400
30:      TOTAL SELLING/BUYING COSTS      3,155
31:
32:TOTAL FUNDS AVAILABLE FOR PURCHASE          67,345
33:
34:SURPLUS/SHORTFALL                          -655
35:
36:
37:
38:
    
```

● Can you really afford that dream home? John MacGibbon's Appleworks spreadsheet may provide the answer.

13

Review

● Bill Hill reviews three packages that provide page layout facilities. Is this the future of publishing?

29

Application

● Fast and efficient kitchen design with an Apple IIe based system.

32

Classifieds

● Bargain offers from your fellow Apple users.

34

Education

● Jenny Preece describes the rationale behind a package aimed at helping teachers select software.

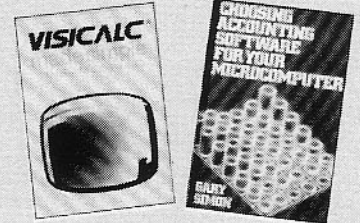
35

Programming

● A. Reicher provides the last word in bomb-proof input routines.

38

Books



● Before you rush out to the bookshop, read these reviews. Choosing accounting software and Visicalc are the subjects this month.

44

New Products

● All the latest ways to spend your money.

46

Fun & Games

● Skyfox, Wishbringer and Rescue Raiders provide the fun and games for our reviewers this month.

51

Feedback

● How to access two screens in Pascal, a Dublin BBS, the Data Protection Act and more.

55

Order form

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

58

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 MONREFG Quoting Ref. 79:MAG001 Teecom 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: Euopress 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.

© 1985 Database Publications Ltd. No material may be reproduced in whole or in part without written permission. 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)

Mac price stays

APPLE UK has denied it intends to lower the price of the Macintosh, despite widespread feeling in the trade that it must do so in order to compete with the Atari ST.

"We are holding the price at its current level", a spokesman told *Apple User*.

"If the price does drop it will be because of decisions taken by individual dealers and not because of any initiative by us".

The spokesman would not comment on American reports that Apple is planning to manufacture a super Big Mac for release later this year, though dealers contacted by *Apple User* said they had heard a 2mbyte Macintosh was in the pipeline.

SECRETS REVEALED

LEADING Apple II experts reveal their tricks of the trade in a new book from Nibble Publications appropriately called *Apple Secrets*.

The American publisher says readers will learn how to streamline program design, speed up Applesoft, create special graphics effects and control DOS.

More than 70 programs to type into the Apple are included, each with a line by line description of the programming logic. Articles range from helpful tips for Apple fun to more advanced techniques for improved programming.

The 220 page book costs \$19.95, with program disc \$29.95.

16 bit Apple on the way?

REPORTS are coming in from informed sources in the United States that Apple is about to launch its long-awaited upgrade of the IIe.

Although the project is still top secret, the new machine is believed to have a built-in 16 bit processor and the Motorola 68020 chip.

This was confirmed by Michael Murphy writing in the authoritative California Technology Stock Letter.

However Apple UK still refuses to confirm that an enhanced IIe was on its way. A

company spokesman opted for being noncommittal – the industry norm for companies on the eve of a new product announcement.

Meanwhile rumours of a new version of the machine are widespread in the trade as *Apple User* discovered when contacting leading retailers.

All said they would welcome such an enhancement of the Apple II to offset increased competition in the marketplace.

P & P's Colin Budgen said: "A 16 bit processor is an essential move for Apple to

make if it wishes to stay ahead of the competition.

"Unless such action is taken to improve the capabilities of the Apple II, I can only foresee a realistic future of six to 12 months".

The spokesman at Apple UK's headquarters said managing director David Hancock was sticking to his comment in last month's *Apple User* which promised future developments bringing the IIe into line with the IIc "so that powerful mouse-driven software will work on both computers".

Trade view on way Apple should go

THE Apple II has the muscle to survive the new onslaught of sophisticated competition from Atari, Commodore and others – provided the company brings out its big guns in support.

But the picture for the Macintosh is nowhere near as bright.

This is the message to Apple Computers from leading retailers of the company's products who were polled by *Apple User*.

Colin Budgen, dealer sales manager, of P & P Micro Distributors, summed up the opinions of most: "If the Apple II is to have a future the company must take several measures.

"It must upgrade to a 16 bit processor to regenerate business interest.

"There must be a massive price drop – as much as 50 per cent if possible – and UK marketing must change totally. The marketing for the Macintosh is terrible and for the Apple non-existent".

Other views from the industry:

John Buckingham, of Advanced Micro Products: "I don't think the Apple II is in much trouble up against the ST, Amiga and others, but certainly the Macintosh is.

"Although the new machines can't compete with the Mac's graphics output, its missing colour is a problem".

Tony Roberts, technical director of Peanut Computers: "The Apple II has a very good future in the small business and education sectors.

"It has a vast variety of software available for it, unlike its competitors, which is a very strong point in its favour. The new Atari and Commodore machines are better machines on the face of it but are down on the amount of software they can offer.

"However the price of the Macintosh has to come down if it is to survive against the Atari ST".

John Robertson, technical director of Cirtech: "In order to maintain its market share Apple must advertise in the UK.

"Apple could reduce the price of the Macintosh considerably, but it depends on the attitude of its marketing people. At present the machine is too expensive to compete with the Atari ST".

Norse ending for Lisa epic

TWELVE months and £9,500 after buying a Lisa 2/10 plus various peripherals and software, Stephen Prescott has admitted defeat in his battle to get it to function properly.

Prescott, director of a London firm of tax and investment consultants, bought the machine to help him monitor and analyse the stock market.

But program bugs, disc drive faults and other problems dogged him from start to finish. Even a replacement machine didn't help.

The first crash came the day the machine arrived and after only 50 numbers were fed into it. Thereafter Prescott's problems mounted despite help, advice and promises from his dealer and from Apple on both sides of the Atlantic.

In the end Apple changed the Lisa's name to the Macintosh XL and finally killed it off. Now Prescott is wondering how many other owners had similar problems with the machine.

"I'm very loyal to Apple - my firm has a network of Apple IIs we're very happy with - and that's probably why I persevered for so long trying to get the machine to do what it was supposed to do", he said.

"Fortunately the Lisa came with a money back guarantee, so all I will have suffered is frustration. I'm returning the machine more in sorrow than in anger. I really wish it had worked, but I suspect now that it was just a flawed product that Apple could not sort out".

His Lisa saga ended like a Norse epic. In true Viking fashion the machine's internal Winchester disc drive burned out.

"I hadn't actually planned a Viking funeral for it, but it seems a fitting way to go", he said.

Move into US market, software developers told



FORMER oilman Jim Mangles wants British developers of software for Apple computers to strike it rich in America.

Until a few months ago Mangles worked with computers in the US oil industry, building up relationships with dealers, distributors and publishers.

Now he wants to use his contacts to create a pipeline for UK and European authors and developers to break into the lucrative American market.

He has set up a company called Ewart Microsystems, based in Scotland, to act as consultant and agent for Apple software produced here.

"Over 80 per cent of the

world market for Apple II software is in the United States, and it's even higher for Macintosh", says Mangles.

"If you want your product to succeed, you can't ignore the American market. And the UK is producing software that should be making a huge impact there if it can be got across in the right way".

Eager

Mangles described his company as a sort of "estate agent" in America for British software. He says there are many distributors and publishers there eager to market good programs.

"It's not so difficult to find a publisher. The problem is finding the right one who will both

market your program effectively and pay a reasonable price for your efforts", he says.

"It's essential to establish good relationships with the right people and maintain them.

"That's an expensive, difficult, and time-consuming proposition for an individual or small software house - they have better things to do than use up limited resources that way".

Mangles is also offering a consultancy service about the US software market - advice on how products may have to be modified, help with the manual and packaging, assessment of sales potential, pricing philosophy, and where the current market gaps are.

Watch that Mac

THIEVES who stole a Macintosh from the offices of Popular Computing Weekly will have a hard job getting it off their hands.

Sunshine, the magazine's publishers, has circulated the computer's FG2110GM001 serial number to all dealers, clubs and magazines and are considering asking software companies to keep a lookout.

Said PCW's Duncan Scott: "There are a relatively small number of 128k standard Macs in the country - only about 10-15,000. Any sales, especially cheap ones, will attract attention.

"Eventually the computer will come into the hands of a dealer or retailer and police will be able to trace it back to the thief".

Fortunately, valuable software was left behind during the theft, in which an Apricot computer - no serial number this time - was also stolen.

Sunshine was able to carry out the administrative work usually done on the Macintosh on a borrowed machine.

The company began tightening up security after the break-in, but not early enough to prevent a second theft - this time of several home micros.

JAZZ HITS No.2 SPOT

JAZZ for the Macintosh has made it to second place in a top American software chart.

It is vying with another Lotus Development product - the Lotus 1-2-3 - for top position in the country's Softsell List.

For Lotus the news brought more celebrations.

These had started before Jazz's formal launch at Ronnie Scotts when it was announced 40,000-plus advanced orders had been placed for the product.

Then, bang on time, French and German versions of the package were released to an already-expectant European market.

Business brisker

BUSINESS is looking up at last for Apple hardware and software products manufacturer Advanced Logic Systems of California.

The company reported a profit of \$58,000 for the first six months of 1985 compared with a loss of \$231,000 for the same period last year.

President of ALS Grant Jasmin said he was pleased with the firm's performance "especially in light of the current industry-wide sales slump" but added that a recovery in the micro industry "is not likely to occur before the fourth quarter of 1985".

Blyth spirits

ONLY one month after launching the Omnis 3/Macintosh package in the US, Blyth Software reports that orders for the product have topped the \$1 million mark.

Production at Blyth's Suffolk headquarters has been stepped up to meet demand for the package, which is one of the first to offer networking facilities on the Macintosh.

Taking blue bits off the board

NAUGHTY words of a type that would make even a sergeant major blush are increasingly confronting Apple users who log-on to bulletin boards.

All over the UK systems operators are being forced to devote more and more of their time to erasing electronic graffiti.

The obscenity problem has been one that to date has baffled the industry. However according to the latest issue of *TeleLink* – a sister publication of *Apple User* – help may be at hand.

It takes a look at a new Naughty Words Editor which is currently being evaluated by MicroLink, the recently launched nationwide service for micro users.

The man in charge of the project is 39-year-old Tim Clarkson. He explained to *TeleLink* just how the Naughty Words Editor should work.

"You initially create a text file or glossary of naughty words or phrases", he is quoted, "so when these turn up in any message, the whole of the text

is pulled out and put in an abeyance file.

"Later the messages are checked over by the sysop to see whether it's safe for them to be released".

That's the theory behind it but in practice – according to *TeleLink* – it has been presenting more than a few problems.

One of the worries that faces Tim Clarkson is what to do with the species most cherished by birdwatchers, the tit.

"Used in the ornithological context, the word could in no way give offence", he says. "However once it becomes anatomical then eyebrows would understandably be raised.

"So you decide to err on the side of caution and classify 'tit'

as a word that might possibly offend.

"What happens then however is that all messages containing the word 'title' suddenly find their way into the abeyance file".

In order to counteract this, the MicroLink arbiter of good taste has created a text file of phrases – and not individual words.

"This removes part of the problem", says Tim. "We can rule that tit is left in as long as it has 'blue' or 'crested' in front of it but not 'big'."

Despite the current difficulties, the company behind MicroLink – Database Publications – believes that the Naughty Words Editor may well provide the answer.

APPLE has extended the upper limit of its Applecard credit card scheme to £2,500 – believed to be one of the highest levels ever offered to first-time buyers in UK commercial history. Previous limit for purchasers who met retailers' credit requirements was £1,500.

The new scheme is currently being tested in 23 of Apple's dealerships nationwide and, if successful, will be made available to the rest of the company's 250 outlets in the UK.

Apples to fill heritage gap

APPLE owners have been invited to help fill a gap in our national heritage caused by the likes of Henry VIII and Oliver Cromwell.

Because of these and other iconoclasts there is no British history of Nativity art as there is in most of the other European countries.

But that is something that archivists and historians Count and Countess Andrzej von Stauffer intend to correct.

They have organised the First British National Nativity Competition in an attempt to put Britain alongside the other countries of the world who already have a longstanding

tradition of Nativity-making.

It coincides with the Twelfth World Congress of Nativitists, which is hosted by a different country every three years and this year is at Innsbruck, Austria, in December.

The competition, in two parts, will be held at Westminster Cathedral on December 8 and at the Christian Resources Exhibition at the Horticultural Halls in London on February 8.

Count and Countess von Stauffer have been working closely with their international counterparts for eight years, as well as with British craftspeople, artists, photographers

and – during the past two years – computer graphics artists.

They have been helped by Epson UK, which was involved in the Christmas Archives Exhibition "Folk Nativities of the World" at the Barbican Centre last winter.

The Epson connection with the national Nativity competition is in the special category for the best computer-generated image of the Nativity.

Any part of the Christmas story may be represented, and entries will be judged on originality, content and approach.

Epson will judge this section of the competition and award a

prize to the winning computer artist.

"The winner in the computer category will also compete for the overall Best in Show trophy", Count von Stauffer told *Apple User*.

"And it is likely to be given a place in the British Nativity archives which will be going on tour to the United States and leading Commonwealth countries shortly".

Apple owners who want to enter the competition should write to National Nativitists Competition, Christmas Archives, 64 Severn Road, Cardiff CF1 9EA enclosing a 24p stamp.

NEWSLETTER

£1½m computer to the rescue

THE phenomenal growth of MicroLink has hastened the purchase of additional computer power by Telecom Gold.

Demand for the new electronic mail service has been such that despite the tremendous processing power of a Prime computer, at certain times of the day users have been inconvenienced by motorway-like congestion caused by the large volume of traffic.

Since it started, MicroLink has had to share its computer with the somewhat verbose members of the European Parliament. The traffic jam

worsened as subscribers from all over Britain and Europe and as far away as Australia and Japan began logging on in increasing numbers.

With MicroLink growing at more than four times the predicted rate, the result left it no option but to request a separate computer for its exclusive use.

Telecom Gold has come to the rescue and on September 7 MicroLink will be moving to its own £500,000 dedicated system – much to the relief of its own users and the Euro MPs.

The move will enable

MicroLink to provide an increasing number of exciting facilities, together with a response time described by its systems manager, Colin Rogerson, as “super quick”.

Telecom Gold officials have been staggered by what they describe as “the phenomenon of a specialist service growing so big in such a short time”.

Rogerson believes he knows exactly why MicroLink has taken off so dramatically.

“It’s more friendly than other electronic mail services, it’s informative, and it’s fun to use”, he says.

It's all systems go . . .

THE ever-ready Help Line came to the assistance of a distinguished early MicroLink subscriber, Conservative MP for Acton Sir George Young.

He mailed to say: “The screen does not scroll when it is in the Telecom Gold mode. The new lines simply superimpose on the old, making it very difficult to read messages.”

“I have a BBC Micro with a Telemod 2 modem and a Micronet 800 ROM. To access Telecom Gold I have to generate a new Return signal. What am I doing wrong?”

What Sir George was doing wrong was trying to access the service using Micronet 800 software.

Help Line was able to give him two options – either get a Commstar ROM, which has both Prestel and terminal emulation, or keep the Micronet 800 ROM and use it with a terminal emulation program such as Termi.

Shortly afterwards Sir George was able to report “all systems go” at his end of the system.

Popular abroad

NOT only is MicroLink the great new national electronic mail service – it’s also making a name for itself on the international scene.

On the Continent there are already subscribers in Belgium, France, Luxembourg, Spain, Switzerland and West Germany.

Further afield are its members in Saudi Arabia, Australia, New Zealand and Japan.

There are even a couple of subscribers stationed at British Forces bases in

Germany, in addition to those in the UK and the Republic of Ireland whose numbers increase daily.

Why is MicroLink so popular abroad?

Says one happy customer: “It’s a very good way of sending information by the international PSS system, it’s faster and more economical than telex, and it’s portable.

“I can take my lap-held computer just about anywhere and still be in touch with MicroLink – at any time of the day or night”.

Showing 'em how

MICROLINK will be notching up another first when it goes on-line from the Electron & BBC Micro User Show in Manchester.

A continuous demonstration of the new service will be held at UMIST from September 27 to 29, with experts on hand to reveal the full potential for users.

Stories about the show will be transmitted live over MicroLink’s own electronic news pages during the three-day event.

MicroLink forges commercial ties

NEW commercial ties between Britain and Japan are being forged by MicroLink’s speed and efficiency.

For several years Bristol electronics engineer Jeff Gearing has been UK correspondent of a Japanese motoring magazine, regularly sending his news reports by mail on floppy discs to its editor, Yuichi Ishikawa.

Other than expensive long-distance phone calls, the two men had to rely on the five-day-minimum airmail service to keep in touch.

Until MicroLink, that is.

Now both Gearing and Ishikawa are subscribers of the fast-growing international mail service, and news about Britain’s motor industry gets to Japan in

seconds rather than days.

MicroLink has paid off for them in another way – high-speed two-way exchange of up-to-the-minute business information that can be sold as a service to commercial concerns.

And this has led to a further profitable spin-off.

For some time Gearing and Ishikawa had been

aware that there was a demand in Japan for luxury European goods, and in Britain for Japanese-made models.

MicroLink is now providing the fast, low-cost medium for import/export orders resulting in an increasing flow of Wedgwood pottery from Bristol and radio-controlled models from Tokio.

Units - the building blocks for extending Pascal's applications

THE concept of a unit is not one that is found in Standard Pascal. Describing the development of UCSD Pascal - on which Apple Pascal is based - the UCSD Pascal Handbook* notes: "Units provided a means of packaging

**The UCSD Pascal Handbook, by Randy Clark and Stephen Koehler, published by Prentice-Hall, is THE authoritative reference book on UCSD Pascal. It is not a tutorial book, but is ideal for those who have learned the fundamentals of Apple Pascal.*

code in a modular fashion, so that application libraries could be built".

This sums up the purpose of units rather well - this month we shall look closely at the advantages of units, and see how one particular unit can be added to the system library.

Once you have written a few Pascal programs, you soon find that they share many common routines.

Obvious examples of such

"building blocks" are robust input routines, or those to produce neat output on the screen.

Others might search a file or perform operations on arrays.

To avoid retyping the procedures and functions each time, you could use the "copy from file" operation in the editor to pull the required routines into your program.

An alternative approach is to keep such frequently-used code

in a file, and use the "Include" directive to tell the compiler to treat the procedures as part of the main program.

Both these solutions have the effect that on every occasion that the program is compiled, the standard procedures are also compiled. This wastes time, and also disc space, since the same procedures will be repeated in many different programs.

By using a unit, we can compile these standard routines separately from the programs that will use them, so that they are only compiled once.

Supplied with the Apple Pascal system are a number of pre-written units. An example of these is the turtlegraphics unit, which allows us to write complex graphics programs.

As page 90 of the Language Manual shows, you simply use the statement "uses turtlegraphics" at the top of your program, and then all the graphics commands can be used.

The code to perform these operations is stored in the file

```

(*$S+*)
unit getcharandint;    intrinsic code 25;    (*must use to compile units *)
                                (*specifies segment number *)

interface
                                (*this is the stuff that *)
                                (*calling programs can 'see' *)
    type chs=string[80];
    function getch(validchs:chs):char;    (* returns the valid character *)
    function getint(x,y,min,max:integer): integer; (* returns valid integer *)

implementation

function getch;                                (*gets a character, which must *)
var ch:char;                                    (*be one of those in 'validchs'*)
    st:string[1];
begin
    st:= ' ';
    repeat
        read(keyboard,ch);                    (*keyboard does not echo to screen*)
        st[1]:=ch                             (*put in string, so can use pos *)
    until pos(st,validchs) <> 0;              (*check for ch in validchs *)
    write(ch);                                (*valid, so echo to screen *)
    getch:=ch
end;

function getint;                                (* gets integer in the range min..max*)
const cleol=29;                                (* char to clear to end of line*)
var    n,i:integer;
        error,neg:boolean;
        ch:char;
        r:real;
begin
    repeat
        gotoxy(x,y);
        write(chr(cleol));
        neg:=false;
        ch:=getch('0123456789-+ ');          (*first char is digit or sign *)
                                                (* CR returned as space *)
        if ch in ['+', '-'] then
            begin
                neg:= (ch = '-');            (* ie neg is true if ch = '-' *)
                ch:=getch('0123456789 ');   (* next chars must be digits *)
            end;
        r:=0;
        while not eoln(keyboard) do        (* ie get chars until CR typed *)
            begin
                if (ch <> ' ') then          (* if not CR, then add to no. *)
                    begin
                        n:=ord(ch)-ord('0'); (*convert char to integer and *)
                        r:=r*10 + n          (*add to number (r) *)
                    end;
                ch:=getch('0123456789 ');   (* now get next char *)
            end;
        write(n);
        if neg then r:=-r;
        write(n);
        error:= ((r < min) or (r > max));   (* check for number being in range*)
        if error then write(chr(7));        (* bell*)
                else i:=trunc(r)
            (* keep going if out of range *)
        until not error;
        gotoxy(x,y);
        write(i:6,chr(cleol));              (* re-display, right justified *)
        getint:=i
    end;
begin (* dummy main program *) end.

```

Listing 1

SYSTEM.LIBRARY, and linked into your program when it is run.

Let us now look at a typical unit – Listing I. This consists of two functions, “getch” and “getint”.

The first is a routine to get one character from the keyboard – it must be one of a set of valid characters which are passed into the function.

The second one goes to a particular screen location, and then inputs an integer, accepting it only if it is within a range specified by the calling program.

Most of the unit consists of conventional Pascal statements. Let us consider those that are new to us:

We must start a unit with the (*\$\$*) directive to the compiler to warn it to allow more working space for the compilation.

The first line proper of a unit is just like the “program” line of a program, except that “unit” is used instead.

The “intrinsic code 25” tells the compiler which segment number to use. Don’t worry about the precise meaning of this – we shall see how it is used later.

The “interface” section contains that information which will be accessible to programs which call the functions. In our case, it is simply the type declaration for the string of valid characters, and the declarations of the two functions.

The implementation part contains the actual code for the functions. Note that we do not re-declare the parameters to the functions – only the names are used, as the full declaration was specified in the “interface” section.

At the end is a dummy main program. It could be used to contain initialisation code to be executed once when the calling program starts to run.

We shall now add the two functions “getch” and “getint” into the system library, so that any program which we write can use them.

The first stage is to compile the unit in Listing I, and save it, perhaps in a file called GETINT.CODE.

Now copy the file LIBRARY.CODE from your Apple3: disc to the disc that you

normally use in drive 4:. This is the librarian program which lets us build and change libraries of useful routines.

Now make sure that you have available the following files: SYSTEM.LIBRARY, GETINT.CODE, LIBRARY.CODE, and that you have a space of at least 38 blocks on a disc that is in a disc drive.

From the Command level, e(Xecute LIBRARY.

To the question “Output code file”, respond “new.library”. All filenames may be preceded by the volume, for example #5:new.library.

We shall first put into our new library the contents of the standard library, so to the prompt “Link code file”, enter “system.library”, and to the request “Slots...”, reply with the = sign so that all of the old library will be used.

The system will inform us

By STUART BELL

that it is “copying all slots”, and it will show each unit being copied, ending with APPLE-STUFF. To the next “Slots...” prompt, reply “N” to indicate that you wish to copy units from a new file.

To the question “Link Code File”, we respond with the name of the code file containing our new Unit, ie GETINT, and the librarian will display the map of the unit, with slot number 1 containing GETCHAR, using segment 25, and being 350 words long.

To add this unit to the new library, reply 1 to the first prompt, then 7 to the “slot to link into” request. Our new unit will now be added after the old units.

Finally, type Q to quit the librarian, replying to the Notice prompt with something like “Copyright Apple Computer Inc & Apple User 1985.”

You now have two libraries on disc, SYSTEM.LIBRARY and NEW.LIBRARY. To use the new one, rename NEW.LIBRARY as SYSTEM.LIBRARY, which will delete the old one, and the task is complete.

The effect of the above process is that we can now use

```

program testunit;
uses getchandint;

var i:integer;
    j:char;

begin
  write(chr(12));
  repeat
    gotoxy(8,3);
    write('please enter a number, 0-999 ');
    i:=getint(33,3,0,999);
    gotoxy(8,24);
    write('OK? ');
    j:=getch('YyNn');
  until j in ['y','Y'];
end.

```

Listing II

“Getch” and “Getint” in our programs, just like the turtlegraphics or Applestuff statements.

Our programs must simply contain the statement “uses getchandint” after the “program” line, and the system will

very large.

I avoid this problem by having several system discs, each with different libraries on them – some with graphics, some for hardware development, and so on.

Version 1.2 of Apple Pascal allows users of 128k Apple II systems to keep several libraries of units, and lets programs use units from different libraries.

On all Apple Pascal systems, the LIBMAP utility on the Apple3: disc will show you what is in any library.

Next month, we shall look at the process of linking Pascal programs and assembly language routines, and then consider the use of such routines in units – the turtlegraphics routines are so fast simply because they are written in 6502 code.

look after everything else for us.

Listing II shows a very simple program that makes use of our new unit.

Serious applications programmers build large libraries of such “building blocks”. They save much programming time, because the units only have to be de-bugged once. They also reduce compilation times.

The only problem is that the SYSTEM.LIBRARY tends to get



t When Applesoft executes a FOR...NEXT loop for which no step value is specified it assumes a step of +1.

If the loop is to count downwards – if the starting loop value is greater than the final value – a “Step -1” must be added.

Sometimes a loop may be required to count up or down if either of the two loop parameters are variables. In this case use may be made of the Applesoft function SGN, which returns a value of +1 if

its argument is positive and -1 if it is negative.

To make the loop:

FOR I=A TO B

work in both directions it must have a step value of +1 if B > A and a value of -1 if B < A. This is done by changing it to:

FOR I=A TO B STEP SGN(B-A)

If a step value other than 1/-1 is needed just multiply the required step by the SGN function, for example:

FOR I=A TO B STEP SGN(B-A)*.25

Gerard Manning

By JOHN MacGIBBON

How Apple's magic carpet can help you move into the home of your dreams

APPLE'S integrated Apple Works program has become one of the world's hottest software products. It is also convincing a lot of Apple II+ owners that it's time to upgrade to a IIe or IIc.

I've had the program for several months now, and I fell quickly in love with its word processor and database features. However for some time I had no real need to use the spreadsheet, and gave it little more than cursory attention.

That is, until I got embroiled in the tricky business of selling my home and trading up to a new property. Finally I had an incentive to extend my spreadsheeting experience beyond Multiplan and into AppleWorks.

Selling and buying at the same time has more traps than just buying a house, and as the various offers and bids fly around it can be handy to be able to make quick assessments of your financial situation, so you can look at things like:

- What estate agents' commissions and stamp duties will I have to pay?
- What "profit", surplus or equity will I get from sale of my present house, after deducting the mortgage I owe on it?
- Given the above equity, if I add my savings and take into account rake-offs by solicitors and estate agents, what level of new mortgages and other loans will I need to secure my dream home?
- Can I still afford my dream home if ungrateful prospective buyers of my existing property aren't offering what I reckon it's worth? Can I juggle mortgages, savings, overdrafts and so on, to achieve my goal?

These are just some of the questions any prospective home buyer and seller ponders, usually on the back of an envelope. It's a lot quicker with a spreadsheet, which can automatically calculate such figures as estate agent and stamp duty charges, which vary depending on selling and buying price levels.

Now that I had some

motivation to open the AppleWorks manual, setting up the spreadsheet was straightforward, an evening's work. This was a relatively simple model as spreadsheets go, but it was noticeably easier with AppleWorks than it would have been with Multiplan.

Of course much of the hard work had already been done — like sussing out my financial situation, discovering typical charges for solicitors and property valuers, and finding out formulae for agents' fees and stamp duty.

Living as I do in antipodean isolation in nuclear-free New Zealand, the UK conditions and percentages for agents' fees, stamp duties and the like were unknown to me. Fortunately the folks at *Apple User* gave me appropriate rates and I was also able to pick the brains of a British nuclear refugee I work with. (He's been here a couple of years now, so please make allowances if some figures are a trifle out of date — the general principles should hold.)

The accompanying table shows a typical workout for the spreadsheet. While AppleWorks has been used, the general principles could be applied to any other spreadsheet, such as Multiplan or Visicalc.

AppleWorks has the rather nice feature of letting you keep typing merrily across column boundaries as long as you don't impinge on information in existing columns to your right. So the first step is to type in everything except the figures, as per the example.

Later on, when you've got the spreadsheet working, you could add or delete categories, as appropriate.

Next you might as well fill in some notional figures. I suggest you start with the figures in this example, because then you will quickly see if your spreadsheet is working properly.

However don't fill in figures yet for adding the following categories — total repayments, total additions, estate agent commission, stamp duty, total

selling/buying costs, total funds available and surplus/shortfall. Each of these categories calculates its own figure, using formulae you provide.

The easiest formulae are those that simply add up a few figures. For instance the formula for total repayments is `@SUM(E11...E12)`.

First put your cell pointer on cell E13. Type `@SUM(`. This will appear on the line underneath the spreadsheet page. Now move the cell pointer to cell E11. Press Return, you'll hear a beep, and E11 will join the formula at the bottom of the page.

Now type three periods, move the cell pointer to E12 and press Return again. E12 will join the formula at the bottom of the page. Add in a right-hand bracket, and the formula is complete. As soon as you either press Return, or move the cell pointer to another cell, the total for the cells E11 and E12 will appear magically in cell E13.

Most other formulae for totals are calculated in a similar way. However "total funds available..." has to perform a subtraction, as well as two additions. The formula is `@SUM(F15+F20-F30)`. The formula for surplus/shortfall also has a subtraction, `@SUM(G32-F6)`.

Estate agents' commission is normally two per cent of selling price. To enter the formula, put the pointer on cell E24 and type `@SUM(F5*.02)`. Cell F5 is where you entered your selling price, at the top of the spreadsheet.

Stamp duty on your new house purchase is a little trickier. You pay no duty if the house costs less than £30,000. If it costs more than that you pay one per cent of the total buying price.

The spreadsheet formula, which will be entered into cell E25, makes use of the IF function. ("If the house costs more than £30,000, pay one per cent; if it costs less than £30,000, pay nothing"). Look at the bottom line of the example

SPREADSHEET

to see the formula. Start typing at the @ sign.

Now you should tell AppleWorks to present all figures in "commas" format as per the example. The quickest way to do this is to take your pointer to the top of the page and press Open-Apple-L. From the options presented, select block, and follow the instructions to highlight your complete spreadsheet. Pressing Return will reveal a further set of options. Choose commas, press Return, and hey presto, you've

got commas everywhere.

Your spreadsheet is now essentially complete. There is just one thing left to do. You should protect all the cells that contain formulae, otherwise someone may accidentally enter figures in these cells and destroy the formulae.

Put the pointer on the cell to be protected and press Open-Apple-L. Then make the following choices as they are offered to you - "entry", "protection", and "nothing".

At this point the calculated

figures in your spreadsheet should be the same as in the example. If they are, things are working OK, and it's time to start entering your own figures.

In the example, you are selling for £45,000, and buying for £68,000. Unfortunately the spreadsheet says you're going to miss out by £655. So you might have to push for a higher price on your existing house, cut your suit to fit your cloth, or touch Uncle Maurice for a tad more. (And add a bit more again, because there are always some

extras you never thought of!)

A useful addition to this spreadsheet was the mortgage repayments calculator included on the loan schedule file on the samples disc that comes with the AppleWorks program.

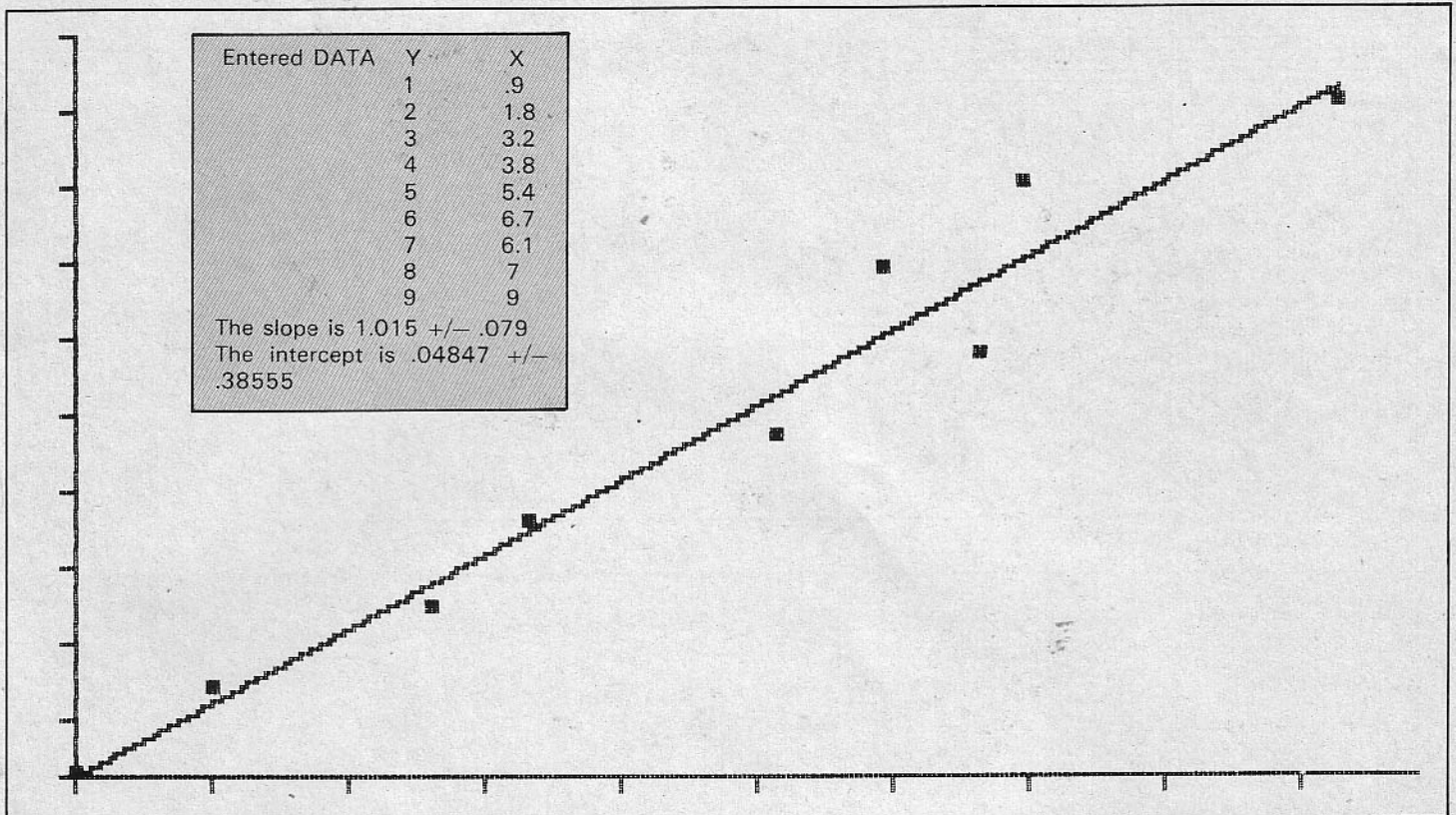
Use the copy function to transfer the few rows shown in my example.

The AppleWorks sample disc also has spreadsheets for personal budgeting and calculation of net worth. With suitable modification these can also help players in the real estate stakes.

File: HOUSECALC	REVIEW/ADD/CHANGE	Escape: Main Menu
A-----B-----C-----D-----E-----F-----G-----H-----		
1:	HOUSE BUYING AND SELLING CALCULATOR	
2:		
3:		
4:		
5:	SELLING PRICE FOR EXISTING HOME	45,000
6:	BUYING PRICE	68,000
7:		
8:		
9:	DEDUCT REPAYMENTS ON EXISTING HOME	
10:		
11:	First mortgage	10,500
12:	Home improvement loan	2,000
13:	TOTAL REPAYMENTS	12,500
14:		
15:	GROSS SURPLUS FROM HOUSE SALE	32,500
16:		
17:	ADD: First mortgage	28,000
18:	Savings - building society	5,000
19:	Loan from Uncle Maurice	5,000
20:	TOTAL ADDITIONS	38,000
21:		
22:	DEDUCT SELLING/BUYING COSTS:	
23:		
24:	Estate agent commission	900
25:	Stamp duty	680
26:	Other legal costs - buying	600
27:	Other legal costs - selling	450
28:	Survey - new home	125
29:	Removal & sundry costs	400
30:	TOTAL SELLING/BUYING COSTS	3,155
31:		
32:	TOTAL FUNDS AVAILABLE FOR PURCHASE	67,345
33:		
34:	SURPLUS/SHORTFALL	-655
35:		
36:		
37:	=====	
38:		
39:	MORTGAGE PAYMENTS CALCULATOR	
40:		
41:	Loan Amt	28000
42:	Pmt/Yr	12
43:	Total Yrs	25
44:	Interest	12 %
45:		.12 Decimal
46:	Interest	12.68 % (APR)
47:	Payments	295
48:		
49:	=====	
50:		
51:		
52:		
E25: (Value, Layout-C0, Protect-N) @IF(F6>30000, (F6*.01), (0))		
Type entry or use @ commands		
e-? for Help		

A typical workout for the spreadsheet



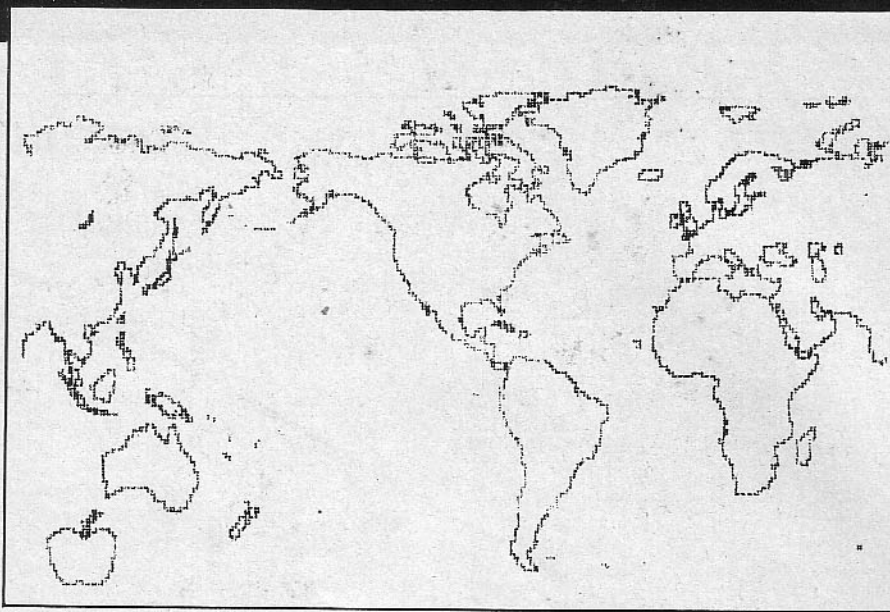


Example of a double size graph in normal mode with input data inset

```

0800      1 ;Imagedump, a utility for graphics
0800      2 ;dumps on the ImageWriter from
0800      3 ;the Apple II, II+, and IIe
0800      4 ;
0800      5 ;
0800      6 ;The syntax is CALL nnnn,A,X (any order for A and X, or even absence)
0800      7 ;where nnnn is the start address
0800      8 ;A is 1 or N where 1 signifies
0800      9 ;inverse and N normal dump and
0800     10 ;X is 1,2, or 3 for the hires pages
0800     11 ;
0800     12 ;
006F     13 STRINGST EP2 $6F
0073     14 HIMEM EP2 $73
00B1     15 CHRGET EP2 $B1
00B7     16 CHRGET EP2 $B7
009B     17 LOWTR EP2 $9B
02F0     18 BLOCK EQU $2F0
02FB     19 COUNTER EQU BLOCK+8
02F9     20 LO EQU COUNTER+1
02FA     21 HI EQU LO+1
02FB     22 IMFLAG EQU HI+1
02FC     23 HPAG1 EQU IMFLAG+1
02FD     24 BLKCOUNT EQU HPAG1+1
02FE     25 OPTION EQU BLKCOUNT+1
02FF     26 ADJUST EQU OPTION+1
E07D     27 ISLETC EQU $E07D
0412     28 ERROR EQU $0412
0B3A     29 BASPRINT EQU $0B3A
FD0D     30 COUT EQU $FD0D
0800     31 ;
0800     32 ;
9460     33      ORG $9460
9460     34      OBJ $1000
9460     35 ;
9460     36 ;
9460 A9 7A     37      LDA #START
9460 A0 94     38      LDY #START
9471 85 73     39      STA HIMEM
9473 85 6F     40      STA STRINGST
9475 84 78     41      STY STRINGST+1
9477 84 74     42      STY HIMEM+1
9479 60       43      RTS
947A A9 00     44      START LDA #0
947C 8D FB 02  45      STA IMFLAG
947F 8D FE 02  46      STA OPTION
9482 8D FF 02  47      STA ADJUST
9485 A9 20     48      LDA #20
9487 8D FC 02  49      STA HPAG1
948A 20 67 00  50      JSR CHRGET
948D F0 2E     51      BEQ NOWGO
948F 20 B1 00  52      GET  JSR CHRGET
9492 F0 29     53      BEQ NOWGO
9494 B0 11     54      BCS LETTER
9496 29 03     55      AND #3
9498 F0 0A     56      BEQ ERR
949A 0A       57      ASL
949B 0A       58      ASL
949C 0A       59      ASL
949D 0A       60      ASL
949E 0A       61      ASL
949F 8D FC 02  62      STA HPAG1
94A2 D0 EB     63      BNE GET
94A4 4C 12 04  64      ERR  JMP ERROR
94A7 20 7D ED  65      LETTER JSR ISLETC
94AA 98 E3     66      BCC GET
94AC C9 44     67      CMP #'D'
94AE D0 03     68      BNE OK
94B0 8D FE 02  69      STA OPTION
94B3 4A       70      OK  LSR
94B4 90 D9     71      BCC GET
94B6 A9 FF     72      LDA #FF
94B8 8D FB 02  73      STA IMFLAG
94BB D0 D2     74      BNE GET
94BD A2 0B     75      NOWGO LDX #1
94BF 8D FB 95  76      *1  LDA DN,X
94C2 20 ED F0  77      JSR COUT
94C5 CA       78      DEX
94C6 10 F7     79      BPL #1
94C8 A9 00     80      LDA #0
94CA 8D FB 02  81      STA BLKCOUNT
94CD AD FD 02  82      MAINLP LDA BLKCOUNT
94D0 C9 18     83      CMP #24
94D2 D0 0C     84      BNE CONTINUE
94D4 A2 03     85      LDX #3
94D6 8D FC 95  86      *2  LDA OFF,X
94D9 20 D0 95  87      JSR PRINT
94DC CA       88      DEX
94DD 10 F7     89      BPL #2
94DF 60       90      RTS ;Back to BASIC
94E0 0A       91      CONTINUE ASL
94E1 0A       92      ASL
94E2 0A       93      ASL
94E3 1B       94      CLC
94E4 48       95      FWA ;CALCULATE BASE ADDRESS
94E5 29 C0     96      AND #C0
94E7 8D F9 02  97      STA LO
94EA 4A       98      LSR
94EB 4A       99      LSR
94EC D0 F9 02 100     ORA LO

```



Example of a normal size dump

```

94EF 8D F9 02 101 STA L0
94F2 68 102 PLA
94F3 8D FA 02 103 STA HI
94F6 0A 104 ASL
94F7 0A 105 ASL
94F8 0A 106 ASL
94F9 2E FA 02 107 ROL HI
94FC 0A 108 ASL
94FD 2E FA 02 109 ROL HI
9500 0A 110 ASL
9501 6E F9 02 111 ROR L0
9504 AD FA 02 112 LDA HI
9507 29 1F 113 AND #1F
9509 0D FC 02 114 ORA HPAG1
950C 8D FA 02 115 STA HI
950F A2 05 116 LDX #5
9511 2C FE 02 117 BIT OPTION
9514 50 05 118 BVC 1
9516 20 C6 95 119 JSR DOUBLE
9519 30 09 120 BMI 3
951B 8D E4 95 121 LDA GPRINT,X
951E 20 D0 95 122 JSR PRINT
9521 CA 123 DEX
9522 10 F7 124 BPL 1
9524 AD F9 02 125 LDA L0
9527 85 9B 126 STA LOWTR
9529 AD FF 127 LDY #FF
952B C8 128 NEXTLP INY
952C C0 28 129 CPY #40
952E D0 25 130 BNE CONT1
9530 A9 00 131 LDA #00
9532 20 D0 95 132 JSR PRINT
9535 A9 0A 133 LDA #A
9537 20 D0 95 134 JSR PRINT
953A 2C FE 02 135 BIT OPTION
953D 70 05 136 BVS CONT2
953F EE FD 02 137 INC BLKCOUNT
9542 D0 89 138 BNE MAINLP
9544 AD FF 02 139 LDA ADJUST
9547 49 40 140 EOR #40
9549 8D FF 02 141 STA ADJUST
954C F0 F1 142 BEQ CONT3
954E A2 05 143 LDX #5
9550 20 C6 95 144 JSR DOUBLE
9553 A0 00 145 LDY #0
9555 A2 07 146 LDX #7
9557 9A 147 TXA
9558 0A 148 ONE ASL
9559 0A 149 ASL
955A 6D FA 02 150 FIRST ADC HI
955D 85 9C 151 STA LOWTR+1
955F B1 9B 152 LDA (LOWTR),Y
9561 9D F0 02 153 STA BLOCK,X
9564 CA 154 DEX
9565 10 F0 155 BPL 2
9567 A9 01 156 LDA #1
9569 8D FB 02 157 STA COUNTER
956C 2C FE 02 158 BIT OPTION
956F 70 16 159 BVS TWO
9571 A2 07 160 LDX #7
9573 5E FD 02 161 LSR BLOCK,X
9576 2A 162 ROL
9577 CA 163 DEX
9578 10 F9 164 BPL 5
957A 4D FB 02 165 EOR INVLG
957D 20 D0 95 166 JSR PRINT
9580 DE FB 02 167 ASL COUNTER
9583 10 EC 168 BPL 4
9585 30 A4 169 JMP BMI NEXTLP
9587 2C FF 02 170 TWO BIT ADJUST
958A 70 1C 171 BVS SECOND
958C A2 03 172 LDX #3
958E 5E FD 02 173 LSR BLOCK,X
9591 08 174 PHP
9592 2A 175 ROL
9593 2B 176 PLP
9594 2A 177 ROL
9595 CA 178 DEX
9596 10 F6 179 BPL 5
9598 4D FB 02 180 EOR INVLG
959B 20 D0 95 181 JSR PRINT
959E 20 D0 95 182 JSR PRINT
95A1 0E FB 02 183 ASL COUNTER
95A4 10 E6 184 BPL 4
95A6 30 83 185 BMI NEXTLP
95A8 A2 07 186 LDX #7
95AA 5E FD 02 187 LSR BLOCK,X
95AD 08 188 PHP
95AE 2A 189 ROL
95AF 2B 190 PLP
95B0 2A 191 ROL
95B1 CA 192 DEX
95B2 E0 D3 193 CPX #3
95B4 D0 F4 194 BNE 7
95B6 4D FB 02 195 EOR INVLG
95B9 20 D0 95 196 JSR PRINT
95BC 20 D0 95 197 JSR PRINT
95BF 0E FB 02 198 ASL COUNTER
95C2 10 E4 199 BPL SECOND
95C4 30 BF 200 BMI JUMP
95C6 BD EA 95 201 LDA DPRINT,X
95C9 20 D0 95 202 JSR PRINT
95CC CA 203 DEX
95CD 10 F7 204 BPL DOUBLE
95CF 60 205 RTS
95D0 48 206 PRINT PHA
95D1 AD 99 C0 207 LDA #C099 ;STORE CHAR
95D4 29 10 208 AND #10 ;SL0T 1
95D6 F0 F9 209 BEQ 1 ;Wait until ACIA Transmit register is empty,
95D8 AD 99 C0 210 LDA #C099 ;P.54 of SSC Manual
95DB 29 40 211 AND #40 ;SL0T 1
95DD D0 F2 212 BNE 1 ;CHECK DATA STROBE
95DF 68 213 PLA
95E0 8D 9B C0 214 STA #C098 ;RECOVER CHAR
95E3 60 215 RTS ;SL0T 1
95E4 30 38 32 216 GPRINT ASC '08206'
95E7 30 47
95E9 1B 217 HEX 1B
95EA 30 36 35 218 DPRINT ASC '04506'
95ED 30 47
95EF 1B 219 HEX 1B
95F0 0D 220 ON HEX 0D
95F1 36 31 54 221 ASC '61T'
95F4 1B 222 HEX 1B
95F5 6E 223 ASC 'n'
95F6 1B 20 00 224 HEX 1B2000
95F9 5A 225 ASC 'Z'
95FA 1B 0D 226 HEX 1B0D
95FC 0D 227 OFF HEX 0D
95FD 63 228 ASC 'c'
95FE 1B 0D 229 HEX 1B0D
9600 230 END

```

***** END OF ASSEMBLY

A FEW weeks ago Majid, the engineer from my friendly neighbourhood Apple dealer, came to see me bearing a particular goody for which I had been waiting with eager anticipation — a 512k upgrade for my Mac — the official Apple version.

I had already added Drive 2 to my system, so the substitution of the new 512k RAM motherboard turned the bottom-line 128k, single-drive machine with which I'd started my love affair with the Mac into a powerful beast which no-one, even an IBM robot, could fail to take seriously.

I mention all that because much of what has happened since is coloured by my experiences in using the Big Mac, and it has a lot to do with my reaction to **Jazz**, the new integrated package launched by Lotus with the expected song-and-dance a few weeks ago.

At first sight, Jazz is the answer to a maiden's prayer — a five-in-one, fully-integrated software package which combines word processing, database, spreadsheet, graphics and communications in the one powerful program.

Coming from Lotus, who devised 1-2-3, the package which for so long reigned supreme in the business software market, it was expected that Jazz would lift the Mac into a whole new arena, and make the business community — who tend to see life through Big Blue-coloured spectacles — sit up at last.

I've now had a couple of weeks to put Jazz through its paces. It's good, no doubt about that — despite what appears to be at least one bug in it.

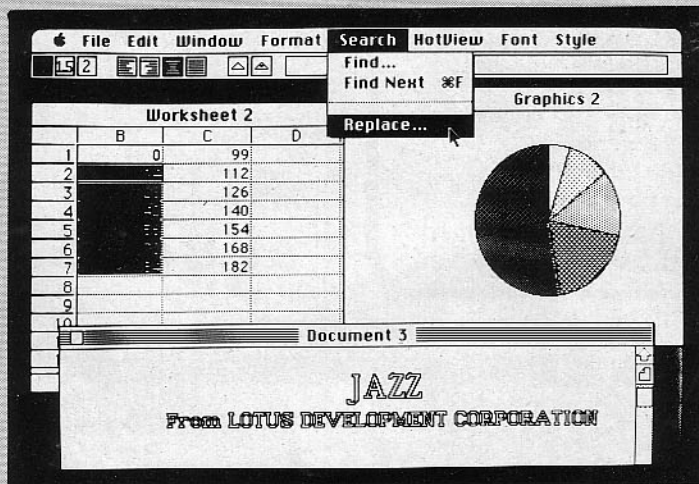
But it may have missed the boat by taking so long to appear and not being quite good enough, for reasons I'll outline later.

Jazz is beautifully packaged, and comes with a primer, a handbook, a reference book, a lot of smaller pamphlets, a holder for program and data discs, and a very smart carrying-folder for the whole thing.

I knew something special was arriving when I saw the

Integration isn't enough to make Jazz my music

— BILL HILL reporting



postman staggering up the garden path bearing a massive parcel.

Like many of the programs now appearing for the 512k Mac, it's also a gigantic listing — so large that the program itself takes up one whole disc. In fact it seems to take up 404k on a 400k disc, by some miracle technique of shoehorning.

As a result, two drives are essential — one to hold the program disc, another to hold a startup disc which carries the operating system and a short program, called Convert, for converting Lotus 1-2-3, Symphony and Multiplan spreadsheets into Jazz ones.

That disc leaves you about 200k for data.

You can, of course, eject either disc and save files to a pure data disc — although you're right back there in the disc-swapping days, which you hoped you had left behind when you plugged in Drive 2 for the first time.

Incidentally, I was a bit disappointed to see that the

system on the Jazz disc carried Finder version 1.1.

Since I got the 512k upgrade, I've been using Version 4.1, which appears to have been written specifically for the 512k machine and is certainly faster at opening and closing windows.

It also has additional features like a Minifinder, allowing rapid opening of applications, and a really magic option in the Special Menu called Shut Down, which does just that — resets the machine and spits out both discs with no waiting-time.

So the first thing I did was chuck V.1.1 off the disc and replace it with the new one.

● **Word Processing.** Jazz WP is good, easy to learn, and will work with a 15in Imagewriter if you've got one.

It will allow you to prepare reports into which graphs, spreadsheets and information derived from the database can be pasted with ease — although I hit at least one problem trying to paste in a graph.

Despite my best efforts, and

repeated attempts, the title at the bottom of my graph, "Income 1985-86", always came out with the "e" of income overprinted by the "1" of 1985.

Not a major problem, but annoying when the whole point of such a system is to be able to produce professional-looking documents.

I also found it a bit too easy to scroll the page sideways while trying to scroll vertically.

In WP you have the ability also to use a mailmerge facility to do such things as print mailing labels.

● **Spreadsheet.** Obviously, from the success of Lotus 1-2-3 and subsequent packages from the same source, you'd expect this to be the strongest part of the package, and it is.

With 8192 rows by 256 columns, it gives you a total of 2,097,152 cells. That would have been truly impressive had it not been for Microsoft's recent unveiling of Excel, with double that number and a host of new facilities.

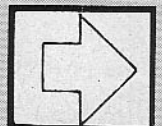
What can you say about a spreadsheet? It appears to work well, and it's fast, although to anyone who has been using Multiplan — or has seen Excel — the need to type in formulae, instead of pasting them in from a Paste Function window, is positively tiresome.

That means that you, not the program, have to get the brackets in the right places. But it doesn't seem to bother the millions already using 1-2-3.

● **Graphics.** Again, these are easy to use, yet powerful, although not in the same league as Microsoft Chart or the integrated graphics in Excel.

I found they printed out without any trouble on their own. It was only when I tried to include them in a WP document that I hit the minor problem I've already outlined.

● **Database.** This can hold several thousand records, and each record can have up to 100



MacReview

fields. It has advanced facilities like field attributes, and all the sorting and searching ability you'd expect.

You can specify up to three sort fields, search for specific records which conform to criteria you enter, print out reports, create forms, and so on.

● **Communications.** You can use this section of Jazz to get the Mac to emulate VT-52 or VT-100 terminals, to allow you to communicate with other micros, mainframes, databases and so on.

But why no TTY – undoubtedly the easiest terminal to use for electronic mail?

Using it, I managed to access Telecom Gold, and send a Jazz WP document which I'd previously saved as Text Only.

Towards the tail end, though, I got an alarming ringing from the machine, which is usually a signal that the disc is running out of memory. That could, however, have been my own fault, and is something I'd like to investigate further.

Summing up, the best thing about Jazz is, as you might expect, not the power of any of the individual packages. We've seen everything before, and more, on programs currently available or soon to be with us.

No, it's the high level of integration which scores here – the ability to cut and paste between applications with ease and speed.

Even if the minor problems I encountered were bugs, and not of my own doing – learning a package so complex obviously takes more than a cursory glance at the manual – Jazz is still a force to be reckoned with.

No doubt, if they are bugs, they will be ironed out fairly sharpish.

But is integration enough? I still don't think so, although I'm sure many will disagree.

Integration on this scale involves a trade-off. None of the individual applications can possibly be as powerful as if they were running singly.

And developments which have taken place recently cast a

The screenshot shows a Macintosh window titled "Worksheet 1" with a menu bar (File, Edit, Window, Type, Plot, Axis, Font, Style) and a toolbar. A "Type" menu is open, showing options: Line, Bar, & Area (checked), Pie, Scatter, Percent, and View Horizontal. The spreadsheet has columns A and B, with rows 1-17. Data in column B: 22 (March), 47 (April), 58 (May), 39 (June). To the right is a bar chart titled "MONTHLY SALES" with a y-axis from 0 to 50. The bars represent the data in the spreadsheet. Below the spreadsheet is a text area containing "DEAR JOHN," and "PLEASE FIND THE ATTACHED FIGURES AND GRAPH. AS YOU CAN SEE WE HAD A VERY SUCCESSFUL HALF YEAR."

	A	B
1	JANUARY	
2	FEBRUARY	
3	MARCH	22
4	APRIL	47
5	MAY	58
6	JUNE	39
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		

whole new light on the matter, anyway.

The Mac has always been a machine which used its own operating system to provide software integration.

You can cut and paste between many applications by using the Clipboard or Scrapbook.

But, even on a two-drive system, it's a bore, because you have to open one package, copy, quit, open the second, paste, quit, and so on.

Until a high-speed Finder appears – and one is on the way – it will always be a slow business.

On a single-drive system it's pure murder – about as quick as watching paint dry and half the fun, with all the disc-swapping that's involved.

But Jazz has to run on the 512k Mac. And a little package that appeared recently – the Switcher – could make it obsolete before it's out of the box.

The Switcher works like this. You load the 33k or so of the program. That gives you a window with the Mac's 512k divisible into sections. So you can load up to four separate

programs into RAM – as many as you have space for, depending on their size.

Say you pick Multiplan and Word. Both are loaded when you select a Switcher file which you have previously configured.

With Word on the screen, you have only to click on an arrow icon in the top right-hand corner, and magically the Word window slides off to one side, to be replaced with Multiplan, almost instantly.

It's the same going the other way. You can cut, copy and paste between applications at tremendous speed. It's integration – without having to compromise on the power of the packages you use.

For most purposes, I can't envisage wanting to run more than three packages simultaneously. And if you have really heavy-duty work to do on any of them, you don't have to make the same compromises...

In a nutshell, that's the philosophy of Microsoft, who are packaging the Switcher with Excel when it appears later this year.

But I've been using Switcher now for a few weeks, to run various packages, and I must

confess that it made me a lot less enthusiastic about Jazz than I otherwise might have been.

If I'm word processing, I'd rather have Word. If I'm spreadsheeting, I'd much rather have Excel. If I'm communicating, I'd sooner use Macterminal. For graphics, I'd rather have Chart. For a database, I'd sooner use File.

I'd be prepared to give up some of Jazz's ease of integration in return for more power.

Where I need to run two or more at once, I'll put Switcher on one of the discs.

That's a very personal view – almost certainly influenced by the fact that I know and love all those packages individually.

Someone coming fresh to the Mac, not influenced by contact with existing applications and not requiring the full power of those individual packages, might well disagree.

I'd certainly advise them to have a good look at Jazz, which could be all the software they'd ever need.

Jazz might strike the right chord with you – but I'll stick to Beethoven.

HAVE you ever added an extra line to an Applesoft program, only to find when you LIST the program that the line isn't there? The explanation is often that you mistyped the line number, say 5010 for 50010 and the line is there, lurking in your program, ready to wreak havoc when you try to run it.

Of course you have your back-up, but you don't really want to throw away all the work done since it was made. If only you could identify the extra lines or the modifications to the lines...

Or did you ever find that revision number 4 of a program didn't work as well as revision 3, and so after revising 3, giving 3b, you revise the revision and then a week later you can't remember which was the up-to-date one and what the difference was anyway?

With long and complex programs, it can be difficult to check that you haven't accidentally garbled some quite unrelated part of your program while at work elsewhere.

The usual ploy, if the programs are fairly long, is to print them both out - tedium, tedium - and then hold up the printouts to the light, one over the other to try and spot the discrepancy. Besides being longwinded, this can often fail, if for example you accidentally introduced a control character into a data listing.

What you need is a utility that

Let Allie list those altered lines

RICHARD SHANN presents a utility to compare different versions of an Applesoft program

compares two programs and lists out any lines that differ. Such is the Applesoft Line by Line Comparator, Allie for short.

To use Allie you don't have to be an initiate of the dark secrets of machine code. While in Basic you just issue the direct command RUN ALLIC.

Program 1 (below) prompts you for the names of the two programs which you wish to compare. Once you have given Allie this information, a series of commands are put up on the screen.

All that remains is to trace over these with the right arrow, pressing Return at the end of each line. The final command LIST will give you a listing of any line present in one program but not in the other, as well as both versions of any line which shares the same line number

but differ in any other respect.

Each line lists exactly as it would in the original program, but with an extra statement, :1 or :2 at the end, which tells you from which of the programs the line has come.

However before you can use Allie you will need to have the machine language program Allie.200 on the disc in drive 1, which unless one of your mates has a copy, means typing the codes in at location \$6200 and issuing the DOS command BSAVE ALLIC.200,A\$6200,L\$149.

Note that the assembly listing shows the code starting at \$200 but it should be entered initially at \$6200 in order to use the BSAVE command as given.

The program works as follows. The commands you issue load the two programs into the

memory side by side, one starting at location \$1001 the other at \$5501. Then the machine code program ALLIC.200 is BRUN from location \$0200, where it is out of harm's way.

ALLIC.200 examines the given programs line by line, building up a third "program" starting at location \$0801. When finished it returns to Applesoft command level, whereupon a LIST command allows all the differing lines to be seen or printed.

The reason for the rigmarole of putting the commands on the screen for you to issue one by one is that by issuing the command LOAD PROG1 an Applesoft program would delete itself. Allie is very fast, and requires no extra disc space to work, unlike programs which

```

10 TEXT : HOME : CLEAR          TURN WITH THE CURSOR,":
20 PRINT : PRINT : PRINT        PRINT "PRESSING RETURN AT
   SPC( 4): INVERSE : PRINT     THE END OF EACH LINE":
   "APPLESOFT LINE BY LINE     PRINT "(TYPE NEW IF A LOAD
   COMPARATOR"                  FAILS)": PRINT
25 PRINT : NORMAL : PRINT       40 PRINT
   SPC( 14): INVERSE : PRINT    "JPOKE104,85:POKE21760,0":
   "ALLIC IV"                   PRINT : PRINT "JLOAD ";A$ +
30 NORMAL : PRINT : PRINT       S$ + D$;
   SPC( 3)"BY RICHARD SHANN":   50 PRINT : PRINT
   PRINT                         "JPOKE104,16:POKE4096,0":
33 POKE 34,6                    PRINT : PRINT "JLOAD ";B$ +
34 GOSUB 1000                   T$ + E$;
35 PRINT "COPY EACH LINE IN     80 PRINT : PRINT "JBRUN
                                ALLIC.200,A$200,D1";
90 PRINT : PRINT "JLIST";       100 VTAB (10)
100 VTAB (10)                   120 END
120 END                          1000 REM
1010 INPUT "GIVE NAME OF
   FIRST PROGRAM ";A$          1010 INPUT "GIVE NAME OF
1012 PRINT "SLOT NO (DEFAULT   6) ";: GET S$: PRINT S$: IF
   S$ = CHR$ (13) THEN S$ =    "6"
1013 S$ = ", S" + S$           1014 PRINT "GIVE DRIVE NO
   (DEFAULT 1) ";: GET D$:    PRINT D$: IF D$ = CHR$
   (13) THEN D$ = "1"         1015 D$ = ", D" + D$
1020 INPUT "GIVE NAME OF
   SECOND PROGRAM ";B$        1022 PRINT "SLOT NO (DEFAULT
   6) ";: GET T$: PRINT T$: IF
   T$ = CHR$ (13) THEN T$ =
   "6"
1023 T$ = ", S" + T$          1024 PRINT "GIVE DRIVE NO
   (DEFAULT 1) ";: GET E$:
   PRINT E$: IF E$ = CHR$
   (13) THEN E$ = "1"        1025 E$ = ", D" + E$
1026 HOME                       1030 RETURN

```

Program 1

first turn the Applesoft program listings into text files and then compare the listings.

The assembly listing for ALLIC.200 is shown in Program II overleaf.

To understand ALLIC.200 in detail requires a little knowledge of how Applesoft stores program listings. Each program line is assigned a chunk of memory, ending with a code £\$00, corresponding to Return.

The first two codes of this chunk are the address of the next line's chunk, and the next two codes are the line's line number. If the address of the next lines chunk reads \$0000, this means by convention that there are no more chunks – end of program.

The address of the first chunk of a program is stored in locations \$67,68 on the zero page, and the first chunk is preceded by a £\$00 code.

Thus ALLIC.200 examines the two programs, which we

can call Prog1 and Prog2, starting with their first chunks, and reads their line numbers. If one is lower than the other it loads that chunk into \$0801.

There are now three addresses to keep track of, \$75,\$76 points to the location of the next chunk in the LISTing at \$0801, \$77,\$78 points to the next chunk requiring consideration in Prog1, and \$7D, \$7E does the same for Prog2.

In addition \$83,\$84 is used to hold either the address in \$77,\$78 or in \$7D,\$7E when a chunk is being read from either Prog1 or Prog2 into the LISTing.

These two cases are distinguished by the sign of the flag \$FA, which is also used to generate the tag - :1 or :2 – at the end of each line.

Thus the program works its way through Prog1 and Prog2 together, always taking lines from whichever offers the lowest line number. When it encounters lines with the same

line number, it examines the lines themselves and if any discrepancy is found it adds both to the listing.

To load ALLIC.200 into the memory you should *not* type the codes in at address \$0200, even though this is where it is BRUN from. This is because page two is used for input, so your BSAVE command would be written on top of the program you were trying to save.

Type the codes into \$6200 instead, then BSAVE ALLIC.200,A\$6200,L\$149. The BRUN command includes the address parameter, so ALLIC.200 always goes to the right place.

The Applesoft program Allic is quite simple. The only trick is to put the lines on the screen so that after each command is performed the cursor ends up nicely positioned for the next command. The POKES tell the DOS where to load the programs, and the BRUN com-

mand does the comparison.

There are limitations to Allic, but these are not serious. It doesn't like programs with no lines at all. Moreover, at the other end of the scale, as both programs are to be loaded at once, neither can take up more than half the memory.

In practice most programs leave considerable space for data which is not needed here. To maximise the space, the LISTing created by ALLIC.200 is designed to overrun the start of Prog1 if necessary.

If it does this before the lowest numbered lines in Prog1 have been dealt with then the program halts, with a bell, and you only get a partial listing. This can generally be avoided by loading the programs in the reverse order.

But, of course, if there are enormous differences between the programs then you don't need Allic to tell you.

appletips

i While glancing through the April 1985 issue of Apple User I noticed that my three sentence Appletip of May 1984 had been incorporated into a half page demonstration of the idea using ONERR techniques.

This reminded me of a more elegant routine which I wrote to do exactly the same thing.

The routine is short enough to be reasonably self explanatory. The main principle is manipulation of DOS location 43624 which holds the current drive number, relying on a RESUME statement returning to a conditional (IF) BLOAD statement which tests the number of times the error routine has been invoked.

After two attempts therefore, flow is diverted around the BLOAD leaving C% as an indicator of success.

Program flow is continued

in the main segment regardless of what has caused the error routine to be used.

This allows the programmer to continue to take corrective action by means of further tests on PEEK (222) in the event of other types of error, say a full disc, and by prompting the user for action, for example, an open drive door.

The default drive is reinstated simply by POKING it back regardless, rather than testing for a change and doing a toggle.

On a point of good programming practice, line 130 is doing a BLOAD to make the demonstration simple.

In general the routine is useful when reading text files where a VERIFY before OPEN would prevent the creation of empty files, and where there are more files than would fit on one disc.

There is less justification

for using this technique for running or loading programs.

Derek Turner

```
100 REM AUTO DRIVE SWAPPING
    BY DEREK TURNER
110 CX=0:DV=43624:DR=PEEK(DV)
120 ONERR GOTO 1000
130 IF CX<2 THEN PRINT
    CHR$(4)"BLOAD FILE":CX=0
140 POKE 216,0: REM CLEAR
    ONERR
150 IF CX THEN PRINT "FILE NOT
    FOUND ON EITHER DRIVE": REM
    CX NOW NO-FILE FLAG
160 POKE DV,DR: REM REGARDLESS
170 STOP: REM REST OF PROGRAM
```

```
1000 REM ERROR TRAP
1010 IF PEEK(222)<5 OR
    PEEK(222)>6 THEN PRINT
    "DISK ERROR DRIVE "
    PEEK(DV)
1020 CX=CX+1: POKE
    DV,1+(PEEK(DV)=1): RESUME
```

i For some time I had been irritated by the confusing layout of Applesoft listings via an Epson printer and the waste of paper.

Accidentally I discovered that if POKE33,33 is typed before issuing the LIST command full carriage width listings are obtained.

Steve Morris

i Under DOS 3.3 the routine at address \$A56E will CATALOG the currently logged drive.

By pointing the & vector at this routine a single key CATALOG can be obtained under Applesoft.

To initialise the & vector do the following pokes:

POKE 1013,76: POKE 1014,110:
POKE 1015,165.

D.B. Peterson

SOURCE FILE: ALLIC

----- NEXT OBJECT FILE NAME IS ALLIC.OBJ0

```

0200:      1      ORG    $200
0200:A9 01      2      LDA    $901      ;Set $70,$7E to point to
0202:85 75      3      STA    $75      ;first chunk ($1001) for
0204:85 7D      4      STA    $7D      ;prog 1 and likewise $77,$7B
0206:85 77      5      STA    $77      ;for prog 2 and locations
0208:85 67      6      STA    $67      ;$67,$68 for the LISTING.
020A:A9 10      7      LDA    $910
020C:85 7E      8      STA    $7E
020E:A9 55      9      LDA    $955
0210:85 78      10     STA    $78
0212:A9 08      11     LDA    $98
0214:85 68      12     STA    $68
0216:85 74      13     STA    $74
0218:20 1E 02 14     JSR    RUNNE      ;Runs Program
021B:4C 00 03 15     JMP    $0300      ;Back to Applesoft
021E:20 49 02 16     JSR    RCLN1      ;Find current line
0221:20 55 02 17     JSR    RCLN2      ;numbers
0224:20 5E 02 18     JSR    CMPR      ;do comparison
0227:A0 01      19     PREND  LDY    $801      ;End of one or other of
0229:B1 70      20     LDA    ($7D),Y      ;the BASIC programs?
022B:F0 04      21     BEQ    BRAF9      ;($7D)=$7E?
022D:B1 77      22     LDA    ($77),Y      ;Signalled by hi-byte of address
022F:F0 0C      23     BEQ    BRAFA      ;of next chunk equal to $90
0231:00 EB      24     BNE    RUNNE
0233:B1 77      25     BRAF9  LDA    ($77),Y
0235:F0 0C      26     BEQ    END
0237:20 81 02 27     JSR    MAKE2
023A:4C 27 02 28     JMP    PREND
023D:20 6E 02 29     BRAFA  JSR    MAKE1
0240:4C 27 02 30     JMP    PREND
0243:91 75      31     END    STA    ($75),Y
0245:C8      32     INY
0246:91 75      33     STA    ($75),Y
0248:60      34     RTS
0249:A0 02      35     RCLN1 LDY    $82      ;Read current line number
024B:B1 7D      36     LDA    ($7D),Y      ;of prog 1 into $7B,$7C
024D:85 7B      37     STA    $7B
024F:C8      38     INY
0250:B1 7D      39     LDA    ($7D),Y
0252:85 7C      40     STA    $7C
0254:60      41     RTS

```

```

0255:A0 02      42     RCLN2 LDY    $82      ;Read current line number
0257:B1 77      43     LDA    ($77),Y      ;of prog 2 into A,X
0259:AA      44     TAX
025A:C8      45     INY
025B:B1 77      46     LDA    ($77),Y
025D:60      47     RTS
025E:C5 7C      48     CMPR   CMP    $7C      ;Look at current line numbers
0260:90 1F      49     BCC    MAKE2      ;in 7B,C and A,X and make list
0262:F0 02      50     BEQ    CMLPB      ;of any incomparable, update 7D,E and 77,B
0264:80 08      51     BCS    MAKE1
0266:E4 78      52     CMLPB CPX    $78      ;Compare the low byte
0268:90 17      53     BCC    MAKE2
026A:F0 28      54     BEQ    CMLP1
026C:80 00      55     BCS    MAKE1
026E:20 74 02 56     MAKE1 JSR    ISLESS      ;Prog 2 has higher line number
0271:4C 02 02 57     JMP    MAKEL      ;so add line from prog 1
0274:A5 7D      58     ISLESS LDA    $7D
0276:85 83      59     STA    $83
0278:A5 7E      60     LDA    $7E
027A:85 84      61     STA    $84
027C:A9 32      62     LDA    $82      ;Flag to say whether $83,4
027E:85 FA      63     STA    $FA      ;refers to prog 1 or 2
0280:60      64     RTS
0281:20 87 02 65     MAKE2 JSR    ISMORE
0284:4C 02 02 66     JMP    MAKEL
0287:A5 77      67     ISMORE LDA    $77
0289:85 83      68     STA    $83
028B:A5 78      69     LDA    $78
028D:85 84      70     STA    $84
028F:A9 B1      71     LDA    $81
0291:85 FA      72     STA    $FA
0293:60      73     RTS
0294:A0 04      74     CMLP1 LDY    $84      ;Line numbers are equal so
0296:B1 7D      75     CONTC  LDA    ($7D),Y      ;compare lines themselves
0298:D1 77      76     CMP    ($77),Y
029A:D0 07      77     BNE    MAKELS
029C:AA      78     TAX
029D:F0 0A      79     BEQ    NEXTLS
029F:C8      80     INY
02A0:4C 96 02 81     JMP    CONTC
02A3:20 B1 02 82     MAKELS JSR    MAKE2      ;Lines differ so add to list
02A6:4C 6E 02 83     JMP    MAKE1
02A9:20 74 02 84     NEXTLS JSR    ISLESS      ;Move to next line

```

Program 11

BASIC

PASCAL

P-tral: BASIC to Pascal Translation Software

P-tral is the user-friendly translation software that converts Applesoft BASIC programs to Apple Pascal.

- Translates any business, scientific, graphics and game software from source.
- Translation results approach 100%.
- Translates any sized program—Large or Small.
- Essential compile directives generated.
- PIONEER version only £125 + VAT.

REQUIREMENTS: 64/128K RAM, Dos 3.3, 80-column card, Apple II, Ile, Iic, Apple Pascal 1.1 or 1.2

To order or request more information contact:

DMS ELECTRONICS LTD

Park House, Manor Road, Kiveton Park Station, Sheffield S31 8PB, England
Tel: Worktop (0909) 773399

P-tral is published by: **WOODCHUCK INDUSTRIES®**
340 West 17th Street, New York, NY USA 10011
(212) 924-0576 / (212) 206-6490

Software

We've turned business/accounting software on its head!

Frustrated by inflexible computer packages and poor support?

We'll give you a powerful Open Item Purchase/Sales/Stock System* and in 2 days, train you to customise it to suit your own specific business requirements (Order Processing, Costing, etc)

Suitable for most popular micros—Apple, Macintosh, IBM, Olivetti, Pinnacle, and others

Single or multi-user versions available

EASY to learn, EASY to change, EASY to use

*Requires the Omnis 3 database

TO DO IT PROPERLY — DO IT YOURSELF

BUSINESS CONTROL SYSTEMS

FREEPOST, Caterham, Surrey CR3 9UH Telephone: (0883) 40861/43553

```

02AC:A0 00 85 LDY #0
02AE:B1 83 86 LDA (#83),Y
02B0:B5 85 87 STA #85
02B2:20 28 03 88 JSR NEXLN
02B5:20 87 02 89 JSR ISMDRE
02B8:A0 00 90 LDY #0
02BA:B1 83 91 LDA (#83),Y
02BC:B5 85 92 STA #85
02BE:20 28 03 93 JSR NEXLN
02C1:60 94 RTS
02C2:A0 00 95 MAKEL LDY #0 ; Add line to list
02C4:B1 83 96 LDA (#83),Y
02C6:B5 85 97 STA #85
02C8:38 98 SEC ;Needed later
02C9:E5 83 99 SBC #83
02CB:18 100 CLC ;Calculate offset
02CC:45 75 101 ADC #75
02CE:91 75 102 STA (#75),Y
02D0:A5 74 103 LDA #74
02D2:A9 00 104 ADC #0
02D4:C8 105 JNY
02D5:91 75 106 STA (#75),Y
02D7:B8 107 DEY ;Add a further #2 for appending
02D8:A9 02 108 LDA #2 ;two bytes (:1 or :2) for
02DA:71 75 109 ADC (#75),Y ;the program number
02DC:91 75 110 STA (#75),Y
02DE:C8 111 JNY
02DF:B1 75 112 LDA (#75),Y
02E1:A9 00 113 ADC #0
02E3:91 75 114 STA (#75),Y
02E5:C8 115 CONTH JNY
02E6:B1 83 116 LDA (#83),Y
02E8:91 75 117 STA (#75),Y
02EA:C0 03 118 CPY #3
02EC:F0 F7 119 BEQ CONTH
02EE:AA 120 TAX
02EF:F0 02 121 BEQ UPDAT ;Look for #00 at end of line
02F1:D0 F2 122 BNE CONTH
02F3:A9 3A 123 UPDAT LDA #3A ;Add in :1 or :2 at end
02F5:91 75 124 STA (#75),Y
02F7:C8 125 JNY
02F8:A5 FA 126 LDA #FA
02FA:29 7F 127 AND #F7

02FC:91 75 128 STA (#75),Y
02FE:A9 00 129 LDA #0
0300:C8 130 JNY
0301:91 75 131 STA (#75),Y
0303:A0 00 132 LDY #0 ;Update #75,4 to end of list
0305:B1 75 133 LDA (#75),Y
0307:AA 134 TAX
0308:C8 135 JNY
0309:B1 75 136 LDA (#75),Y
030B:85 74 137 STA #74
030D:8A 138 TXA
030E:85 75 139 STA #75
0310:A5 7E 140 LDA #7E ;Check to see that listing is not
0312:C5 74 141 CMP #74 ;overtaking program 1 at #7D,E
0314:D0 12 142 BNE NEXLN
0316:A5 7D 143 LDA #7D
0318:38 144 SEC
0319:C5 75 145 CMP #75
031B:90 0B 146 BCC NEXLN
031D:20 3A FF 147 JSR $FF3A ;Overtaken so ring bell
0320:A9 00 148 LDA #0 ;and end
0322:20 43 02 149 JSR END
0325:4C 0B 03 150 JMP #03D0
0328:A0 01 151 NEXLN LDY #1 ;Move to next line
032A:B1 83 152 LDA (#83),Y
032C:85 84 153 STA #84
032E:A5 85 154 LDA #85
0330:85 83 155 STA #83
0332:24 FA 156 SKIP2 BIT #FA ;Test for prog 1 or prog 2 chunk
0334:10 09 157 BPL BRAE9
0336:A5 83 158 LDA #83
0338:85 77 159 STA #77
033A:A5 84 160 LDA #84
033C:85 78 161 STA #78
033E:60 162 RTS
033F:A5 83 163 BRAE9 LDA #83
0341:85 7D 164 STA #7D
0343:A5 84 165 LDA #84
0345:85 7E 166 STA #7E
0347:60 167 RTS

```

*** SUCCESSFUL ASSEMBLY: NO ERRORS

OMNIS® DATABASES

SUPPLIERS
CONSULTANTS
TRAINING

Basic and advanced courses on OMNIS 2 and OMNIS 3 using Ile's or Mac.

One-day training run continuously -
Phone for dates and details.

MicroFinance Systems Ltd.

Chartwell House, 38 Church Street, Malvern, Worcestershire WR14 2AZ.
Tel: Malvern (06845) 60141

C.I. CAYMAN LTD.

ACCELERATOR BOARD II+ Ile £199

Processing is three and a half times faster, 3.6MHz 6502 and fast memory on board. FULLY documented.

BACKUP CARD II+ Ile £49

Copy protected software, FULLY documented.

16K RAM card £35

HEAVY DUTY POWER SUPPLY II+ Ile £49

CUMANA DISC DRIVES II+ Ile £115

Phone: 021-705 7097

C.I. Cayman Ltd. P.O. Box 77, Solihull, West Midlands B91 3LX

TURN YOUR APPLE INTO A REAL TIME IMAGE PROCESSOR WITH THE IMAGE III FRAME STORE

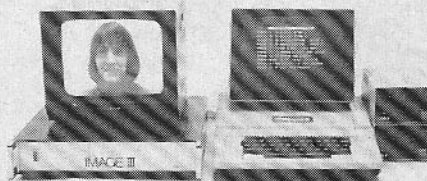


IMAGE III is a high resolution Frame Store which can capture and display pictures in real time from any 625/525 line video source. Once captured in the 512 x 512

frame memory, the computer can access the stored image for processing or manipulation. The store utilizes 6 bit A/D and D/A converters to give up to 64 grey levels per pixel. A major feature of this store is that if a lower resolution picture is selected then the store can be partitioned to store multiple pictures, e.g for 256 x 256 resolution, four pictures can be stored. This allows the computer to compare two or more pictures captured from the same or different video sources.

The IMAGE III Frame Store turns your computer into a low cost image processing system and opens up a range of possibilities such as Robotic Vision, Medical Imaging, Factory Inspection etc. Alternatively the store can be used in applications where picture data is arriving slowly, e.g. weather satellite transmissions, ultrasonic imaging, enabling the user to have a steady display without the need for long persistence display devices.

IMAGE III is available for the IBM PC, Apple and BBC computers. The interface card connects directly to the expansion ports of the computer and software is supplied which demonstrates the features of the store.

Price: £1,990.00

Eltime Ltd.

Unit D29, Maldon Industrial Estate, Fullbridge, Maldon, Essex, CM9 7LP
Tel: 0621 59500

The TV Picture Store Boards used in IMAGE III was developed by British Telecom Research Laboratories and is manufactured under licence by Eltime Ltd. This board can be purchased separately for OEM applications.



LAST month I outlined developments already under way in the USA, aimed at using the Macintosh as the centre of a page-layout and typesetting system which would take the printing world by storm.

Perhaps the most crucial part of the whole system is the third-party page layout and typesetting software which would be needed to drive it, and happily at least one company has lived up to the highest expectations of those who've been watching developments closely.

So far, three packages have emerged from the USA — MacPublisher, from Boston Software Publishers distributed by P & P Micro at a price of £109, Ready Set Go, from Manhattan Graphics Corporation distributed by Heyden Datasystems at a price of £129, and PageMaker, from the Aldus Corporation.

Aldus was kind enough to send me a Beta Test version of PageMaker, which should be available in the USA by the time you read this, and in Europe by the beginning of next year.

It's just as well they did, for my own experiences of testing the three packages, and driving them hard, showed conclusively that PageMaker — as you might expect, a good bit more expensive than the others at a US price of \$495 — is so much better that it makes them look pale in comparison.

That's not the case at first sight, though. In the early stages of testing, I had only a demonstration copy of PageMaker, with many of its facilities not yet implemented, and it seemed as if MacPublisher was the clear leader.

But my experiences of working with it threw up some annoying quirks, and one or two features which in my opinion make it entirely unsuitable for professional, or even serious amateur, use.

At first sight all three packages look similar. The starting point for all is the blank page, into which you place text and graphics. But they all go about it slightly differently.

Both MacPublisher and

This is the future — and it works

Following his last article on typesetting with Mac, **BILL HILL** reviews three software packages designed to create computerised page layouts

PageMaker work to a column-type format. Ready Set Go works with boxes of text. You create and change the size of the boxes, and the text fills them accordingly.

In all three packages you can paste in text from either MacWrite or Microsoft Word and graphics from MacPaint or MacDraw.

MacPublisher is the only package which will run on a 128k Macintosh — although when I was using it before my machine was converted it would frequently run out of memory — and all of them really need the external disc drive.

For any sort of serious use a 512k Mac is essential, and, since you are really working with large and complex graphics when dealing with page layouts, they are hungry for disc space.

If you were working seriously with any of these systems, I'd say there's little alternative but a hard disc.

To digress for a moment. There's another piece of Mac software which is important to make working with these systems easier.

If you think about it, you will want to create text in a word processor like Microsoft Word. You'll want graphics software on hand, too.

What you really need is the ability to integrate either of those with the page layout package, and switch back and forth instantly between them.

Well, the software to let you do this exists — The Switcher.

I set up each of the page

layout packages in RAM with Word and/or MacPaint — depending on how much memory they took up. That proved to be the answer to creating copy and graphics and instantly transferring them to the page.

Then I got down to business...

MacPublisher has some nice features. You start off with a blank page, and you can create a magazine of as many pages as you like — provided you have the disc space — turning over from one page to the next with the mouse.

You input text into galleys — long single-column runs. You select one-third page width, half-page width, or two-thirds page width. The text is automatically justified when you change.

Getting it in the page is really quite clever. From the Layout Menu, you pick up a pair of scissors with the mouse, and click in the text below the last line you want to cut.

You can then lift the first section up, drop it in the page, and MacPublisher automatically creates a "carry-over" with the rest — and remembers where the first section has been placed.

You can just carry on pasting in section after section, altering typefaces or formatting on each section if you like.

Runs of text or graphics appear on the page as boxes, although there is a Show Minipage option in the Layout menu which lets you see what they really look like as you are

building up the page.

To place graphics, you take a MacPaint graphic, and copy it to the clipboard. You can then, from the Apple menu, select a Camera.

This pops up on screen, allows you to crop your graphic — in depth only — and you then click the mouse.

The graphic is saved as a MacPublisher picture, and can be dropped on the screen in the same way as text.

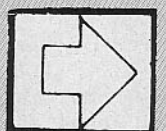
Anything you place on screen is automatically constrained to sit correctly in the column format.

Say you have chosen a three-column — one third page — format for your text. If you then place a two-thirds page wide piece of text, it will sit on the page so as to leave you exactly one column at either the left or right side for more text, or a graphic.

However, there is a Free Form format, which allows you to place items anywhere on the page — even on top of each other or overlapping, if you want.

When I first tried the package, I was entranced. But the more I went on, the less happy I became about detail — and the final results.

For example, it's almost impossible, when pasting in columns of text alongside each other to ensure that the text in



MacPublishing

adjacent columns is exactly level.

Sounds nitpicking, perhaps, but when you see the results printed out, it looks really amateurish.

There are problems with graphics, too. Because you have to copy a MacPaint document into the clipboard first, you can only carry across what appears on one screen, not the full MacPaint page. So you can't get a graphic the full width of your MacPublisher page.

My attempts to overcome this when using MacPaint to create a masthead for a hypothetical magazine were frustrating in the extreme, and disappointing at the end of the day.

I tried the package for about three weeks. By then I was so turned off that I turned to the only other one I had at that stage, Ready Set Go. I'd already dismissed the demo version of PageMaker as being worse than useless for review purposes — you couldn't even Save a publication.

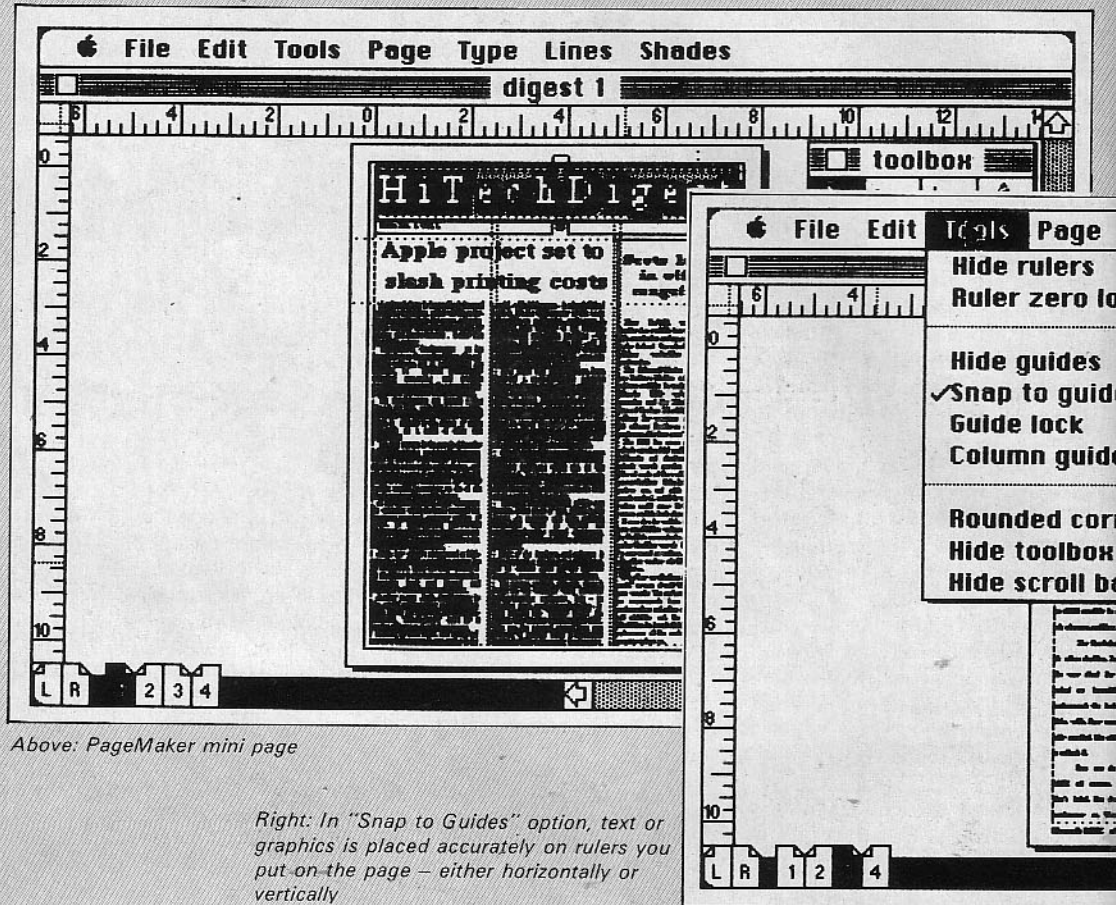
RSG has a lot fewer gimmicks than MacPublisher. It doesn't look as good on screen, and the way you have to create box sizes for text and graphics is a bit finicky. But for all that, if I wanted a package just to allow me to turn out a few pages a month, I'd pick this over MacPublisher.

You create a box of whatever size you wish, and then place the text or graphics in it. Once it's there, altering box size automatically alters what's inside it — so a piece of text, fully justified, in a two-inch box, will also be fully justified in a four-inch box — although half the depth.

The final results with Ready Set Go looked much cleaner, and it was much more precise than MacPublisher, with an option to allow you to see the full-sized page in sections, so you could line up everything correctly.

It's simple to learn, as well. But it wasn't up to the mark for what I wanted.

Luckily, before terminal despair set in, a Transatlantic telephone conversation which



Above: PageMaker mini page

Right: In "Snap to Guides" option, text or graphics is placed accurately on rulers you put on the page — either horizontally or vertically

I'd had with Aldus in Seattle a few days earlier — you ought to see my telephone bills — produced a Beta Test version of PageMaker.

It came with two pages of dire warnings about not depending on it to meet deadlines because it was still a development version, and a list of known bugs which were being fixed before final release.

There were a few others which weren't listed, but which subsequent talks on the phone revealed had already been turned up by other Beta Testers. Some of them caused the system to crash.

But for all that, a night's work convinced me that this, indeed, was the one — once all bugs were out of it — and what has happened since has reinforced that in the most practical manner possible.

Even the Beta version is very

workable, if you know where to make allowances.

PageMaker is spectacular — a fully professional page layout package, which will let you create pages in multiples of 16.

It has all the tools the designer could wish for to place text accurately on the dummy pages, to place, crop and size graphics, to create mastheads, headings — the lot.

The final release price may sound steep by comparison with the others, at \$495. But make no mistake about it, a year ago a professional typesetting company would happily have handed over ten times that much for this kind of power — and felt it was getting a bargain.

I drove it about as hard as you possibly could. That's a story in itself.

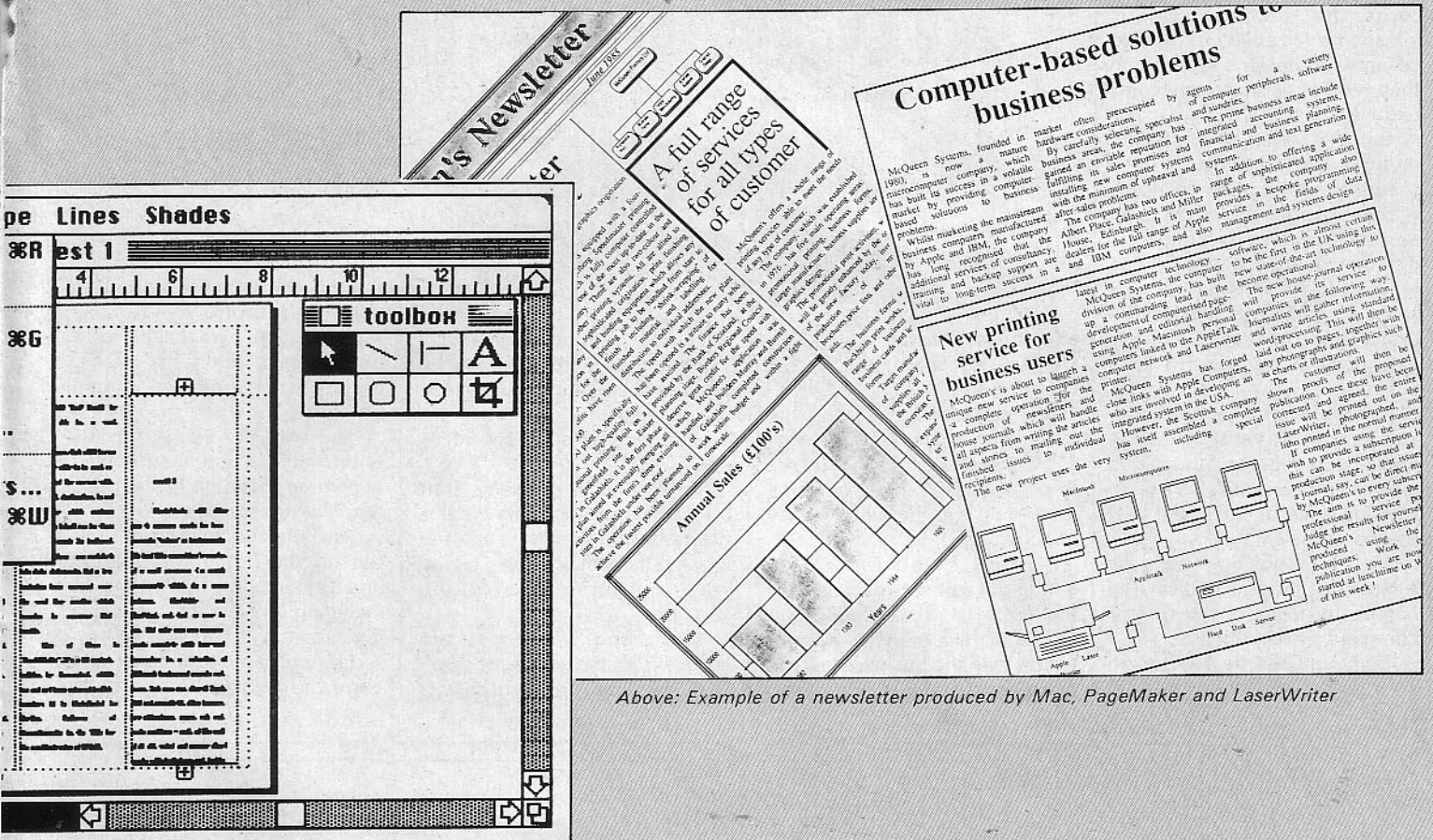
One day, I was taking part in an Apple seminar with a friend of mine. He mentioned that his

company was opening a new printing factory on the Friday, and it would have been nice to give all the invited guests a two-page newsletter produced especially for the occasion.

Now, it so happened that at the seminar we had three Macintoshes, networked to a LaserWriter. So we thought "Let's try it". We had nothing but a Friday-morning deadline. No text, no graphics, no masthead — and it was lunchtime on Wednesday...

Within five hours of starting — and we'd had a few crashes, caused by ignoring the bug warnings in our haste — we were ready to output to the Laser Writer.

The final results were great — more than good enough for camera-ready artwork. Next day, they were at the new print factory, where they were photographed and 500 copies were



Above: Example of a newsletter produced by Mac, PageMaker and LaserWriter

litho printed.

It looked professional – as good as you'd get by sending out to a typesetting house.

The job we had tackled was asking far more than we had any right to expect from a Beta Test package. But PageMaker passed with flying colours.

This is NOT a piece of gimmicky software to let amateurs turn out poor imitations of professional work. PageMaker is a fully functional, page design software tool for the pro, which is so easy to use that an amateur can turn out the same quality work.

As such, what the Macintosh, with the FileServer, network, LaserWriter and PageMaker will do to the typesetting trade is revolutionise the whole business overnight.

This is perhaps the most significant development since the Mac itself was launched.

Already, in the States, one company has started to set up a chain of print shops where Joe Soap can just walk in off the street and pay by the hour to use the system to turn out magazines, newsletters, company reports, and so on.

There isn't, to my knowledge, another system in the world that's easy enough to use to allow that kind of operation.

And the LaserWriter results are just amazing.

Remember, we are talking here about a hardware/software system to produce full camera-ready copy for less than £15,000, even with the 512k Mac, a LaserWriter and a hard disc – which you'd really want for full functionality.

Professional print firms have been paying £150,000 and upwards for that sort of capability, and it's nowhere near as easy to use.

By itself, the capability it brings, at a fraction of the cost of existing professional typesetting systems, is certain to spawn hundreds of new, small-circulation publications, completely alter the way company documentation is produced, and lots more.

The fact that PageMaker, the laser printer, and all the other parts of the system use the Adobe PostScript page description language is also highly significant. It means that anyone can save his or her publication as a PostScript file – and transmit it by modem to a remote printing location.

All the possibilities offered by the whole shooting-match are going to come together to give Apple something to offer which is unequalled by any other company.

There are lots of other developments on the way, some of which I can't talk about yet.

All I can say is, if you have ever had a hankering to produce a magazine, for your company, or for a specialist profession like

speech therapy to which you belong which couldn't support its own publication at current production costs, the answer is here.

On glossy paper, the quality could be as good as the excellent magazine you're reading at this moment.

If you buy MacPublisher – because it's available now – you are likely to end up regretting it, and eventually shelling out for PageMaker anyway.

If you have to settle for a low-cost option which you could operate without a hard disc, buy Ready Set Go instead.

But, if you really mean business, there's no other answer but PageMaker.

You might have got the impression by now that I've been slightly inspired. You'd be dead right.

It's not often, reviewing software, that you see a revolution in the making. But I have seen the future, and it works. Its names are three – Macintosh, LaserWriter and PageMaker.

RONYAR Ltd is a small but thriving kitchen and bathroom design company in Denham, Bucks. Graham Raynor, who owns the business, prides himself on his ability to provide clients with designs that match their requirements at very competitive prices.

But the task of producing layouts and drawings to a customer's brief, then costing a whole range of units and accessories from a number of different suppliers was at one time a real headache.

As Graham explained: "The first drawing or costing is hardly ever the final one in this business.

"Doing everything manually was making it very difficult to make changes for our clients quickly and accurately. We were wasting a lot of time, and that means money".

He started looking round for a microcomputer specifically to handle the design and costing of complete fitted kitchens.

Early in 1984 he decided on an Apple IIe with a memory

Designing new kitchens - in Apple pie order

By **BRYAN WILLIAMS**

expansion card to 128k, Duodisk, dot matrix printer and colour plotter.

His reason for choosing the IIe?

"Quite simply", he said, "the Apple was the only machine that could run the software we needed.

"There is an incredible amount of software available for the Apple II and that's important for people like me who want to get on with the business instead of spending hours struggling

with programming techniques".

Ronyar now has the Apple IIe system installed in Graham's office working full-time on designs and costings for kitchens.

When one of the sales staff brings in a new client's details, the name and address is entered into the system, followed by the wall measurements of the kitchen.

Floor and wall units are keyed in as blocks of a certain size, then the program selects

all possible stock items of all sizes required.

Selection is made by number, then the same system is followed for worktops and accessories such as ovens, sinks, electrical appliances and so on.

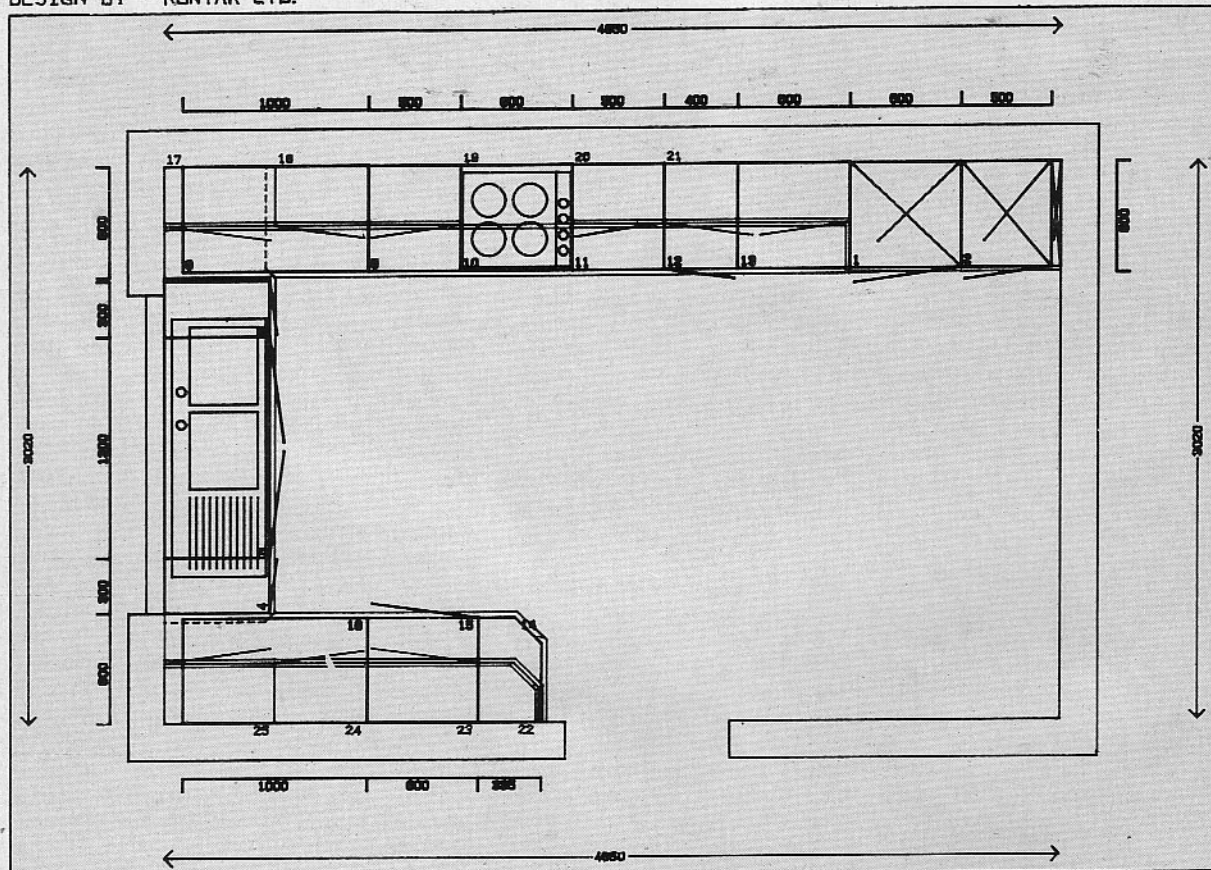
The selected items for the entire kitchen are then costed according to the particular price range of the client's choice.

An itemised costing is printed on the dot matrix printer, and Ronyar has the option of including its own costs, such as consultancy, fitting and so on.

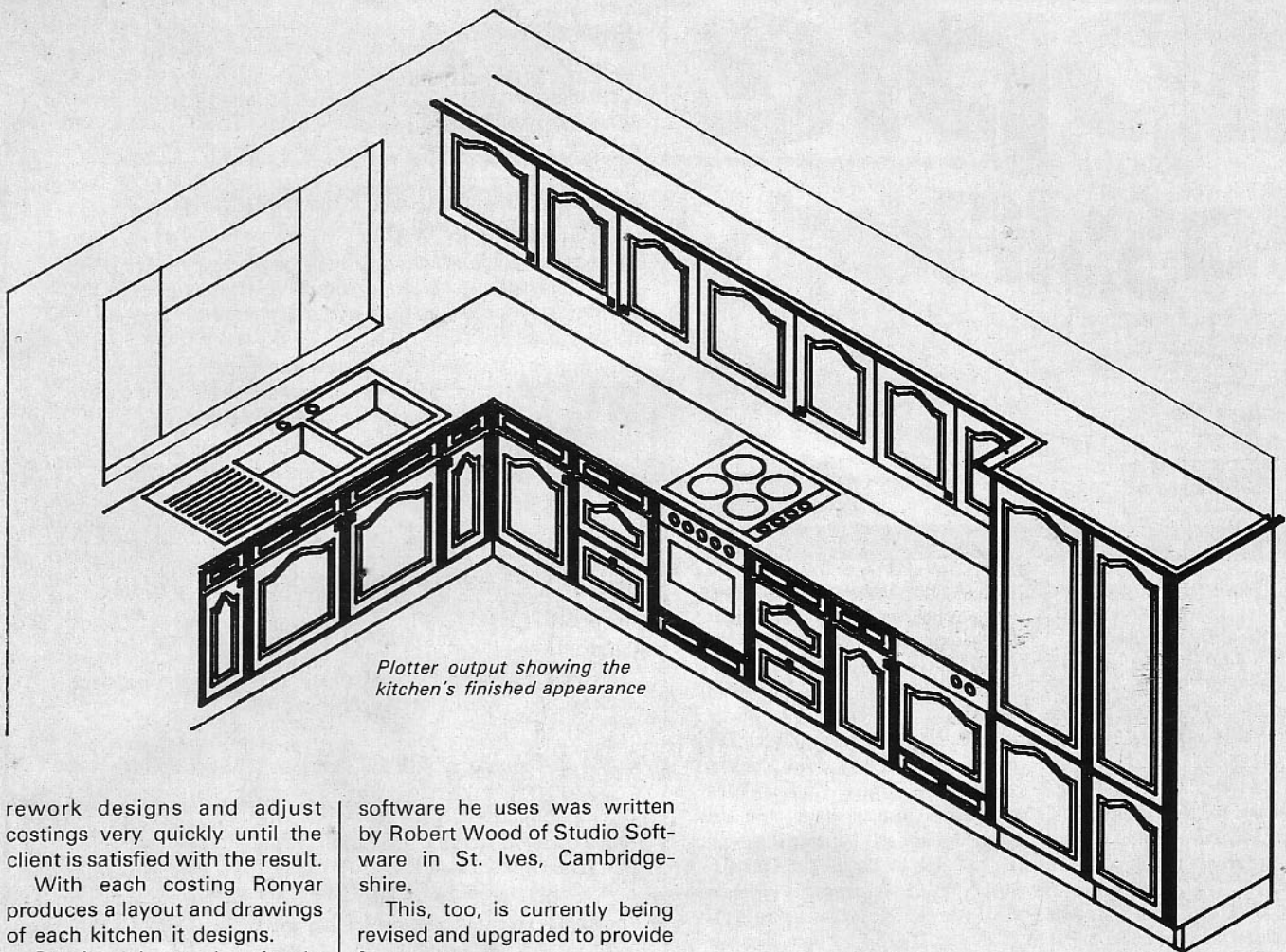
Editing can be done at any time, allowing Graham to

DESIGN BY - RONYAR LTD.

SCALE 1:20



Layout of a kitchen. The output also includes a colour-coded list of all units.



Plotter output showing the kitchen's finished appearance

rework designs and adjust costings very quickly until the client is satisfied with the result.

With each costing Ronyar produces a layout and drawings of each kitchen it designs.

Produced on the Apple colour plotter, these consist of a layout of the entire kitchen with a colour-coded list of all units, and three-dimensional drawings of each wall.

The next step here will be for perspective drawings, which Graham hopes to be able to produce in the future.

The system greatly simplifies Ronyar's ordering process, since items are ordered from several suppliers for one kitchen.

With all the relevant order codes and suppliers already stored on the Apple 11e, the error margin has been reduced quite considerably in this area and a great deal of time has been saved.

Ronyar uses its Apple 11e to capacity. Only one of its expansion slots is idle, and it can process all but the most complex briefs.

As Graham puts it: "It's amazing how much the machine can cope with".

His only requirement at the moment is more speed, and he intends to buy a new card for the Apple 11e to provide this. The

software he uses was written by Robert Wood of Studio Software in St. Ives, Cambridge-shire.

This, too, is currently being revised and upgraded to provide faster response.

At present it works only on the Apple II, but is now being adapted for the Macintosh.

Graham Raynor has had very few maintenance problems with his Apple 11e, but he has taken out a service contract with his Apple dealer, Action Data of High Wycombe, which undertakes to be on site within 24 hours and to replace any of the Apple equipment if it cannot be repaired within 1-2 hours.

The continued success of Ronyar Ltd in an industry that has suffered during the recession owes a great deal to the service Graham Raynor is now able to provide on the Apple 11e.

"Quite a few similar companies offer computer costings, but our system, offering designs and handling our administration, is fully comprehensive. Our response times have improved at every stage of the work".

Ronyar is, in fact, now considering buying another Apple 11e to handle its accounts - the other system is far too busy getting on with the job in hand.

COMPUTER REF. RG/1/25
FILE REF. A4524
DATE 20 FEBRUARY 1985
RANGE KOMET
FINISH RUSTIC OAK

REFERENCE	DESCRIPTION	COST
TALL UNITS		
6D-60.24-R	APPLIANCE HOUSING 6D-60.24 (FRIDGE)	369.20
D1 90 33	CARVED DECOR PANEL	123.60
HG-50.01-R	500MM TALL LARDER 4 SHELVES	311.20
HP-5.19	50MM TALL FILLETT FRONT COLOUR	40.80
BASE UNITS		
U-30.01-L	300MM BASE 1 SHELF DRAWER & DOOR	138.00
US-120.02	1200MM SINK BASE DUMMY DRAWERS	290.00
U-30.01-R	300MM BASE 1 SHELF DRAWER & DOOR	138.00
UP-5.72	50MM BASE FILLETT FRONT COLOUR	26.60
UE-110.01-L	1000MM CORNER BASE DRWR/DOOR L-BLANK	273.40
U-50.24	500MM BASE DRAWER & 2 P/OUT BASKETS	233.40
UH-60.01	600MM UNDER OVEN NEST WITH DRAWER	74.00
U-50.24	500MM BASE DRAWER & 2 P/OUT BASKETS	233.40
U-40.01-L	400MM BASE 1 SHELF DRAWER & DOOR	147.40
D-70.33	BASE DECOR PANEL	123.60
UR-32.03-R/L	325MM BASE END SHELF VENEERED	179.00
U-60.01-L	600MM BASE 1 SHELF DRAWER & DOOR	175.60
UE-110.01-R	1000MM CORNER BASE DRWR/DOOR R-BLANK	273.60
WALL UNITS		
D-60.01-L	600MM WALL UNIT 1 SHELF	116.40
D-100.01	1000MM WALL UNIT 1 SHELF	190.80
OG-60.00	600MM DOOR FOR COOKER HOOD (58CM)	83.20
O-50.01-R	500MM WALL UNIT 1 SHELF	110.80
OV-100.05	1000MM WALL UNIT 1 SHELF GLASS DOORS	232.80
OR-32.65-L	WALL END BALUST. VENEERED (LH)	160.68
O-60.01-L	600MM WALL UNIT 1 SHELF	116.48
OV-50.05-R	500MM WALL UNIT 1 SHELF GLASS DOOR	134.40
O-60.01-R	600MM WALL UNIT 1 SHELF	116.40
OTHER ITEMS (NONE REQUIRED)		

A section from one of the printed costings

THE effective use of micros in education depends ultimately on the software available.

Here **JENNY PREECE**, a member of the Open University's Microelectronics in Schools Project, describes the rationale behind "Educational Software", a package designed at the Open University which aims to assist teachers in their selection of software.

EDUCATIONAL Software is designed to give some insights into the nature of educational software and how it is produced. The aim of the pack can be paraphrased as helping teachers to become critical selectors of software.

Its other aims include:

- Providing an appreciation of fundamental programming concepts.
- Introducing some principles underlying educational software design.
- Helping to develop an appreciation of which kinds of software are suited to which learning tasks.

The only prerequisite for this pack is to have studied either the Apple edition of the Micros in Schools Awareness pack or have equivalent experience.

Educational Software is designed to address teachers' needs.

There is widespread agreement that there is very little good educational software, and that teachers have difficulty in selecting it from the heterogeneous range of available software.

While many teachers claim that what they need is more technical skills, we believe that being able to judge the educational value of software without being prejudiced by technical wizardry and glossy graphics is more valuable.

This depends on developing a sound understanding of the principles underlying the program which in turn requires an appreciation of program structure and design, and of the potential of the microcomputer.

Briefly, the route through the Educational Software pack is as follows:

Case study: Looking in detail at one educational program.

Programming: How educational software is designed and written, introducing programming concepts, mainly using Logo.

Styles of educational software: Introducing different kinds of educational programs, how they're written, and what they're used for.

Selecting Educational Software: Developing criteria for selecting educational software, by looking at commercial

Teaching teachers how to pick the best software

software.

Developing educational software: Examining how educational software is produced in Britain.

The key issue in our rationale for including this material and in following this particular route is the need to develop teachers' understanding of the potential role of the micro.

This potential can be viewed as operating on two levels.

Firstly, an understanding of the different types of educational software, the pedagogical principles underlying them, and the role that they play in the learning/teaching process – this is the micro level.

Secondly, an understanding of the potential of an individual piece of educational software which involves an understanding of both program structure and design, and some fundamental programming principles. We can think of this as the macro level.

While many people would argue quite rightly that it is the former which is important, we believe that it is not possible to

develop a sound understanding at the micro level without a good feel for the macro level.

The pack begins with examining the structure of an educational program and learning some fundamental programming concepts.

In order to support these initial activities it made sense to use Logo – although to some of our critics the most obvious language to use at the time, 1982, was Basic, which many teachers had some familiarity with.

However if teachers are to gain an understanding of what is possible in an educational program they need to appreciate which parts of the program reflect the underlying educational design and which are manifestations of how the program is written.

Logo was designed for novices, and its readability and modularity makes it ideal for writing programs which teachers can dissect in order to examine the structure.

This, in fact, is exactly what one of the first activities in the

pack consists of.

The teacher is given a partly-written Logo simulation program and guided through a series of exercises.

In these she examines and modifies the procedures that make up the program and in so doing develops an understanding of the role that each procedure plays in the program, and gets a feel for how big a job making various different modifications would be.

Several sampler programs are included which illustrate different types of software, and are also written in Logo so that teachers can examine their underlying structure, and can relate it to the educational role of the program.

In the final section of the course, the two "levels" of looking at the potential of the microcomputer – macro and micro levels – are brought together.

Three commercial programs are provided for the teachers to run and examine.

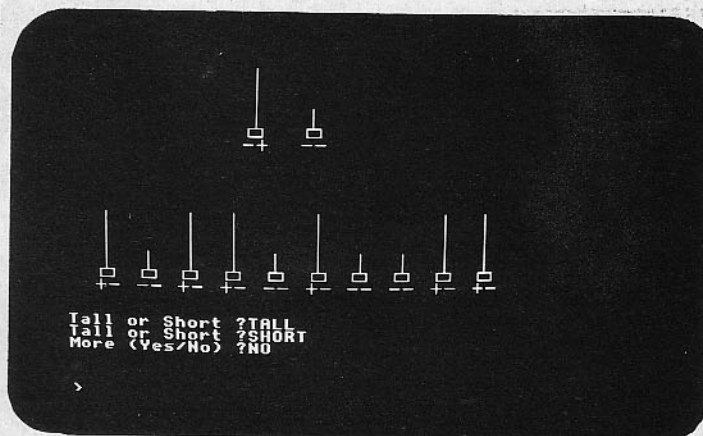
These are not seen as exemplary programs, but were chosen to illustrate a diversity of styles, covering simulation, drill and practice and a mathematical gaming environment, user interfaces, and technical diversity from fancy graphics to nothing.

Through this series of guided activities students are helped to bring together their micro knowledge and their macro knowledge in a very practical way to develop criteria for selecting educational software.

The Apple edition of Educational Software costs £50 and contains approximately 40 hours of independent self-study material.

In order to run the pack the purchaser will need an Apple II or IIe and Apple Logo.

The contents of the pack include a study book, reader, two activities books and handy trouble-shooting guide, three commercially produced software packages with accompanying documentation, eight programs written in Logo or Basic, an audio cassette and audio notes and an optional video of approximately 35 minutes with video notes.



Mendel is the program that introduces software editing and design



Alternative INPUT routines

I AM employed as a pharmacist in a pharmacy in Deventer, Holland. We prepare a lot of the medications which we dispense ourselves, out of raw materials.

Of course this must be done with the utmost care. For this purpose an Apple II Europlus 48k RAM, two 5¼in disc drives DOS 3.3, printer and three electronic Mettler scales connected via a special interface are used.

The software is written in Applesoft Basic. Since this stock preparation program contains many options it has become quite large, about 20k.

In the earlier versions of this program, input from keyboard or disc was done with the INPUT statement. This solution appeared to be unreliable. If the right keys were pressed one could upset the program very easily:

- The INPUT statement does not accept commas or colons.
- Ctrl-X ends the input.
- Ctrl-C at the first position ends the program.
- One can introduce the bell character in a string with Ctrl-G.
- One can lose one's way on the screen with too many backspaces, ESC-A, -B, -C, -D, or ESC-I, -J, -K, -M, or Ctrl-J.
- After pressing Return the rest of the line from cursor to the right side of the screen is cleared.
- One could enter a string that is either too long or too short.
- One could enter alphabetical input, when numerical is wanted and vice versa.
- It is even possible to enter

```
DOS commands in INPUT
strings like Ctrl-D + CATA-
LOG, Ctrl-D + INITHELLO,
Ctrl-D + IN #6, Ctrl-D +
PR #1, etc, which may
become effective when the
string is PRINTed. You could
try these entries with this
little program:
```

```
100 HOME
110 PRINT: INPUT ";A$
120 PRINT: PRINT A$
130 GOTO 110
```

Most of the mentioned problems arise because the input string isn't checked by Applesoft before a Return is given. Some of the problems can be filtered out with program lines directly following the INPUT statement.

Of course this checking of the input string must be done after every INPUT statement. To avoid many identical checks it is more economically done in a special input subroutine.

Some alternative solutions have been proposed that do not have the error possibilities I have mentioned (ref. 1, 2, 3).

They are all based on an input subroutine that separates every character by means of a GET statement and after approval adds it directly to the string, or saves it into a special input array variable and after completing

Error eradication, easy formatting and simple testing go hand in hand in A. REICHER's program

the entry assembles the input string.

Control characters are not allowed in the string, but are used to activate some special editing features.

In my stock preparation program this approach appeared to be awfully slow, regularly forcing the Apple into the housecleaning-old-strings process. This may take up to several seconds, depending on the total amount of defined strings.

Unfortunately large programs tend to use a large number of defined strings and at the same time with less free string space available.

The explanation for this process is quite simple. Every time a string manipulation is executed during the program, such as:

```
INPUT A$,
GET = B$,
A$ = B$,
A$ = B$ + C$,
A$ = LEFT$(B$,5),
A$ = MID$(B$,5) + RIGHT$(
C$,5)
```

a new A\$ is created and added to the string space.

A short calculation shows the implication of the aforementioned string manipulations for a GET-type input subroutine. If we enter a string of N

characters long, the string space used is at the best:

$$N \text{ (from subsequent GET\$) } + 1 + 2 + \dots + N - 1 + N \text{ (from concatenation) } = N + N*(N+1)/2 = N*(N+3)/2.$$

So for entering a string of 10 characters this figure is 65 bytes, for a string of 40 characters 860 bytes and for a string of 100 characters 5k. Not to mention, if we use the maximum string length of 255 characters 32k!

Of these calculated string concatenation overheads only 10, 40, 100 and 255 bytes respectively hold the final string.

Besides, we haven't even yet taken into account the (max) $3 \times 255 = 765$ extra bytes used for the array descriptor which holds the approved string characters, and the time it takes for Applesoft to actually concatenate the string, about 2 seconds for a 255 character string. Not to mention the overhead and speed loss if we use an input routine with a lot of substring manipulation (ref. 4).

However the principle of GETting a keystroke, testing it and, depending on its Ascii value, performing the task, incorporating it into the string or discarding it is good, but the string concatenation overhead is murder.

Another problem arises – which none of the referenced authors mentions – when we have accepted characters like commas, colons or double quotes into a string, or when we composed a string of only space characters or with leading spaces, WRITE them to disc in a textfile and then try to retrieve them again from disc.

The standard solution for retrieving strings from a textfile is the use again of the INPUT statement. All these examples will appear to produce erroneous results.

Conclusion: It was thinking time again.

The new strategy to follow was as follows:

- Find the starting address of the string in memory.
- Get a character from the keyboard.
- Test it. If it is an assigned control character, perform the task, if it is a valid string character, POKE it directly into the right place in the string and PRINT it at the right place on the screen. Discard any other character. This means that at any moment what you see on the screen is what you get, no matter where the cursor is in the string at that moment.
- It should also work with input from disc, so we must be able to READ any character from a textfile. This implies a solution other than using an INPUT statement.

In the listing the final input routines are presented within a simple demo program. I will cover both keyboard input and disc input separately in detail.

For the sake of speed the keyboard input subroutine is placed at the beginning of the program. The rest of the program starts at line 30000.

References:

1. G. Kielian: *Bombproof Data Entry – Creative Computing Nov. 1980.*
2. S. Luca: *Bombproof Data Entry – A Revision – Creative Computing May 1981.*
3. C. Glenn: *Absolutely Bombproof Input Routine – Creative Computing Dec. 1983, p. 251-252.*
4. R.J. Beck: *Input Line Editor – Creative Computing Dec. 1983, p. 255-259.*

	INPUT	GET	POKE
Max. string length	239	255	255
Acceptable characters from keyboard	limited, see intro.	all characters, selected as desired	
Acceptance speed of char. from keyboard *	18 per sec	18 per sec	18 per sec
String space used with a string of N characters	N	(N+3)/2	2N
Reading speed of char. from a textfile **			
A. sequential	230 per sec	46 per sec	188 per sec
B. random access	194 per sec	45 per sec	170 per sec

*) Tested with the REPT key.
 **) Tested with a program reading 100 strings of 30 characters each.

Table 1

The first thing the main program has to do is to introduce the working string of the subroutine, IN\$. Because it is the first string declared by the program, it is also the first variable in the variable descriptor table.

So the starting address of the string descriptor IN\$ is also known, and variable X holds this address. Both tasks are performed in line 30010.

Next, a string is defined from which during the program "empty" strings are derived. This X\$ is defined in line 30030, consists of only spaces and is of sufficient length to accommodate the longest string present in the program.

Before entering the subroutine a few variables must be set correctly:

V : Vertical tab position on the screen, 1-24.

H : Horizontal tab position on the screen, 1-40.

I : Input status flag: 0 is all characters, 1 is only numerical input.

A typical input line looks something like line 1080. V and H – and I when needed – are set correctly, and with the instruction IN\$ = A\$(B) the variable X is made equal to the starting address of the string descriptor of A\$(B).

The actual input routines are the lines 40-160. I have

equipped it with only a basic set of possibilities:

- RETURN** = Accept string and proceed.
- ESC** = Go 1 step back in program.
- /** = At first pos: clear string and proceed.
- >** = 1 pos forward in string.
- CTRL-G** = 5 pos forward in string.
- <--** = 1 pos backward in string.
- CTRL-F** = 5 pos backward in string.

The first thing (line 40) the input routine does is to establish the upper and lower Ascii value acceptance limits. Normally these are from 32 to 95, but if the input status flag is 1 these values become 45 and 57 respectively, excluding character 47:

	normal	numerical
AL = lower Ascii limit	31	44
AU = upper Ascii limit	96	58
AN = non-valid ASCII char.:	0	47

The second thing it does (line 50) is clear the keyboard strobe to prevent accidental exit from the input routine, and reset the position in the string pointer P and the mutation flag M to zero. The length L minus 1 and starting address S of the input

string IN\$ are calculated.

Next (line 60) the input string is displayed in inverse mode on the screen at the position given by V and H. The displayed string functions as its own cadre from which the cursor can't get out, (lines 70 and 80).

Then a character is asked from the keyboard with the CALL 2054 and transferred to variable A in line 90. If we enter a slash, Ascii 47, at the first position the string is cleared – filled with spaces – and we leave the routine.

If it is a valid string character it is incorporated in the string and displayed on the screen (line 100) if it is a valid control character, perform the function, otherwise discard the input character and ask for another (lines 110-150).

The routine is left when either the Esc or the Return key is pressed (line 150) no matter where the cursor is at that

moment – *what you see is what you get* – the input string turns to the NORMAL status on the screen and the input status flag and nonvalid Ascii character value are reset to zero (line 160).

When leaving the input

routine two signals are available, A holds the last character entered from the keyboard and if M = 1 then changes have been made to the input string.

The CALL 2054 in line 90 is a simulation of the GET statement, with the advantage that no new strings are added to the string space every time a character is got from the keyboard. It uses the monitor ROM RDKEY subroutine and returns with the character in memory location 513. The starting address of this little routine is 2054, hex \$0806.

This is in the first line of the Applesoft program line:

0806-	20 OC FD	JSR	\$FDOC	jsr rdkey,
0809-	29 7F	AND	\$7F	turn off high bit,
080B-	8D 01 02	STA	\$0201	store at \$0201,
080E-	60	RTS		and return

How do we get it there? First, introduce as the first line of the program.

10 REM*****

Go to the monitor:

1CALL -151

Replace the asterisks by the routine:

***806:20 OC FD 29 7F 8D 01 02 60**

Enter Basic again:

***3D06**

Finish the rest of the program. Done! This explains the somewhat strange appearance of line 0 in the demonstration program listing.

For disc input we are going to use the monitor routine GETLN

at \$FD6F, decimal 64879. Since we have already filtered out all control characters at the keyboard input routine, we can safely use it to simulate the Applesoft INPUT statement.

This routine collects characters with bit 7 = 1. Since Applesoft works with bit 7 = 0, the POKEs in line 30020 take care of that problem.

Line 40000 and the next present a typical disc input routine. The idea behind it is similar to that of the keyboard input routine—OPEN and READ parameters are as in normal DOS.

Then the starting address of the receiving string is determined and the starting address is POKEd in the area of DOS where the INPUT command is handled, 42585-42586.

Normally these locations hold the starting address of the

input buffer, \$0200. So the input string is put directly at the right place in memory and not via the input buffer.

In Table I the basic characteristics of the three mentioned input solutions are summarised.

As you can see in the demonstration program, only the INPUT statement is replaced by alternative routines, while OUTPUT to screen or disc still uses the PRINT statement, with no problems at all.

The routines presented for keyboard and disc input offer the advantages of the availability of all Ascii characters, no error possibilities, easy formatting and testing, at the expense of a slightly higher string overhead, and a slight speed loss when accessing textfiles. Still it is flexible enough to customise it to your own needs.

```

0 REM h PLOT
10 REM
/-----
- /
11 REM / THE FINAL INPUT
ROUTINES /
12 REM / BY A.REICHER
/
14 REM
/-----
- /
20 GOTO 30000
30 REM
31 REM KEYBOARD INPUT
32 REM
40 AL = 31:AH = 96: IF I = 1
THEN AL = 44:AH = 58:AN =
47
50 POKE - 16368,0:P = 0:M
= 0:L = PEEK (X + 2) -
1:S = PEEK (X + 4) * 256
+ PEEK (X + 3)
60 VTAB V: HTAB H: INVERSE
: PRINT IN$:
70 IF P < 0 THEN P = 0:
PRINT CHR$ (7);
80 IF P > L THEN P = L: IF
L > 0 THEN PRINT CHR$
(7);
90 VTAB V: HTAB H + P: CALL
2054:A = PEEK (513): IF
A = 47 AND P = 0 THEN
FOR A = S TO S + L: POKE
A,32: NEXT :M = 1: GOTO
160
100 IF A > AL AND A < AH
AND A < > AN OR A = 32
THEN PRINT CHR$ (A);:
POKE S + P,A:P = P + 1:M
= 1: GOTO 80
110 IF A = 8 THEN P = P -
1: GOTO 70
120 IF A = 6 THEN P = P -
5: GOTO 70
130 IF A = 21 THEN P = P +
1: GOTO 80
140 IF A = 7 THEN P = P +
5: GOTO 80
150 IF A < > 13 AND A < >
27 THEN 90
160 VTAB V: HTAB H: NORMAL
: PRINT IN$:I = 0:AN =
0: RETURN
1000 REM
1001 REM DEMO PART
1002 REM
1010 V = 5:H = 36:IN$ =
LEFT$ (X$,1): GOSUB 40:
IF A = 27 THEN 1010
1020 IF IN$ = "Y" THEN
GOSUB 40000
1030 B = 0
1040 IF B < 0 THEN B = 0
1050 B = B + 1: IF B = 11
THEN 50000
1060 VTAB 7: HTAB 1: PRINT
" ": VTAB 7: PRINT B:
VTAB 7: HTAB 4: PRINT
"ALFANUMERICAL"
1070 IF INT (B / 2) * 2 =
B THEN I = 1: VTAB 7:
HTAB 4: PRINT ":
NUMERICAL "
1080 V = 9:H = 1:IN$ =
A$(B): GOSUB 40:A$(B) =
IN$: IF A = 27 THEN B = B
- 2
1090 GOTO 1040
30000 REM
30001 REM ENTRY PROGRAM
30002 REM
30010 IN$ = IN$:X = PEEK
(106) * 256 + PEEK (105)
30020 POKE 42582,41: POKE
42583,127
30030 X$ = "
"
30040 D$ = CHR$ (4)
30050 FOR A = 1 TO 10:A$(A)
= LEFT$ (X$,10): NEXT
31000 HOME : HTAB 5:
INVERSE : PRINT "THE
FINAL INPUT ROUTINES
DEMO": PRINT : HTAB 13:
PRINT "BY A. REICHER":
NORMAL : VTAB 5: PRINT
"READ STRINGS FROM DISK
<Y>ES/<N>O:"
31010 VTAB 17: PRINT
"<RETURN> = TO NEXT ITEM"
31020 VTAB 18: PRINT "<ESC>
= TO PREVIOUS ITEM"
31030 VTAB 19: PRINT " <--
= 1 POS BACK IN STRING"
31040 VTAB 20: PRINT
"<CTRL>-F = 5 POS BACK IN
STRING"
31050 VTAB 21: PRINT " -->
= 1 POS FORWARD IN
STRING"
31060 VTAB 22: PRINT
"<CTRL>-G = 5 POS FORWARD
IN STRING"
31070 VTAB 23: PRINT " /
= CLEAR STRING (POS 0)"
31080 GOTO 1000
40000 REM
40001 REM DISK INPUT
40002 REM
40010 VTAB 15: HTAB 12:
FLASH : PRINT "READING
STRINGS": NORMAL
40020 PRINT D$"OPENFILE"
40030 PRINT D$"READFILE"
40040 FOR A = 1 TO 10
40050 IN$ = A$(A): POKE
42586, PEEK (X + 4): POKE
42585, PEEK (X + 3): CALL
64879:A$(A) = IN$
40060 NEXT
40070 PRINT D$"CLOSE"
40080 VTAB 15: HTAB 12:
PRINT " "
NORMAL
40090 RETURN
50000 REM
50001 REM DISK OUTPUT
50002 REM
50010 VTAB 15: HTAB 12:
FLASH : PRINT "WRITING
STRINGS": NORMAL
50020 PRINT D$"OPENFILE"
50030 PRINT D$"WRITEFILE"
50040 FOR A = 1 TO 10
50050 PRINT A$(A)
50060 NEXT
50070 PRINT D$"CLOSE"
50080 GOTO 30050

```

THE book VisiCalc Made Simple is plentifully illustrated, and the text has a clear, precise tone, showing the academic background of its author.

Its "voice" avoids the over-familiarity of American texts.

While the demise of Visicalc reduces its potential market, it is recommended to anybody who has Visicalc and has not found the reference manual helpful.

It has many illustrations of specific applications of spreadsheet functions, and covers areas such as /PF command files that are not mentioned in most "template" books.

It starts with the basics of cursor control, using the spacebar command of Apple II+ Visicalc.

The clear pictures are all based on a 40 column screen and show both formulae and answers.

Be sure to read *all* the instructions if you are following the examples, for he doesn't repeat himself.

By page 18 you are already printing – always an exciting point for the first-time user.

He explains that you save the formulae rather than what you see on the screen at any given instant.

Exercises at the end of each chapter prevent the fast reader from becoming over-confident, even if their tone is that of the exam – "Do this ... What happens? ... Explain".

Chapter 2 explains forward and circular references.

Memory expansion boards are mentioned, but not that one usually has to buy extra software and one has to consider whether they are compatible with your make of 80-column card.

Chapter 3 introduces the /Replicate command.

From my own experience giving training courses, I can only /Reiterate the /Recommendation to /Repeat the /Ruddy thing often to /Reinforce it ...

I'm not happy with the use of the expression "Relative to its position". I find it easier to explain the concept using the Lotus 1-2-3 or Multiplan definitions, where a formula itself is expressed in relative terms, rather than the action of copying the formula.

Chapter 4 introduces Global

The lucid line to VC success

and Local Format commands. By this time users want to tidy things up, having succeeded in getting the software to give the right answers.

It gives a brief description of the /X command.

Chapter 5 describes the functions.

It documents, with a useful budget/variance example, the peculiarity of @COUNT() which requires a RANGE, even if that is one cell – (A1...A1). Do you know that Lotus 1-2-3 CAN'T do this?



An unusual use of @ABS() is described. He introduces date arithmetic. I look for little things like that and the unusual /X and /PF features to test the comprehensiveness of a book – this passes.

However, I think that my "indexing" technique to protect @LOOKUP tables from returning spurious results is better than the method he describes of "bracketing" the values with incrementally-different values.

The examples of the logical functions @AND, @NOT, @IF,

@ISERROR are particularly good and include columnar analysis, debt ageing, and production scheduling.

Chapter 6 introduces the editing commands /E, /F, /D, /I. He describes /SL overlays, which with the /F command naturally leads to the use of /PF files for consolidation.

In mentioning the use of a "border" overlay to give printed row and column labels, I would have thought that to see the formulae in print, one should /SS,S1 the sheet in its original form rather than the overlay form.

Chapter 7 hints at what is available beyond VisiCalc both in spreadsheet and modelling language terms.

There is an impressive list of references and – in true non-competitive academic style – other teaching aids. And, finally, a decent index.

Patrick R. O'Beirne

Title: VisiCalc Made Simple
Author: Thomas M. O'Donovan
Publisher: John Wiley
Price: £7.95

Right approach to software

MOST books which start off "Choosing Bla Bla Software for your Superwizzo Mk 23a" normally have a software package which the writer probably has in mind after its extensive use over the last x months.

The end result is that eventually he will recommend that product X, Y or Z is the only one to use.

The big trouble is that with better, more comprehensive, and recently integrated software packages coming on to the dealer shelves, the book is no sooner printed than it is out of date.

Choosing Accounting Software for your Microcomputer is different, as it teaches the evaluation method of approach in choosing your software.

Gary Simon does not actually say which software package he would recommend you to buy

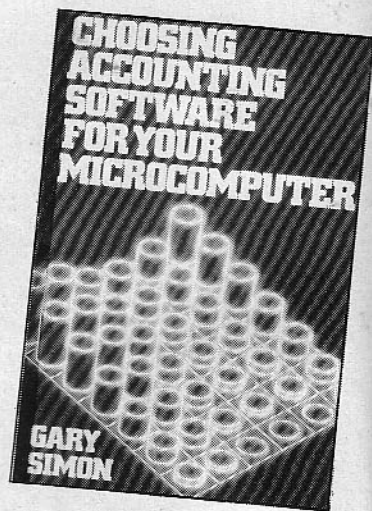
and the book is therefore not machine-dependent.

He is correct in suggesting that you buy your software first and then choose a machine on which it will run.

The book is logically divided into checklists covering software supplier, sales, purchase and nominal ledger requirements, with each of the 69 sales, 37 purchase and 43 nominal ledger checklist items covered in depth.

The correct emphasis is placed on the consultation, involvement and training of your staff who are going to use the package – and even such subjects as evaluating the need of computerising in the first place, having the right amount of maintenance for your equipment and so forth.

I have read the book carefully and Mr Simon seems to have covered all aspects of choosing



and using sales, purchase and nominal ledger accountancy software on any computer.

Recommended if you have just started out on the sometimes-traumatic experience of choosing the right accountancy software for your company.

Peter F. Wilson

Title: Choosing Accounting Software for Your Microcomputer
Author: Gary Simon
Publisher: Collins
Price: £7.95



Out-fox the enemy— in glorious 3D

SKYFOX is a 3D flight and combat simulation, fought in the future. You are the hero of the game, Sky-Pilot and you get to pilot Skyfox, a very sophisticated piece of flying machinery and state-of-the-art electronics.

Your mission is basically the same every time — you have to defend your main base and usually other peripheral bases, which contain your family and friends, as well.

When the game starts you can choose from 15 different scenarios and five ranks, ranging from Cadet to Ace-of-the-Base. The first seven scenarios are for training purposes, and provide various combinations of tanks, planes, etc, for you to destroy.

In these training scenarios your plane is attacked, but the base isn't. When you've destroyed all the enemy units you just get more to practise on. The other eight scenarios are the real thing, the enemy actually advances towards your base(s) and if you don't act fast you are doomed.

Your plane is equipped with all sorts of advanced electronics. The main control panel looks quite complex at first, as

you have plenty to look at. You have an overhead radar on the panel itself, which can be switched to forward radar by pressing the space bar.

You also have a computer available. When it is activated, a window appears, in real Mac style, which covers most of the screen except the control panel. The computer really lets you see what's going on and helps you plan your moves accordingly.

Skyfox requires an element of strategy on the harder levels. There is a tendency to use the game just as a shoot-em-up, but that kind of approach usually means that you lose control of the situation, and you won't last long, especially if your main base is destroyed.

If your main base remains intact, you can land and refuel there. You also have some heat-seeking missiles and guided missiles, but once used they cannot be replaced. Skyfox

is very fast. The makers claim 0 to mach 4 in four seconds!

The graphics are very nice, as they are solid 3D, which look rather like Flight Simulator 2, but much, much faster. The game looks excellent in colour, but I found it quite playable on a green screen too.

Control is by joystick only, and I found it very responsive. The overall flying effect is quite convincing.

The sound effects are reasonable on a standard Apple, but if you are lucky enough to have a Mockingboard you can enjoy much more dramatic effects, which add considerably to the realism as well as the superb title music — a must for any Mockingboard owner.

I discovered that pressing two secret keys when you're at the base will let you play Space Invaders. It sure plays a mean game of invaders, since you have no barricades and the

aliens are quite fast. Pressing the Escape key gets you back to Skyfox.

Probably the best thing about all this is the price, only £17.95. If you have any difficulty getting hold of it, you can deal direct with Ariolasoft, the UK distributor. Their address is Suite 105/106, Asphalte House, Palace Street, London SW1E 5HS.

With so much going for it, I feel that Skyfox will surely be a success.

Leon Seltsikas

*Title: Skyfox.
Author: Ray Tobey.
Publisher: Electronic Arts/
Ariolasoft.
Requirements: Apple II+, IIe
or IIc with 64k minimum and
joystick.
Optional: Mockingboard.*

RESCUE RAIDERS is yet another masterpiece from those hugely successful SirTech people. It is a strategic war simulation in which you command aircraft and ground forces advancing toward an enemy with the hope of destroying him.

The good thing about Rescue Raiders is that, unlike most strategy war games, you can see and control what happens directly.

You actually see your ground vehicles, soldiers and tanks trundling off towards the enemy in your quest for victory. This is much more exciting and enjoyable than looking at odd symbols on a map.

I suppose at this point you are wondering who you are up against. "Not those blasted green furry things from Sigma 4, again", I hear you cry.

Nope, this time you're coming to grips with a group of Time Terrorists from present-day Earth. These unsavoury lads have the idea that by altering the outcome of the Battle of Normandy their group can benefit in some way.

To do this they have to travel back in time, taking some 20th century weapons with them to help win.

You also have to travel back in time to try to stop them by doing the decent thing and laying your life on the line - as usual.

You must try to destroy their base at the far left of the terrain and the enemy will try to destroy yours at the far right.

There are eight battles in Rescue Raiders and you have to win them all or submit to total victory by the enemy.

The game uses very smooth, flicker-free, colourful graphics.

You control a fully-maneuvrable helicopter - you have three of these - using a joystick with two buttons. This takes a little getting used to.

By pressing different keys you can deploy tanks, men, engineers, anti-aircraft guns, demolition team vehicles and even buy extra helicopters if you have enough money to help you in your struggle against evil.

You can pick up men in your helicopter and then let them parachute out over enemy

Commanding view of war strategy

territory by pressing the spacebar.

You receive money depending on how long you stay in the battle area.

The screen scrolls left and right with a radar at the top of the screen.

An interesting feature of the game is that if some of your men or vehicles meet opposing ones they actually battle it out

automatically, which can be quite enthralling to watch, and the winning party continues on its way towards the opposing base.

A nice feature of the game is that before each battle you are shown a picture of Europe with a flashing star showing where the next battle will begin.

The game can in theory go on for a considerable length of

time, so a SAVE feature is available. Unfortunately you can only save one game on the program disc. There is a backup on the other side, but you can't save games on it.

The game contains all the usual convenience features except a sound on/off function. This is probably due to the fact that the sound in this game is virtually non-existent, which is a shame.

This is a 64k game and so if you are using an Apple II+ like me, you will need a 16k RAM card.

Jason W. Smith



*Title: Rescue Raiders
Authors: Greg Hale and Arthur Britto II
Publisher: Sir-Tech Software
Requirements: Any Apple II series with 64k*

ALL YOU COULD WISH FOR IN AN ADVENTURE

ONCE upon a time there was a nasty Queen who, as the law governing fairy stories dictates, had a beautiful and virtuous stepdaughter.

Now the Queen, being the crabby, cantankerous and malicious monarch that she was, decreed that no one should marry the fair Princess Morning Star until they first proved themselves worthy.

Naturally Queenie devised tests that would ensure that none of the macho knights that sought the Princess's hand would live to tell the tale. One besotted wretch was even sent deep into the Mines of Mendon, there to slay a Grue and drag the carcass up where all might see it.

Now any Infocom adven-

turer worth his or her salt could have told the poor sucker that he'd have been better off staying at home reading a good magazine like Apple User but no - off he went. Darkness soon overcame the hapless knight who, lost without a lamp, was soon devoured by a lurking Grue.

Since nobody ever survived the ordeals, the Princess was condemned to a life of loneliness and died without ever seeing a single knight make it to the finals.

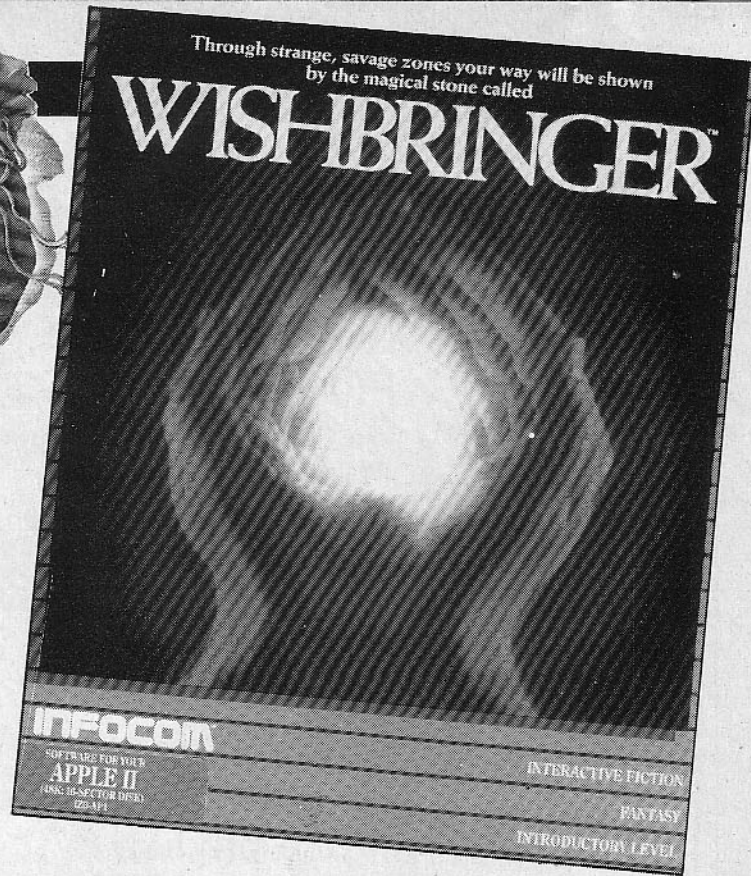
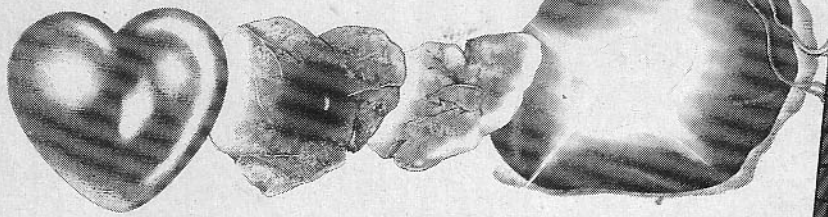
Many kingdoms later, a scholar happened to be rummaging through the ruined

tombs of monarchs and chanced across a glowing object amid the dust and decrepitude.

It was the Princess's heart, now long since turned to stone but shining brightly with the unfulfilled wishes of a lifetime. And that, dear reader, was how the Magick Stone of Dreams was discovered.

In Wishbringer, from Infocom, you begin not as a knight in shining armour but as a humble Post Office worker. The game begins atop a hillside in the coastal town of Festeron.

You may wish to explore the town before or after going



to the post office – the cinema, cemetery, police station, light-house, pleasure wharf, park, library and church are well worth a visit.

Your boss, Mr Crisp, is not the most pleasant of postmasters – when you first meet him, he's reading other people's postcards – and will tell you in no uncertain terms what he thinks of you.

Your first job is to deliver a letter to the proprietor of Ye Olde Magick Shoppe, way up on a cliff top on the other side of town. You can't afford to hang about too long – the Magick Shoppe shuts at 5pm and the game starts at 3pm.

Getting there can present a problem. The bane of all messengers in the shape of a vicious poodle blocks the main route, and even if you do

ing himself as the Evil One has kidnapped the old lady's cat and wants the Magick Stone in return.

The lady is evidently distressed. In a voice breaking with emotion, she whispers "Many seek to gain the Stone of Dreams yet few can imagine the price. For years I have fought to conceal it from the Evil One and others like her.

"My youth, my home and family all were forfeited for its protection. And now, now it claims my only companion".

Guess who's about to be entrusted with the Stone and the task of rescuing Chaos the cat? Right first time!

Before you know it, you're thrust out of the shop and into thick fog. When you peer through it, all you can see is the summit of Post Office hill. Only trouble is, there's now a massive tower where the Post Office used to be.

If you make it through the fog without falling to your death off the cliff, you'll find the whole scene has changed. Where birds once sang, vultures now croak. Trolls lurk. Your beloved town now seems to be in the grip of decay and despair.

And if you thought a nip on the trouser seat from a poodle was bad enough, what are you going to think about being chewed up by the enormous hellhound which has usurped the poodle's position?

Fortunately, the Magick Stone – Wishbringer – can help. With it, you can wish for advice, darkness, flight, freedom, luck, foresight and rain. You can only wish once for each, so must choose the occasion with care.

You must also have certain objects in order to make the wish effective. For example, and most appropriately, to wish for darkness you must first have drunk some Grue's milk.

Wishbringer is ranked as an introductory level adventure – the other Infocom gradings in

order of difficulty are standard, advanced and expert – which means it is more suitable for the apprentice adventurer. Nevertheless, even seasoned adventurers will find it a joy to play.

The package comes handsomely boxed with glossy manual, map and your very own Wishbringer stone.

Wishbringer is not intended as the sequel to the marvellous Sorcerer and Enchanter as some people thought it might be – that pleasure is still to come.

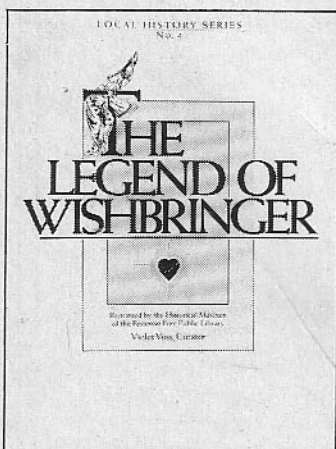
Yet once again Infocom has come up trumps. Wishbringer is filled with humour, excitement, detail and atmosphere.

It maintains its high standards and keeps Infocom in pole position as the best adventure publishers in the

world. Definitely not to be missed.

Bob Chappell

*Title: Wishbringer
Author: Brian Moriarty
Publisher: Infocom
Requirements: 48k Apple II family*



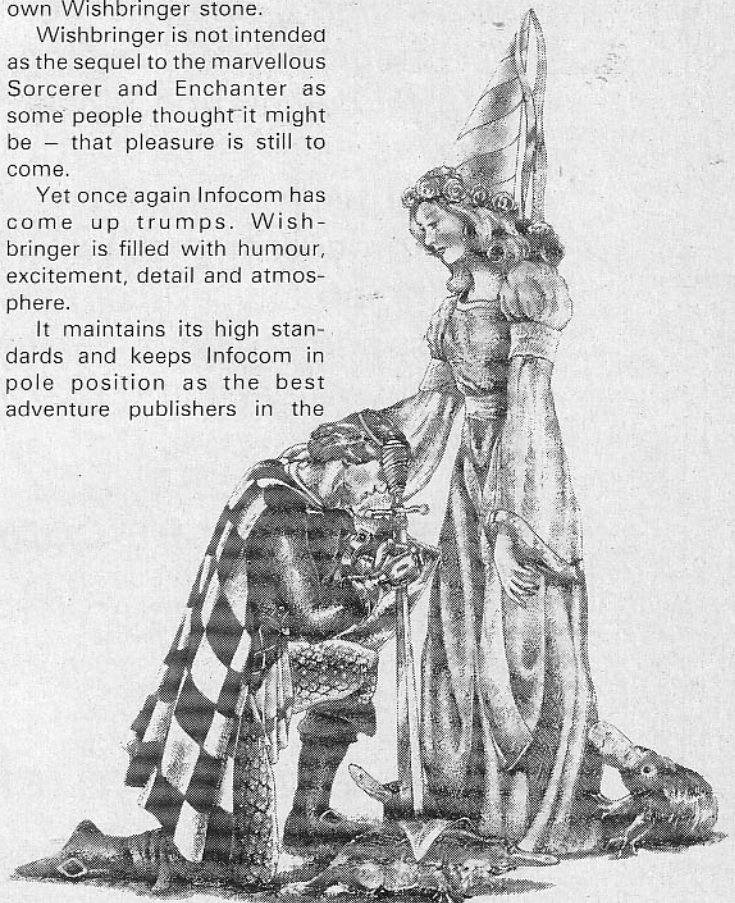
The "library book" from Festeron's Historical Museum

manage to pacify it the Pooch doesn't stay tranquil for long.

Once at the Magick Shoppe an old lady who is as fragile-looking and pale "as a faded signature in an antique book" receives your delivered letter.

She is clearly upset when she sees the writing on the envelope: "It has been a long, long time since I last saw this handwriting. Hoped I never would again".

The letter turns out to be a ransom note. Somebody sign-



Accessing two screens under Apple Pascal

MAX Parrott was entirely correct in his diagnosis of the problems encountered in any attempt to use both 40 and 80 column screens under Apple Pascal.

If the standard 40 column screen is being used the lowest part of memory used by the system is \$0C00 – see page 255 of the Pascal OS manual.

However if an external terminal or 80 column card is in use, the second text page can be used by the system, that is from \$0800.

The reason that the first text page cannot be used is that certain peripheral cards and the Apple Pascal system itself use the fringe areas round the screen, as is the case with under DOS – see page 134 of the Apple II reference manual.

Hence only 1k, not 2k, is gained with an alternative screen. To gain the extra 1k the BIOS has to be fiddled with. I've done it, and it's not an easy task.

The code that achieves this checking is shown right. It is in two parts. A section of the BIOS checks for a card and sets the variable SCRMODE accordingly. Later in the boot process part of the p-code interpreter checks SCRMODE, and sets the new pointer – see page 255 of Pascal OS Manual – as appropriate.

I hope that this clarifies the position. By patching the interpreter so that the BEQ FORTY becomes a JMP FORTY – pinching a byte from the next instruction – it should be possible to access both screens without any problem, as the system will not try to access the second text page. – **Stuart Bell, Cambridge.**

Dublin Users' BB

WOULD it be possible for you to put details of my bulletin board in your magazine?

My BBS is the first to be set up in Ireland and it is called DUBBS – Dublin Users' Bulletin Board System.

It is running on an Apple II+, Super Serial Card and the Miracle Technology WS2000

```

;in the BIOS      Assuming Apple Pascal 1.1
LDA  SLOTYP5+3   ;load the type of card in slot 3 for initialising
LDY  #030        ;used as offset for I/O on card
JSR  GENIT       ;call general initialisation
CPX  #00         ;if X is zero, there WAS a card in the slot
BNE  PASBOOT    ;if not zero, leave SCRMODE unchanged
LDA  #004       ;card in slot 3 so set SCRMODE to 4
STA  SCRMODE
PASBOOT JMP JPASCAL ;enter the Pascal system

;in the boot section of the interpreter
LDA  SCRMODE     ;SCRMODE is 4 if card in slot 3 (eg.80-col)
AND  #004
BEQ  FORTY
LDA  #008       ;external terminal (can be 04 with modified BIOS)
STA  NPH        ;set new pointer's high byte to use 2nd text page
BPL  LDNPL
FORTY LDA #00C   ;no external terminal -- use 40 column screen
STA  NPH        ;set new pointer's high byte above 2nd text page
LDNPL LDA #004
STA  NPL        ;set its low byte
    
```

modem.

The main problem I have at the moment is storage – I only have one disc drive.

DUBBS is on-line from 8pm to 8am Monday-Friday and 24 hours at the weekends using the V21 protocol. It is also on-line from 1.30pm to 4.30pm on Wednesdays using the Bell 103 protocol.

I would also like you to publish the fact that I am extremely grateful to Quentin Reidford, Maurice Cohen, Andrew Kinsella, Mike James, Tony Game and Ewen Wannop. – **Stephen Kearnon, Dublin.**

Sluggish Applewriter

I HAVE used an Apple IIe for two years now and find it hard to fault and I use Applewriter IIe – by far the best all-round word processor – for all correspondence.

But I have one small problem.

Every program I use loads perfectly error-free every time

except Applewriter and this usually takes two or three boots before it will load correctly – which is annoying because it is the program I use most.

Most times the disc just keeps running with Apple at the top of the screen, but sometimes it stops with a row of letters on the screen.

I have written three times to Pete and Pam Computers who supplied the program but they have not replied.

I have also asked two of the Edinburgh dealers for help, but they do not know the answer.

Finally, one small point in Applewriter – the screen heading shows a Position number which I find of little help, whereas a line number would be invaluable. – **F.S. Robertson, Kelso, Roxburgh.**

● This loading problem could be due to the disc or to the drive.

Sometimes if drives are a little slow or are not correctly aligned for track 0 they will function perfectly with DOS but not with a protected DOS.

On the other hand I have had discs which will not rotate correctly in their envelopes until

slightly rotated by hand.

This is sometimes caused by the hole in the disc becoming frayed at the edges. A reinforcing ring – if one isn't present – may help.

You don't say whether the other programs are protected nor how old your Appewriter is – if I were you I would get the drive checked.

If you have two drives, swap them over and try again.

You would do well to try your Appewriter disc on another Apple system and to take a back-up copy using one of the well-known bit-copiers.

Because Appewriter only formats the text at print time it cannot easily give a line-number rather than position. – **Max Parrott.**

The future with Apple

WHAT do I want from Apple? In a sentence, I want to remain with Apple, but is Apple going to provide me with the product I want?

It is two years since I got my first Lisa – yes, I was one of the first Lisa owners in the UK. Since then I have bought another and a Mac 512k.

Before I attempt to define what I want, let me explain what my needs are.

I am a professional accountant involved in the management of a number of businesses. High volume data processing is looked after by the mainframe IBMs and the mini DECs. The applications that I am interested in and need are the normal executive applications which include:

- Spreadsheets.
- Graphics.
- Word processing.
- Database.
- Communications – electronic mailbox.
- 3270 IBM emulation and

VT100 emulation.

□ Simple accounts.

All of this is nothing very special and none of these applications involve a large database. If they did they would be off my desk and into the mini.

It is worth emphasising the relatively straightforward requirements. Regression analysis may be used in universities and in large corporations, but in most medium size businesses it is simple cash flow statements that are required.

Again, the operation of these programs must be simple to master and share common instructions. We all remember the days when the command to exit from a program could be anything from E to X.

This aspect is vital for executive workstations as the applications are not going to be used continuously, and in fact several months may elapse before an application is used again.

Wordstar has a complicated set of command characters which require a lot of learning, but it does not matter much because the operator is using word processing all the time and once she becomes familiar with the commands she has not got the opportunity to forget them.

So the Mac concept is exactly on the right lines and deserves more success than it has so far received. Any Mac spreadsheet user will tell you what it is like returning to SuperCalc without a mouse — back to the Stone Age.

So what do I want? I want a better Mac. There is nothing wrong with the model except that it is just not as good as it should be.

I want a Mac with a double density micro disc drive increasing the storage from 400k to 800k — nothing I would have thought technically very difficult, but it's got to come. Jazz on two discs has got to be wrong.

I want a 20 megabyte Winchester, I could manage with 10 but I have always found that you usually finish up with twice the storage you thought you first needed. This Winchester has to have something special about it. Ideally, it

should live inside Mac, but if that is not possible why not design it so that it can sit under it.

If it is not put inside Mac it has to be small enough to transport. Most important of all, it has got to be fast, files must open quickly and the data interchange must be speedy.

I have not got time to look out of the window while a file is being opened.

There has to be a way to improve the speed of the printer, and the Mac must be fully operational while it is printing without any degradation of speed.

An upgraded version of Mac must be capable of moving into the emulation mode quickly and simply — no hunting for discs, no sorting out preferences, just a touch of a button or key puts you into or out of IBM 3270 terminal emulation.

The software is now nearly there. What I have seen of Jazz makes me think it is a real knockout. Microsoft Excel will, I am sure, be a worthy competitor. The Omnis III database is fantastic and I would not be without it.

Emulation and communication I am sure will be cracked. What is now happening is that the hardware seems to be falling behind the software. That has to be put right, especially as the technology is not difficult.

Finally, what is Apple going to do about its Lisa users? Is it true that the 7/7 software will no longer be supported? Can 7/7 data files be converted to Jazz? What option if any will be available to convert Lisa to Mac in a more elegant fashion than at present?

The quality of Apple's marketing seems inversely proportional to the quality of its products. Apple must have a database of its users. Why doesn't it use that database to mail quarterly news bulletins of new products and details of third party software?

Surely this would be a much more effective budget spend than money which has been poured into the recent advertising and promotion campaigns.

Do you really sell the Mac concept through half page

advertisements in the Sunday Times? I very much doubt it.

You may excite a little interest, but the best Apple salesmen are the users. Why not keep them interested and confident about the future by letting them know what's happening? — **B.F. Trent, Chalfont Management Services, Chesterfield.**

Data security

I AM starting to use AppleWorks for a personnel database system. Is there a password facility that would comply with the computing regulations of the Data Protection Act?

As the Data Protection Act originated in the USA where Apples come from, I wonder if this problem has been overcome in some way, or whether any of your readers has any idea of how to comply with the Act. — **M.L. Stringer, Folkestone.**

● I don't think that a password is enough protection because the data is on the disc in machine readable form.

The answer surely is to lock all data discs away when not in use and make sure that the screen and printer output cannot be read by people casually passing by.

We would be very interested to hear readers' views on the Act and their means of complying with it. — **Max Parrott.**

Splitting programs

I HAVE an Apple IIe with an extended 80 column card (Apple Card) which at present is very under-used. I have been writing a program for my own use, which uses graphics and large arrays.

The first problem, and a recurrent one reading the letters page, is that the graphics page interferes with the program.

I have solved this by loading the program above the graphics

page and by moving DOS on to the language card.

However when the program has a large amount of data and hence large arrays I get the dreaded "Out of memory" error.

It would be useful either to use the memory space below the graphics page and split the program (I know this can be achieved but can you advise me how?) or secondly, use the auxiliary graphics page on the extended card instead of the main graphics page, so leaving the full memory available for the program.

Has anyone achieved this?

It would seem sensible in view of the above problem with graphics page clashing with the program that someone should produce a simple graphics card with two pages of normal graphics and I would have thought that this would be moderately priced. Is this possible or does one exist already? —

F. Creasy, Epsom.

● I have sent some information on "splitting" Basic programs — any reader who wants such a utility can have a copy if they send in a disc. Another approach would be to use Pascal II which can use the extended 80-column card fully. I have never seen a "graphics card" but some years ago American magazines contained adverts for a device which puts the graphics page at the top of memory rather than at \$2000.

Max Parrott

Stolen

OUR offices were burgled during the weekend of June 8/9 and among the items taken was an Apple Macintosh.

I wonder if you could bring the serial number of the computer to the attention of your readers in case any of them chance upon it.

It is a standard 128k Mac, serial number FG2110GM001.

We are offering a substantial reward for information which leads to the return of the Mac and the conviction of the thieves. — **Duncan Scot, Popular Computing Weekly, Little Newport Street, London WC2H 7PP. Tel: 01-437 4343.**



August 1982

Games review (Bandits, Suicide, Swashbuckler, Fly Wars) - Instruction file editor - Teach yourself Morse, Part I - VisiCalc section - Pstext II review - Asynchronous data transfer, Part II - Omnis review - A melody from your micro - Summary of 10 utilities - Make your own user port, Part II - Mah Jong - Number sorting - Elements of the Apple, Part V - Guidelines for buying a school Apple - Educational programs reviewed - PLUS four pages of Compucopia and two Appletips.

February 1983

Think Tank - Interactive editor-assembler, Part III - Development of Scrabble on the Apple - VisiCalc's storage command DIF - Games reviews (Escape from Rungistan, County Fair, Snake Byte, Snack Attack) - Software reviews (Structured Basic, GraForth, Visischedular and Lisa and the Ile - Pascal Pointers - Network analysis - Handling interrupts - Makeweight grading system - Date-stamping DOS - Educational game (listing) - Formatted Applesoft. PLUS four pages of Compucopia and seven Appletips.

September 1983

Games reviews (Evolution, Wayout, Aztec, Crisis Mountain) - First impressions of Lisa - Think Tank - Reviews (Apple Interactive Data Analysis, File-Fax, Storyboard) - Replicating with Visicalc - Printers Daisywheel v. dot matrix, maintenance contracts, stationery, Pipeline printer buffer and Fingerprint reviewed, new products, printer jargon, A-Z guide to printers, plotters and intelligent interfaces) - Apples and youth training - PLUS three and a half pages of Compucopia and 11 Appletips.

April 1984

Reviews (TK1 Solver, Rapid Reader, Homework, Pen-Pal, Cache 16/64 printer cards) - Macintosh software flashes - Build your own graphics package part III (Shape Tables) - Developing ProDOS programs - Pascal Tutorial Part IV - 16-page guide to Apple extras - from Typist to Apple User - Games reviews (Wizardry III, The War of the Samurai, The Spy Strikes Back) - Encoding routine - Book reviews (Visicalc for the Apple II Plus, The Elementary Apple) - PLUS News, New Products, and Letters.

November 1984

Apple in a primary school - Games (Flight Simulator II, Drol, BC's Quest for Tires) - Graphics Part IX (including review of Cat Graphics) - Pascal Tutorial: Introduction to Pascal Operating System - Macintosh Basics & Macintosh (MacBasic, MBasic & Instant Pascal - Two Macintosh books reviewed - Communications Part III: Software - Loading DOS Toolkit assembler onto language card - Software reviews (Digisolve's Pixel Paint, Hilderbay's Payroll) PLUS Letters, News and New Products.

February 1985

Steve Wozniak talks about Apple II developments - Quicksort algorithm in Forth and Basic - Games (Deadline, Witness, Planetfall, Enchanter, Scorerer, Expedition Amazon) - Graphics DIY part XI - Targeting with spreadsheet - Apple to Apple file transfer - Miners' strike resolved by computer? - Chemical formulae on Lisa - Two Macintosh books reviewed - World of the 6809 Part III - Software reviews (Sales Edge and Management Edge) - Application: book publishing - Split screen techniques - PLUS News, new products and letters.

October 1982

Games reviews Knight of Diamonds (the second wizardry scenario) and Pig Pen - Think Tank (with listings) - Med-res graphics, Part II (filling in shapes) - Lisa assembler language review - Magic of VisiCalc - VisiCalc Business Forecasting Model review - Cross reference listing program - Apple-vox speech synthesiser review - Morse Code, Part III - Computerised flash card for schools - French Verb program review. PLUS four pages of Compucopia and seven Appletips.

March 1983

Darts game listing - Think Tank - Beginner's look at System Master - Games reviews (Blade of Blackpoole), Banner Magic, Free Fall, Computer Scrabble) - Lower case displays in Basic - Buying a financial spreadsheet - Reviews of Multiplan; Apple-writer III; Geometry and Measurement, Drill and Practice; CLIP - News about Lisa and the Ile - Applesoft error handling, Interactive editor-assembler, Part IV - Apple on a pig farm - Fickle Finger proofing, Part I. PLUS four pages of Compucopia and four Appletips.

October 1983

Games reviews (Ultima II, Pot O'Gold Plus, Sherwood Forest, Juggler) - Think Tank - In-Circuit Emulation Part One - Lisa (emergency planning with the N.W. Health Authority, developing Busifile) - reviews (Basiccode 2, Metacraft's Forth) - Graphics (Digisolve Vector Graphics board and Apple Business Graphics) - Visicalc v. Beebcalc - Training (DIY course selection, what is training, computer-based training) - Package Deal game listing - improving life for the disabled. PLUS Compucopia and Appletips.

May 1984

Special report on the Apple IIc micro - Preview of Appletips - Lisa mapping - Using DOS within Basic - Who should buy Multiplan? - Grandad gets his Apple - Graphics Package Part IV - Games reviews (The Missing Ring, Color Me, Black Death) - Game Listing: Apple Raid - Pascal Tutorial Part V - Macintosh: the Sand Project - Semi-standard letters with Appletips' WPL - Reviews (PaperGraphics, Transitions, Mem/DOS, Ultraterm), PLUS News, New Products, Letters and Appletips.

December 1984

Games (Spare Change and Gumball) - Desert Island Discs with Pam Fisher - Pascal Tutorial (final look at the Operating System) - Spreadsheet (VisiCalc command table) - Lisa 7/7 Software - Macintosh games (Pensate and Frogger) - How Macintosh helps an actress - The world of the 6809 Pt. I: the Rehaflex board - Escher game listing - Graphics DIY Part X (including review of Sweet-P plotter) - Hilderbay SSP software review - Communications - PLUS News, New products and letters.

March 1985

Circle drawing algorithms - Super Pilot System Log - Summarising data with VisiCalc - Competitive estimating with Multiplan - Graphics DIY part XII - Ampersand editing - Macintosh (MacTerminal, Mouse Stampede, optical mouse, plus Mac book) - Reviews (Merl modem, Intec hard drive, Vision 128/256 card, the Editor, plus three educational packages) - Fun and Games (Xyphus, Fighter Command, Picture Writer) - PLUS News, New products, letters and Appletips.

April 1983

Games reviews (Type Attack, Microwave, Tubeway) - Word Processing (Supertext, Executive Secretary, Wordstar, Word Handler) - economics of using electronic worksheets - Fishing (game listing) - Apples in the pet foods and film slides industries - Anatomy of the Ile - Beginner's programming - Reviews (Omnis, Strobo 100 Plotter, Hilderbay Bookkeeper, Turnkey CP/M) - Programming for the classroom - Fickle Finger Proofing Part II. PLUS four pages of Compucopia and six Appletips.

November 1983

Think Tank (Pascal Blockwrite/read, fast data-logging, input validity checking, date verifying) - Games reviews (The Alien, New World, Crime Wave) - Neat Pascal listings - Interactive Video at American Express - Reviews (CP/M Card, Disc-o-doc, The Graphics Magician) - In-circuit Emulation Part 2 (Rovino ICE II card) - Graphics (Super resolution, hi-res text strings) - Graphs from Visicalc - Lisa's Cullinet mainframe link - Logo Part I - schools' software library. PLUS Compucopia, News, Letters and Appletips.

June 1984

The Steve Jobs interview - Pascal Tutorial Part VI - Arbitrary byte patterns in memory - Graphics package Part V - Spreadsheet: Visicalc's @LOOKUP function - Compile and solve crosswords with Appletips' WPL - British games reviewed (Derby, Soccer Manager, Election, Necromancer, Mekhula) - Modems, micros and bulletin boards - Program for marking schoolwork - Reviews (Pixy plotter, TG Track Ball, C/WP colour modulator) - Logo round-up. PLUS News, New Products and Letters.

January 1985

John Sculley's View of 1985 - Games (Geffling Adventure, Story Maker, Stellar 7) - Application: Apples down on the Farm - Cloze Technique (Plus review of Clozemaster) - World of the 6809 Part II: Flex Operating System - Apple II v. IIT 2020 - Reviews (Ormbeta Compact Accounting System, CGL Half-Height Drive) - Apple IIe and IIc compatibility - Handling Interrupts and large arrays in Pascal - Reporter's view of Macintosh - PLUS News, New Products, Appletips and Letters.

April 1985

Apples in the dental surgery - Adding graphics commands to Applesoft - Using the VBLANK signal - Getting to grips with software - Reviews (Speed-Demon card, PFS File/Report for Macintosh, W-P-LAB) - Weather forecasting with Mac - Weather Filer's D command - Fun and Games (La Triviata, Design Your Own Home: Architecture, Interiors, Landscape) - Books (Appletips, VisiCalc, Machine level programming) - Index to Windfall Vols. 1 and 2. PLUS News, New products, Letters and Appletips.

May 1983

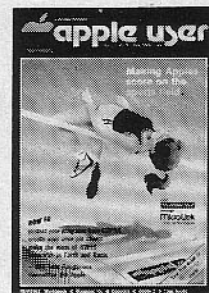
Think Tank (Visicalc Magic, Appledarts sound, hi-res routines) - Games reviews (Spy's Demise, Teleport, Beer Run, Prism, Bug Attack) - Moans about manuals - To copy or not to copy - The outdoor Apple - Reviews (Wildword, Apple Circuit, Personal Data Analysis) - Date conversion - Understand the Epson Part I - Visicalc Review of Vergecourt 128K RAMcard and Cdex Visicalc training course - Graphics (generating bar indicators with listing) - Standing Wave Plotter. PLUS Five pages of Compucopia and seven Apple tips.

December 1983

Think Tank (memory dump in Forth; shape filling) - Games Reviews (Dark Crystal; Dawn Patrol; Minit Man; Flip Out; Snooper Troops 1 and 2; Dragon's Keep; Troll's Tale) - Reviews (Word Juggler; Koala Pad Touch Table; Wildcard Reviews) - How to choose software for your business - Talking to dolphins - Write your own adventure games - Estate Agency with a network of Apples - Lander Game Listing - Visicalc cash flow projections - Drawing with Logo. PLUS News, Letters and Appletips, Compucopia.

July 1984

Capitol CAD package - Automating CP/M with Pseudo disc drive - Wordstar on Epson printer - Relational Databases - Games (Lode Runner, Coveted Mirror, Crypt of Medea, Queen of Hearts, Quiz Listing) - Screensplicer - Ile super-res graphics - Pascal tutorial part VII (defining procedures) - Volume control for Apple II - Appletips review - Lisa helping visually handicapped - Calculating mortgage repayments with Visicalc - PLUS News (including report on Apple '84), New Products and Letters.



June 1983

Think Tank - Games reviews (Pie Man, Asteroid Field, Star Thief, Cyclotron, Star Blaster, Warp Destroyer) - Security with Data Encryption - Product reviews (Routine Machine, List Handler, Apple III CP/M Softcard, Savy, Apple Project Manager and Micronet) - Apple '83 preview - Screen editing for beginners - Understanding the Epson Part II - Book review (Create Word Puzzles with Your Micro) - More Apple Plo facilities. PLUS five pages of Compucopia and eight Apple tips.

January 1984

Hi-res text generator - Game listing (Patience) - Games reviews (Apple Cider Spider; Theesus and the Minotaur; Thunderbombs; Buzzard Bait) - Lisa Workshop - Logo in the US - Pascal Tutorial Part I - Pascal PEEKing and POKeIng - Reviews (Bit Stik version 2; CIA utility; Nano 6502 Assembler; Aviette FDD 820 disc drive; KGB-40 printer; AK-GC joysticks; Praxis 35 typewriter/printer; Visicalc Advanced) - Appletips' word counting. PLUS News, New Products, Letters and Appletips.

August 1984

Communications - Apple in a haulage company - Book Review (Apple Basic Data File Programming) - Reviews (Scribe 3D CAD package, Sage CP/M Database, Codewriter IIe, Ramdrive IIe, ShortCuts) - Games (Plasmania, Bouncing Kamungas, Pinball Construction Set) - Graphics package part VI (text handling by machine code) - Lisa organising meetings - Pascal Assembler - Speech input via Voice Input Module - Multiplan helps with cricket scores PLUS News, New Products, Letters and Appletips.



May 1985

Sports Day runs smoothly with Apples - Graphics DIY Part XIII (pie charts) - Reviews (The Workbook, Macputer IIc, Copytext, Omnis 2 on Macintosh, seven Logo books) - The RWTS explained and demonstrated with a disc verify routine - protecting programs from Copya - Pascal (directory access from within programs) - Bin-search in Forth and Basic - Reaction Timer - Apples in Hungary - Fun & Games (Smart Shopper, Plantin' Pal, Micro Cookbook) - PLUS News, New products, Letters and Appletips.

June 1985

Apples keep track of music companies and Macintosh designs record sleeves - Fun and Games (Music Construction Set, Song Writer, Music Readiness) - Pascal Tutorial: start of a new series looks at records - Reviews (Tick-Tack translation package for Apple II+/Ile, Musicworks for Macintosh) - Graphics (three books reviewed) - Mughrah: light dependent resistors making sounds - Ampersound: routines for making music and sounds from Basic - PLUS all the latest News, New Products and Readers' Letters.

apple user ORDER FORM

Valid to September 30

Air mail rates on application
All prices include postage, packing and VAT

Please enter number
required in box £

November 1982

A beginner's guide to PEEKs and POKEs. Part I - Games review (Galactic Wars, Night Mission Pinball, Raster Blaster, David's Midnight Magic and three Quick Spins) - Think Tank (with listings) - Three 80 column cards evaluated - Visicalc: Brush up your algebra - Bit Stik graphic system reviewed - Pitfalls in producing educational software - Treasure Islands educational game reviewed - Med-res graphics, Part III (Ampersand routine). PLUS four pages of Compucopia and six Appletips.

July 1983

Apple '83 review - Think Tank - Games reviews (Zork I, II and III, Hitch-hiker's Guide to the Galaxy, Wavy Navy, Shuffleboard) - Using a printer with DOS - Reviews (Micro Planner and The Spreadsheet) - Visicalc potpourri - Beginners' PEEKs, POKEs and CALLS - Creating a turnkey system - Atomic research Apples - File organisation methods - Insurance broking with an Apple - Pilot Animation - Tip for using both sides of a disc. PLUS five pages of Compucopia and seven Appletips.

February 1984

Macintosh Revealed - Apple in the clothing industry - Book Review - Reviews (ProDOS, Apple's new operating system; Bank Street Writer; Keystar for Wordstar; Word Weaver III) - Logo: manipulating human language - Lisa Workshop Part II - Build your own graphics package - Games (The Quest, Story Machine, Repton, Sammy Lightfoot) - Date Manipulation - Voice Darts - Pascal Tutorial Part II - Pascal Animation - Visicalc cashflow models. PLUS News, New Products, Letters and Appletips.

September 1984

How an Apple helps police hero - Updating Apple graphics and arcade design - Serial data transfer - Games (Early Games Music, Learning with Leeper, Fuzzywomp, Halloween) - and DIY Graphics Part VII (including review of Doublestuff) - Macintosh (journalist's view of MacWrite, Software Development, Transylvania, Linking Lisa to ICL mainframe) - Pascal Tutorial - Pilot Interpreter - Spreadsheet (including bug in VAV) - Reviews of Graphpak and Format-80 Enhanced - News, New Products and Letters.

January 1983

Think Tank - Book reviews (Apple Graphics and Arcade Game Design) - Games reviews (Wizard and Princess, Transylvania) - Six-page guide to memory storage (guide to disc drives, new bubble memory, 128k RAM cards, disc back-up, mini-Winchester drives, new Apple drives) - Walt Disney's TRON - Graphmag review - Installing Wordstar - Business cash flow with Visicalc - Pilot review - Interactive editor-assembler, Part II. PLUS four pages of Compucopia and eight Appletips.

August 1983

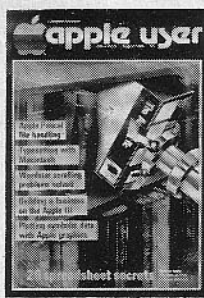
Think Tank - Reviews (The Accelerator Board - tripling the speed of an Apple II; Micro-planner Part II; The Ramview 80 and Vision 80 80 column cards for the IIe; SuperPilot - does it set a CAL standard?) - Games reviews (Kabul Spy, Super Taxman 2, Succession, Jaw breaker, Spectre) - III or IIe? the Apple III's place in the market - Use indices for What If? analysis with Visicalc - Basic editing for beginners - Pascal Disc Directory - PLUS five pages of Compucopia and six Appletips.

March 1984

Games listings (Noughts and Crosses; Twenty Questions) - Pop music Apple - Reviews (Studentdata, Sidevise) - Games reviews (Facemaker, Police Artist, Microbe, Adventures in Flesh, Dungeon!) - Pascal Tutorial Part III - Lisa (Launch of Lisa 2 series; SunAccount ledger system) - Graphics pages in memory, plus two histogram routines - Indexing Forth discs - Critical Path Analyses with Visicalc. PLUS News (More after Macintosh; multiplying mice; Lisa wins Rita), New Products, Letters and Appletips.

October 1984

Appletips (drive-cleaning, DOS, REMs and Lists) - Apples in a clothing factory - Book reviews (games programming, Pilot, Apple IIc) - Games (Gruds in Space, Cherryspin, Aquatron) - Graphics Part VIII (including review of Spectrogram colour card) - Mac software reviews (MacForth, Click Art and Mac the Knife) - Simulation of radioactive decay and Einstein solids - Pascal tutorial - Flashcalc - Reviews of Blackboard printer card and CWP drive - PLUS Letters, News and New Products.



July 1985

Apples at the heart of Papworth Hospital - Fun & Games (Secret of Arendarvon Castle, Antagonists, Fahrenheit 451, Rendezvous with Rama, Amazon, Shadowkeep, Adventure Writer) - Pascal Tutorial: using files of records - Binary file load utility - Using extended 80 column card memory - Macintosh (Flowcharting, Preview of Guide) - Book reviews (Business Basic, Epson printers) - Reviews (FingerPrint and Printinterrupt) - Graphics DIY Part XIV - DOS patches - PLUS News, New Products, Letters and Appletips.

August 1985

Spreadsheet secrets shared - Apple IIIs provide power behind computer bureau - Graphics DIY Part XV - Wordstar scrolling problems solved - Descartes data processing program generator - Fun & Games (Winnie the Pooh, Mickey's Space Adventure, Print Shop, Hitch-hiker's Guide to the Galaxy) - Mac at the centre of a publishing revolution - Pascal Tutorial: random access files - Review of Micro Planner for Macintosh - Restore to any Data line - PLUS News, New Products, Letters and Appletips.

Subscriptions

UK & EIRE £12 sterling only 1001
OVERSEAS - Surface mail £20 1004
- Air mail £40 1005

TOTAL _____

Windfall Back Issues

£1.25 UK
£1.50 Overseas

1981	1982	1983
JULY 1007	JAN 1013	JAN 1025
AUG 1008	FEB 1014	FEB 1026
SEPT 1009	MAR 1015	MAR 1027
OCT 1010	APRIL 1016	APRIL 1028
NOV 1011	MAY 1017	MAY 1029
DEC 1012	JUNE 1018	JUNE 1030
	JULY 1019	JULY 1031
	AUG 1020	AUG 1032
	SEPT 1021	SEPT 1033
	OCT 1022	OCT 1034
	NOV 1023	NOV 1035
	DEC 1024	DEC 1036

TOTAL _____

Apple User Back issues

£1.25 UK
£1.50 Overseas

1984	1985
JAN 1037	JAN 1049
FEB 1038	FEB 1050
MAR 1039	MAR 1051
APRIL 1040	APR 1052
MAY 1041	MAY 1053
JUNE 1042	JUNE 1054
JULY 1043	JULY 1055
AUG 1044	AUG 1056
SEPT 1045	
OCT 1046	
NOV 1047	
DEC 1048	

TOTAL _____

Binders

£3.95 UK
£5.00 Overseas

1067

TOTAL _____

Payment: please indicate method (✓)

TOTAL _____

Access/Mastercard/Eurocard

No. _____

Barclaycard/Visa

No. _____

Cheque/PO made payable to Database Publications Ltd.

Name _____ Signed _____

Address _____

Send to: **Apple User, FREEPOST, Europa House,
68 Chester Road, Hazel Grove, Stockport SK7 5NY.**
(No stamp needed if posted in UK) Please allow 28 days for delivery

Orders by phone: 061-480 0171
- by Prestel: Key *89. Mailbox No. 614568383
- by Telecom Gold: Mailbox No. 79:MAG001

Don't forget to include your credit
card number and full address.

AU9