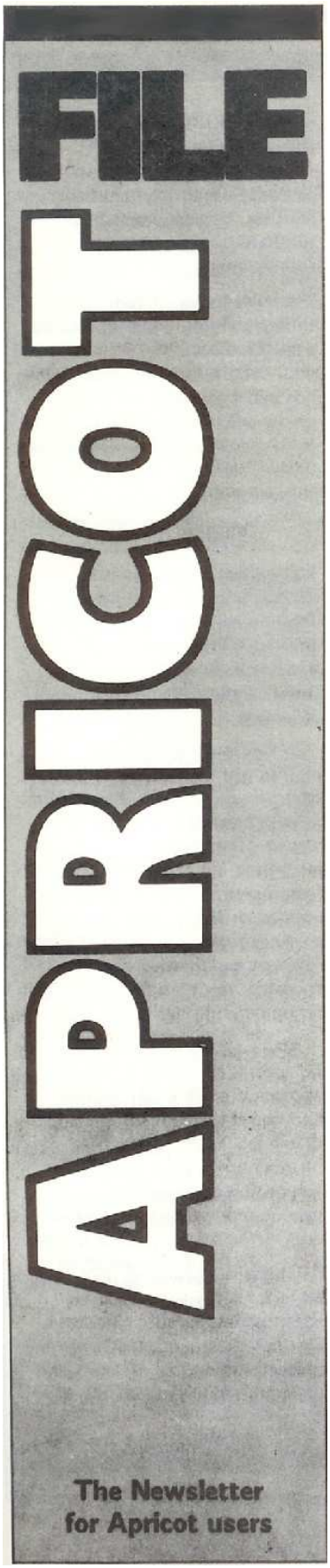
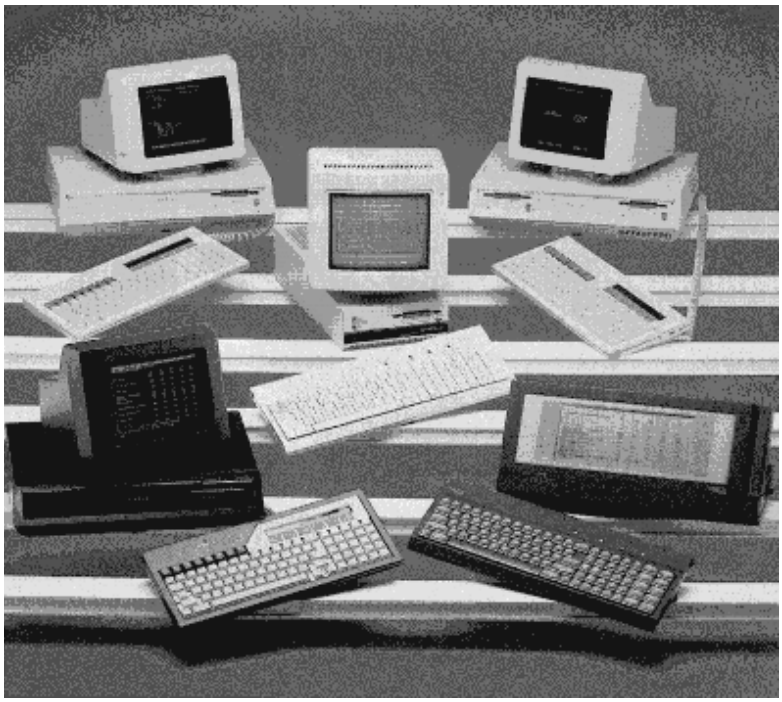


Apricot File 2_8



Scans courtesy of www.actapricot.org

Copyrighted content republished with thanks and permission from:

> Dennis Jarrett of TP Group
> Stuart Bunt
> Eddie Bromhead
> Mark Whitehorn

Many additional thanks to Eddie Bromhead for scanning all of Release 3, 4 and extra inserts.

Release 2.8

FILE APRICOT

The Newsletter for Apricot users

May 1987

Forty days revisited

Fixes for the BIOS 2.7 date bug from Ansible — and Apricot

Our story about the 40-days bug in BIOS 2.6 and 2.7 that featured on the front page of APRICOT FILE Release 2.7 apparently put the cat among the pigeons in Edgbaston.

As soon as Apricot plc saw the April issue, we starting receiving phone calls asking for more details of the fix. And commendably enough, a revised BIOS 2.7 appeared within a fortnight.

The official fix is called NEW_BIOS2.7 and it's dated 22 April 1987. On floppy systems, you insert your boot disk in drive A and the fixing disk in B — then press the Reset button, and your boot disk will be updated automatically. The BIOS in hard-disk Apricots will be updated similarly when you Reset with the new BIOS on a disk in B.

The revised BIOS has been distributed by Apricot at no charge to its dealers, who may or may not pass it on to users on request. Apricot is not recommending any upgrade price, which in our view is just as well. ("Any charges made are up to the dealer", we were told.)

If your dealer can't or won't help, and if you really need to stick with BIOS 2.7 to run certain applications, we can supply the patched version. Send a blank disk and a fiver to **Roger Gann** at APRICOT FILE, **PO Box 509, London N1 1YL.**

Or you might try the Unofficial Fix, which given Apricot's track record might be the preferable alternative. As promised, **David Langford** of **Ansible Information** has come up with the goods in the form of a tiny COM file called LENT.COM; you just load it as part of your AUTOEXEC routine, and it hangs around in memory — correcting date reporting by MS-DOS and the BIOS, and putting the right date-stamp on files.

It checks to make sure it's inside a PC or Xi with BIOS 2.6 or 2.7 before loading, and it can be removed from memory if required ("leaving no unsightly stains").

The price is **£11.50**. That includes VAT and despatch, but not a disk — send your money and formatted disk to **Ansible Information** at **94 London Road, Reading, RG1 5AU.**

Turbocharged Apricot

There's another unofficial fix you might consider for your Apricot. Those of you who thought the Apricot PC/Xi was inextricably tied to the rather pedestrian clock speed of 4.77MHz, think again! The go-faster Apricot has been discovered alive and well, and living in Streatham.

So — who, how, why and how much? The 'who' is **Martin Filmer**, a former *Tour de France* veteran turned Apricot boffin. He runs a modest

▶▶▶ CONTINUED ON PAGE 2

In this issue

Review: Cirtech Cachebox 4
Contributors in this issue 6
News of the Comms World 7
The Workshop: Inside MS-DOS — redirection and piping . . 8
Review: Ansible's WordPerfect utilities 14
Report from Bandage and Plaster 16
Soapbox Response Analysis 21
Review: Signwriter — attention-grabber for Epsons 24
The Workshop: All About Apricot RS-232 — part two . . . 26
Diary Dates 29
Doodles by David: Tips and Tricks 30

▶▶▶ CONTINUED FROM PAGE 1

repair and maintenance outfit in sunny South London and specialises in Apricots; an example of that is his ability to 'chip-up' the motherboard to 896KB (and beyond!) thus freeing a valuable expansion slot for other uses. The office Xi now boasts a 960KB motherboard, thanks to Martin's handiwork.

How the turbocharging is done isn't so easy to explain. Martin began with an 8MHz NEC V30 replacement for the 8086 CPU. Now, many people have dabbled with swapping their 8086 for the 5MHz version of the V30; and they have reported modest gains (and screeching loudspeakers). The 8MHz version is a somewhat different matter: to use it, you have to change a total of nine support chips on the motherboard and upgrade the CPU crystal clock to 24MHz.

Another desirable move involves the replacing of the DRAM memory chips from the standard 150 nanosecond spec to the more rapid 120ns variety. This is trickier than it sounds, because the chips to be swapped are not socketed. Martin removes the old chips and thoughtfully installs sockets before putting in the faster chips.

One of the more striking changes involves the monitor. Sadly, the 15MHz bandwidth of the original monochrome monitor precludes its use with the faster system unit — and it isn't possible to convert it, either. The screen has to be replaced with a 24MHz monitor of comparable resolution. And those aren't too thick on the ground. After some hunting around, Martin found a suitable monitor made by Manitron. It offers a 15in screen, which is highly desirable: but it is a little on the large side — to put it mildly. It would be no exaggeration to say that you could put the system unit on top of the monitor ... That aside, the larger screen is a joy to behold, really doing justice to the Apricot's superb hi-res graphics.

As for why you should soup up an Apricot — well, with the Filmer mods the thing now goes like a rocket!

There are various ways of measuring computer performance, but the commonest is probably Peter Norton's SI System Information program which gives you an index of performance relative to an IBM PC running at 4.77 MHz. So the IBM PC would return a value of 1.0. Our Xen-i, a 10MHz 80286 machine, returned 10.3 (phew!); the Olivetti M28, a machine with an 8MHz 80286, gave 7.3.

By contrast, a plain vanilla-flavour Apricot PC does slightly better than the IBM PC, at 1.2. That's because it has a true 16-bit processor (the Intel 8086) rather than the psuedo 16-bit (or 8/16 bit) 8088 in the IBM PC.

But the Filmerised 8MHz V30-based Apricot PC returns a whopping 3.5, making it some three and a half times faster than the IBM PC and nearly three times as fast as the standard Apricot.

As well as the increase in raw processing speed, other aspects of performance are improved as well: disk accesses are quicker and the screen handling is decidedly snappier. Drawbacks? Well, none so far. Martin reports no known faults: "everything works as expected — just a whole lot faster!"

The conversion is not particularly cheap, the main cost item being the monitor. You can expect to pay around **£700**, for the monitor alone is £400 or so (but Martin says he's seeking out cheaper alternatives). The undeterred should seek out Martin Filmer on **01-761 0425** for further details on price and availability.

Apricot cuts dealers

In line with its declared policy of becoming a 'systems' manufacturer, Apricot has appointed a couple more distributors and told some 300 of its smaller dealers that they won't be able to buy direct from the company unless they apply to be upgraded. These are the so-called 'Apricot Authorised Dealers', once the lowest rung on Apricot's three-step ladder of dealer status. The AADs can try for the glamour of joining the 250 or so Apricot Business Centres — who get the best margins and the best support, but in return have to guarantee a level of business and make certain commitments about their marketing and technical support capability. Otherwise the AADs will have to buy from Apricot's master distributors, two more of who were named recently — **Northamber** and **Data Base Limited** join DDL and MSD, who were appointed late last year.

The existing Apricot Business Centres seem to applaud the move, the general view being that it will cut out the cowboys. The ABCs are of course the established larger dealers who Apricot sees (quite legitimately) as crucial to its new image; the AADs are mostly the smaller guys, and an Apricot person said "we are a systems company — we have got to give support to our major dealers handling multi-user systems". Apricot systems division MD **Ed Sherman** put it in a nutshell: "we have focussed on becoming a systems company, and people in the marketplace should understand this ... Authorised Dealers do a level of business which does not justify our commitment".

Re-inventing the Apricot PC

What would you think of a slim and stylish desktop personal computer powered by an **Intel 8086**, featuring a **full 16-bit bus** and sporting **twin 720KB 3.5in disk drives**? A machine that offers better than IBM-standard monochrome screen resolution, parallel and serial ports and three internal expansion slots to boot?

Nothing new here: the 1983-vintage Apricot PC had it all. If I told you that it came equipped with 640K of RAM on the motherboard and the aforesaid processor ran at 8MHz, would you still think it was an Apricot PC? But that's nothing startling either — a few chips and a new crystal would see the Apricot PC up to that level of standard equipment quite handily.

What if we told you that it was announced on 2 April in Greenock, and that IBM was the proud parent? And that the price is **£1,307** including monitor?

According to some minds at IBM, Roger Foster got out of the 'commodity' PC market just in time for their liking. Apricot couldn't stand the heat in that particular kitchen and moved up-market with the sleek 80286 and 80386 Xen machines.

With the need to amortise the astronomic investment in largely automated plant at Greenock (currently churning out Personal System/2 Model 30s at a high rate of speed), the last thing IBM wanted was a competitor with a virtually identical spec that had been introduced four years previously ...

The drives on the new **IBM PS/2 Model 30** won't read the old Apricot-format 3.5in disks despite the similarity of media. Instead of the memory map built into the old Apricot that allowed MS-DOS to use 896KB of RAM, the new IBMs are still restricted to 640K like their PC/XT brethren. The monitor specifications, while greatly improved over previous IBM practice, still live with a 640x400 resolution — and they still don't match the 800x400 spec of the old ACT Sirius 1 which was subsequently incorporated into the now defunct PC/Xi range. *This* is progress?

At the same time, IBM launched DOS 3.3. This is destined not to thrill the world either, offering as it does only a handful of new commands and the ability to access Winchester disks of over 32 megabytes. It does, however, run on Apricot's IBM-compatibles; one wonders if Apricot's assurances regarding support of its orphaned machines will include making this modest improvement available to the legions of PC/Xi owners?

Nowhere in sight of the hoard of computer journalists jetted to Scotland was the oft-rumoured operating system to exploit the multitasking abilities of the 80286 and 80386 chips (as found in Xen-i models). But scheduled for release by Microsoft "early in 1988" will be OS2, which will allow single-user multitasking on all 80286 and 386 machines (Xens included, presumably). In addition this new operating system will allow direct addressing of up to 16 megabytes.

This may be worth waiting for: it promises compatibility with all old-style MS-DOS applications when run as foreground tasks, allowing new OS2-developed applications to run simultaneously in the background. IBM assembled impressive lists of software vendors already committed to supplying subtly-tweaked programs that fit the as-yet-unseen operating system.

Further in the future, IBM has indicated that an extended version of OS2 will include some very PICK-like features including built-in communications and database facilities (as afiçionados will know, PICK is a largely unsung competitor to Unix in high-performance multi-user operating systems). Presumably, the new 'MicroChannel' (tm) 32/16 bit architecture in the new IBMs won't cause these enhancements to be invalid on Apricots. Although there were assurances that these new developments would in fact run on any 286 or 386 machine, the official word from Big Blue was "We cannot speculate as regards to any other manufacturer's products".

Tandy clear-out

Last month **Tandy** decided to clear inventory of Apricot goodies left over from its abortive joint retailing enterprise in **AT Computerworld**. There are four Computerworlds left, now operating exclusively under the Tandy umbrella, and they all continue to stock current Apricot products; the present bargain-basement sale covers "old and slow-moving" stock. Which means some bargains — Roger saw serial **mice** going for **£6** and bought an apparently duff **12in monitor** for **£10** (said duffness constituted one detached wire, and the thing now works beautifully).

But get your skates on: Roger arrived a few hours after the Tandy sale started, to discover that most of the good stuff had already gone ...

... US cleared out

In the States, Apricot's abortive US venture included lots of own-branding of popular software packages in Apricot format. It's all been sold now — to an English firm with an eye to the main chance. As recipients of our last DEALS FOR READERS mailing may deduce from the advertising insert, **Sierra Systems** scooped the lot — Sierra's **Geoff Wilkinson** heard about the stocks, had a quick word with his bank manager, flew to California, and bought up software with a notional resale value of around $2 million. It's now overflowing from a warehouse at Heathrow ... call Geoff if there's anything you want, for he has some decent discounts on big-name products (the number is **03727 22890**), but we'll be taking some of the more useful-looking items from his stock-list for future issues of the DEALS FOR READERS mailer. ◆

Contact points

APRICOT FILE **is a publication of the TP Group — which until we get the name changed is the trading style of Permapoint Limited, registered in England no 1929195.**

Postal address

TP GROUP
PO Box 509
London N1 1YL
England

Personal calls and deliveries

TP GROUP
Unit 4
47-51 Wharfdale Road
London N1 9SE
England

Phone, E-mail, Telex

Telephone	**01-833 3501**
Telecom Gold	**81:JET767**
[from overseas 10081:JET767]	
Mercury Link	**19011170**
The Source	**BDE566**
Telex	**265871 MONREF G**
	quoting reference 81:JET767

The People

Editor	**Dennis Jarrett**
Production	**Fiona Carey**
Writer	**Garret Keogh**
Subscriptions	**Terrance Duggan**
Technical	**Roger Gann**

ISSN 0268-6066

Review

A scroll down memory lane

Cirtech's Cachebox reviewed

By Ronald Ellison

What's the last computer peripheral you saw that got you excited? Laser printers, perhaps? Maybe page scanners? Well, it's easy to forget older technology that may still be quite helpful today. This brings us to the subject of memory, and specifically to printer buffers.

What's a print buffer?

Although they don't ever seem to have been terribly fashionable with PCs, print buffers have been friends to computer users for years, starting with the very largest computers and eventually making the evolutionary links downward through the minis to the micro.

When computer memory was very expensive, when having 64K bytes was something to boast about, time spent on the computer system was very costly. So various schemes were devised to offload the printing time, so that the computer (which was relatively fast) did not have to wait for the printer (which was relatively slow) to finish printing before it could free the chunk of memory to continue with other tasks.

Comparatively cheaper types of memory, such as reel-to-reel tape, were used to take in the data to be printed; such devices would pass it on as quickly as the printer could print. At one time even spools of wire were used for temporary magnetic storage, and thus print buffers are sometimes still called print *spoolers*.

The use of a print buffer means that the computer can squirt a lot of material to be printed at the maximum possible speed down the cable and forget it; the printer then chugs away at its own best rate, printing whatever is in the buffer rather than taking information to be output directly from the computer itself.

When the buffer is full, the computer obviously has to stop sending characters until there's room for them; and at that point it might slow down again to the printer's speed — since the printer can't take information to be printed from its own memory any faster than it can handle print data from the computer.

But in any case you should be able to get use of the computer again much sooner than otherwise. And if the buffer is big enough, it's possible that everything you want to print can be stuffed into it almost immediately — and you can back to the screen and keyboard.

Most printers these days have a small print buffer in the form of a few kilobytes of RAM actually inside the printer — typically a buffer like that will hold at least a thousand characters. And more expensive units may have a buffer capable of holding up to 6,000 or more characters.

You can however help things along dramatically by buying a separate print buffer. Essentially these are little black boxes (or rather, not so little in some cases) that fit somewhere between computer and printer. The box contains a small microprocessor and some pretty basic control software ... along with a chunk of memory. Generally that's between 8K bytes and 512KB, with prices rising commensurately from around £100 to something over £300.

These units won't affect the operation of the computer in any way — except to minimise the time the computer spends tied up on printing. And a 64KB buffer is large enough to swallow a good deal of material due for printing in one gulp, so the effect can be virtually instantaneous.

Print buffer units need some power to operate. They might be able to take enough from the computer-to-printer connection, which generally includes a thin trickle of electricity; but most buffers need a separate connection to the mains.

Some buffers can do more than stack up material for the printer to print. Once there's a microprocessor with some memory in the black box, after all, a clever person could program it to do a variety of different tasks — even if the thing is dedicated to passing on information. In particular, a 'general-purpose' buffer like this will probably be able to convert between different types of interface.

It is of course possible to use part of the RAM inside your computer as a buffer. Some multi-function RAM boards include special software to let you set aside part of internal memory to act as a buffer. An assortment of commercial and public domain versions are available too; and even MS-DOS includes a PRINT.COM program which gives the basic functions of a buffer.

Like many software approaches, the DOS PRINT utility is intended only for pure text and handles only one file at a time. The black box scores here — and in several other attributes, as we'll see.

What this one does

The subject at hand is a particular print buffer made by Cirtech. 'Cachebox' is a clever title since memory set aside for special tasks such as printing is sometimes called 'cache' memory; this unit comes in a small box which looks rather smart and durable. It provides either 256K or 512K bytes of temporary RAM memory, using the same type of chips as are used in your computer. Now that's a lot of memory, but then memory is cheap today!

Depending on the type of file, and how you send it to the printer, it may save a great deal of time for the first printing; and certainly it will save time for the second and subsequent copies. That's because this box, unusually, has the handy addition of a COPY button. Even if the computer is turned

off, you can reprint as often as you need — or rather, up to 65,535 consecutive times (according to the manual; we didn't check that). This is helpful for repetitive printing, like small in-house desktop publishing runs, since you need the computer only for the original print.

The COPY reprints *everything* that's in the Cachebox. The contents accumulate each time you print through the box — unless you press its RESET button to clear the memory. So when you send one letter first, then a second and a third, all three are kept in the buffer; and if you then use the COPY button, you will start to print the first letter first, then the second letter, and then the third. This could be an advantage if you want multiple copies of more than one print job. Or it could be a real irritant if you forget to use RESET.

You can also 'print' into the buffer first, loading up several documents and then actually print them. Just turn the printer off-line while you load the buffer; then set it on-line again when you want to print and press COPY.

Also, you can press COPY more than once to 'program' it automatically to print out multiple copies — each press increments the number of copies. In this way you could send to the buffer a report with text pages and graphics pages and set Cachebox to print the entire report say a dozen times.

But the print buffer won't necessarily mean a dramatic improvement in overall throughput, and not all files are faster on the first time print. This is because many programs have been purposely designed to print slower, section by section. Some are so designed to accommodate the difference between computer speed and printer speed, while others must compose print-time effects as they process section by section. Examples of the latter are most word processors, graphics programs and, again, desktop publishing packages.

Of course, with COPY the second and subsequent printings will be as fast as the printer can go, and will be independent of the computer.

Setup

Cachebox is easy enough to set up, whichever of the three models is used. One will connect Centronics-style parallel interfaces on the computer and the printer ends: there's an RS-232 serial-to-serial equivalent: and the clever third version lets you mix 'n' match, connecting parallel or serial computer interfaces to serial or parallel printers.

Installing the parallel model is as simple as removing the plug from the printer, plugging it into the box and plugging the box's cable into the printer.

The serial version adds the requirement of setting switches to select two of the usual serial parameters: speed of transmission, and 7 or 8 data bits; the box uses two stop bits, which is a bit curious these days (one or none would be more usual for an RS-232 connection). Don't worry if these details seems a bit unfamiliar: such information should be available in the manual for any serial printer, and besides you probably select the same information for it already using the MODE command.

The mixed-mode model is the most interesting one of the three; and it may need a little more attention, especially when you use its 'translation' facility. You get all the capabilites of the 'straight' models, though; and for basic operation it requires only an additional switch selection to indicate the type of interface *from* your computer and the type of interface *to* your printer.

All interfaces are marked for serial and/or parallel, computer input side and printer output side. Plug in the cables, if needed set the switches, plug in the power, and just forget about it until you want to use its special abilities.

Besides the COPY button on the front panel, there's also RESET which clears everything in the buffer and two indicator lights: the DATA indicator tells you when there's data in the buffer memory, the COPY light shows that the buffer is copying and won't take any more data from the computer.

All models use a transformer/power supply which plugs into the mains and into the back of the box. When it's plugged into both sides, the buffer is ready — there is no ON/OFF switch.

In use

Once properly set up, operation of the Cachebox is simple; just print as you normally would. We tried text, graphics, a mixture of both ... absolutely no problems. Everything printed as expected, and everything COPYed just as if it were the first time print. Graphics copied as perfectly as text did, and did not decline even after several copies were printed — the ability to print graphics is especially helpful since most software print buffers can't handle these.

To test some of Cachebox's limits, we attempted to push the buffer box *really* hard. Various combinations of parallel and serial, 'translating' between the two, produced no difficulty. And since we had a 512KB model for review, we tested it with a very large file — 360KB of it. To avoid generating an enormous stack of paper, we let it 'print' by using a comms package to send it in ASCII form over an RS-232 link — from one computer, through the buffer, to another computer. Again no problem. The FC and COMP (file-check and compare programs, respectively) checked the received file against the original, and it passed.

Not to be stopped, 15 hours later we tried the COPY command to send another copy of the same file — which was still in the buffer memory — to the receiving computer. This time we used FC and COMP to test against the first file received; they were identical.

What's it worth?

Hopefully this review gives you some ideas as to the uses of print buffers, especially the 'smart' and COPY-equipped boxes. Of course, if you have a use for a buffer it probably will be for more

conventional applications than our tests, which were designed to show off some of Cachebox's capabilities. But our results did show Cachebox to be efficient and reliable, and the box proved itself a good performer.

Accolades in particular for the parallel-serial model. Particularly if you have multiple computers and/or multiple printers, it can be handy to be able to mix-and-match freely without regard to the type of interface of each. And the mixed model best assures for future expansion, again without concern for the specific interface.

The panel indicates the prices, but the mixed parallel/serial model has to be the best value since its greater flexibility costs little more than the simpler models. It's also cheaper than many print buffer boxes on offer from mail-order suppliers, and it has more memory and more features than most.

As to the size of memory, that depends on the kind of print jobs you heap on your printer(s) or how many separate print jobs you want to accumulate in the buffer. Generally speaking, 256KB will get you through many text applications; 512K might be better for dense graphics or print effects such as those produced in computer-aided design jobs or desktop publishing.

As a final high note, Cirtech has been making such microcomputer boards and boxes for some time now; and Cirtech's people seem to be both knowledgeable and friendly. It's good to know that behind a smart box are some nice people. ◆

Cachebox

Description:	Print buffer with COPY facility; two memory sizes, choice of serial-to-serial, parallel-to-parallel, and parallel-serial conversion.
Supplier:	Cirtech (UK) Limited
Address:	Currie Road Industrial Estate Galashiels Selkirkshire TD1 2BP
Telephone:	0896 57790
Price:	■ 256KB: parallel only £198 serial only £198 parallel and serial £238 ■ 512KB: parallel only £288 serial only £288 parallel and serial £328
Configuration:	Any Apricot
Conclusion:	Reliable, efficient, effective, quite competitively priced: a good buy for heavy print applications, and in particular the parallel-serial model could be extremely useful.

Contributors in this issue

Dennis Jarrett is Editor of the newsletter. His magazine credits include being founder-editor of 16 BIT COMPUTING, OLIVETTI USER, COMMODORE USER, OFFICE AUTOMATION REPORT, PRACTICAL COMPUTING and WHICH COMPUTER? Books include THE IBM PC BOOK (not very thick), THE SIRIUS/VICTOR 9000 BOOK (very much thicker), THE ELECTRONIC OFFICE – A MANAGEMENT GUIDE (fairly thick), three titles about Commodore home computers (one thick, one a bit thick, one pseudonymous), and THE GOOD COMPUTING GUIDE (*really* thick and very cheap too at £2.95 — or £3.00 inc P&P from DEALS FOR READERS).

Ronald Ellison is a metaphysician for hire who supports his habits of being with computer consulting, training and writing. He swam through the sea of CP/M, early MS-DOS and into the vast oceans of Unix and Xenix before he washed ashore on the steps of the TP Group. Having spent much time in the field in the States, he's now staying more around town to farm out his skills.

Mike Florey works for British Telecom as the Technical Manager for a large transmission network contract. Outside work he is closely involved with the off-stage activities of Streatham Hill Operatic Society.

Terence Green is an unemployed Apricot owner (currently looking for a job). Formerly an unlicensed truck driver in Lesotho, he started on IBM 370s and worked his way down to micros; these days he minds the kids and scrapes a living washing dishes (and writing occasionally for APRICOT FILE).

Simon Potter is a business systems consultant with a speciality in word processors, file managers and spreadsheets. He says he'd like to like computers but can't find one that likes him enough to warrant that.

Bob Smith has worked his way from big computers that did little to big computers that did a lot to little computers which did little once but now do a lot. This took some 18 years to accomplish, usually with big firms in big towns, but he now tries to do a lot for a little with an Apricot and a minicomputer for a company in a small town in the West Country.

David St. John-Wallis is a sort of psychologist who runs an export marketing consultancy and writes *enormously* thick marketing books about the developing world. He's a chastened man since he first leapt to journalistic prominence in these pages; vituperation, vilification, abuse, rejection and guffaws have nearly broken his spirit. So he now runs a sort of computer software company which doesn't do very well. He says that all contributions will be gracefully acknowledged.

News from the Comms World

Smart cards, more and cheaper Gold, personalised e-mail Telex

Remember those so-called 'Smart Cards' developed in France a couple of years ago, with a microprocessor and memory chips embedded inside something that looks like a slightly lumpy credit card?

For some time this has looked like a classic case of technology seeking an application; but the Swiss company **COMCO** has come up with what sounds like a Good Idea, putting those cards to work on the potentially messy business of international comms.

Communicating by computer with your e-mail service or your home base when you're travelling abroad can be difficult enough because of the need for lengthy dial-up and log-on procedures; and you often need a subscription to a local comms service. COMCO has made arrangements with public and private telecom groups, including British Telecom and various national PTTs, whereby its smart card can connect you automatically through a local number to the destination system. The COMCO Smart Card communicates with computers installed within the packet switching networks run by the PTTs; and so a user can access the PSS network in a foreign country without a subscription. This avoids the complications of arranging a Network User Identity, screen prompts in a foreign language and invoices from abroad.

And since it's a smart card, it carries a running balance of phone credits available — you pay for them *before* you use them — and it will warn you a month before it should be renewed.

Telecom Gold apparently feels this is a convenient and cost-effective method of accessing a mailbox from abroad for users who travel regularly, and it's one of the first services to sign up. The punters do seem to want it; an on-line survey on international access conducted by Gold in February showed that users would like a more convenient method of accessing their mailbox while abroad.

To use the system, a phone call is made to the nearest COMCO access point and the Smart Card is placed in a small reader which is connected between the computer and acoustic coupler. The card then dials the number stored in it. Once you're on-line, the Smart Card is debited with the cost of the call and International PSS charges. The only payment at the time of the call is the local telephone charge.

At present there are access points in London, Paris, Geneva and Amsterdam; more numbers will shortly be available in West Germany, Australia, Hong Kong, Canada and the USA, with additional points planned for the future.

Golden Anniversary ...

Telecom Gold was five years old on **1 April 1987** (no comment) and celebrated by announcing that the service now has **76,000** 'active' customers.

... Gold is cheaper ...

Telecom Gold has dropped its **minimum monthly bill** of £10 per mailbox; you'll now pay only for storage and services actually used. This applies to 'Club' accounts, the basic service for individual customers and small businesses.

... More messaging

And Gold is taking over the **Prestel** messaging system, which among other effects will permit messages to pass between the two services — so e-mail users on Gold should be able to message Prestel subscribers (and *vice versa*). This incidentally increases the combined head-count of subscribers to around 130,000. Eventually, Gold users may get access to the information database services found on Prestel, and Prestel may get the more advanced mail/message facilities of Gold.

If you want to know more, Telecom Gold is touring a mini-exhibition over the next few months, with demonstrations of electronic mail, Telex and some database services. The roadshow stops for two days each in **Glasgow** (Scottish Exhibition Centre, 12 May) and **Newcastle** (Civic Centre, 20 May).

E-mail Telex for you

You can send and receive Telexes via most e-mail services, but only via a common Telex number with an 'attention' answerback line directing the replies to your mailbox. But **Mercury Link 7500** (formerly **EasyLink**) can now provide a personal Telex number for its subscribers; more info from **01-847 6070.** ◆

COMMS FILE

If you're interested in using your Apricot for comms, you should be reading our latest newsletter.

COMMS FILE is a monthly in the classic TP Group mould — full of workshops, reviews, technotes and advice for the desktop computer owner who has a modem or is thinking of getting one.

Release 1.2 has just appeared; it includes a Hayes-compatibility workshop (use Hayes modems to dial and log voice calls), Prestel for the PC user (a regular slot), a continuing Simon Potter series on comms for the beginner, a look at comms from a cheap home computer — the Commodore 64, reviews of two bulletin board systems, a workshop on the MS-DOS MODE command, and Dermod Quirke's famed Up at the Sharp End column.

COMMS FILE is for anyone interested in electronic mail, Telex from your computer, bulletin boards, RS-232 connections, dial-up databases ... just about anything to do with comms.

For existing APRICOT FILE subscribers we have a special subscription offer — 15 issues for the price of 12. We don't bill you until you've had the three free copies, and if you don't like the newsletter you can cancel the invoice. Why not ask for details?

Inside MS-DOS

Redirection and Piping

By Simon Potter

So far in this occasional series we've mentioned 'redirection' on a couple of occasions. It's time to explain and explore this facility of MS-DOS a bit further.

When any MS-DOS command or batch file is run, any input that it requires will usually come from you via the keyboard. Similarly, any output will normally be displayed on the screen. This applies to many programs too, especially the utilities provided on your MS-DOS disk — if any parameters or extra instructions are needed, you'll be expected to type them in: and the results will be shown on the display.

Files can be devices too

But it is possible to 'redirect' both the input and output devices used by a command or a program — to tell MS-DOS to get any input from some other source and/or to send the output somewhere else.

It *is* possible ... unless you are using the speed-up utilities FAST (DEALS FOR READERS disk AS001), HARE (on Ansible's ACOLYTE disk, DEALS FOR READERS no AS045) or the QUIK utility on OmrC's SCANNERS disk. They all kill the redirection facility.

But unless you have one of these loaded, you can use redirection. That you do by including a special 'redirection' sequence after the command or program name, such that the syntax becomes ...

```
command symbol filename
```

The filename is one of the standard MS-DOS filenames — so it could be an actual file on disk, or it could be an MS-DOS device like PRN (printer). If it's a file, the name can include disk drive and/or path specifiers.

There are three possible redirection symbols:

- < a left-pointing 'less than' arrow means "take input from the filename following"
- > similarly, a right-pointing 'greater than' arrow means "send output to the filename (or device name) following"
- >> a pair of 'greater than' arrows means "append output to the filename following"

Here are some examples of the technique in use.

```
DIR B: > PRN
```

In this case the 'file' is an MS-DOS device name — PRN, denoting the printer. Here you're listing the files on drive B (by DIR B) — but instead of them appearing on the screen, they will be printed out. Of course, you could do the same by entering DIR and using the CTRL-P key combination. (The panel at the end lists the MS-DOS device filenames.)

```
DIR B: > DIRB
```

Again you're listing the files on B — but instead of them appearing on the screen, the command creates a new file called DIRB and puts into it the same information as you'd normally see on the screen. So when you enter TYPE DIRB, you'll see something like this:

```
 Volume ID for drive B: has not been set.
 Directory of  B:\

CG86       BAK      6506   7-10-86  12:31p
CG86       NEW      6616   7-10-86  12:34p
TEST                 338   7-10-86   5:24p
Q-GEN      TXT     12726   1-01-80   7:28a
PIPE-RED            3205  24-09-86   5:18p
DIRB                   0   7-10-86   5:41p
        5 File(s)    328704 bytes free
```

When you're redirecting output from a command like DIR which normally displays its results for you, the screen will appear to go dead for a while — until the command has finished executing, and until the output has been redirected to the destination file, at which point the system prompt will reappear inviting you to enter another command.

This particular example illustrates one wrinkle with redirected DIRectory commands — the destination file is created before the command is executed. So the DIR B: will show the new file as being in the directory ... but since it hasn't received any output yet, it will have a zero length.

As the example stands, DIRB would be created in the default directory of the default drive — the last drive and directory you specified. You could have used ...

```
DIR B: > B:DIRB
```

That way you'd get your directory of drive B actually created on the disk in B.

Be warned: if you try to redirect output to a file that already exists, you'll lose the existing contents. So be careful when selecting a name for the redirection file; or use the double right-arrow.

```
DIR B: >> A:DIRALL
```

This option works in exactly the same way, except that the output is *added* on to the end of the filename specified — which in this case is a file on drive A called DIRALL — and none of its previous contents are lost. If the file doesn't currently exist, it will be created for you.

The example here could be useful to create a master reference to the contents of your disks. Subsequently you might use that to search for a particular filename using your word processor or the FIND command (of which more later).

Input redirection

You can also use redirection to nominate a source of input other than the keyboard.

```
DEL B:*.* < YES
```

The left-arrow means *all* the input the command expects is actually read from the filename specified. Ordinarily this would be a simple ASCII-format text file that you've created, and it would contain exactly what would have been typed at the keyboard. So in this case, you'd have a (dangerous!) automatic delete-all — the file YES would comprise simply the letter 'Y', and that would answer the question 'Are you sure?' which appears after a DEL *.* command.

Note that the source file must contain *everything* the command or program would otherwise expect from the keyboard. If there's something missing, MS-DOS will do as much as it can and then lock up on you; it won't return to the system prompt and give you back keyboard control. You'll have to use the Reset button.

It can be useful to create a file called YES and another called NO (containing 'N'), to be used as required when a command or a program requires a Y or N input.

You can combine both input and output redirection. You might want to create a file called TODAY by redirecting the output for the DATE command:

```
DATE > A:TODAY
```

That creates a new file called TODAY; and provided you've set today's date correctly, the file will contain today's date in the format that the DATE command produces. But you'll need to hit Return twice after entering this — once to execute the command, once to answer 'Enter new date' prompt (which will not be displayed, because all messages from the DATE command are being redirected to the file called TODAY).

But this removes the need for you to hit Return when asked for a new date:

```
DATE > TODAY < RET
```

RET simply contains one line with a Return on it (try using EDLIN and pressing Return on the first and only line of the file). So while the 'Enter new date' message will still be produced by the DATE command, MS-DOS will go to the file RET for its answer — which will be a Return, just as if you'd pressed the Return key yourself.

So you could have something like this to log a day's work:

```
DATE > A:TODAY < A:RET
DIR B: >> A:TODAY
```

The first line creates a file called TODAY that has today's date (admittedly followed by an unnecessary 'Enter new date' line, but you could remove that with EDLIN or a word processor). Then you add the DIRectory of drive B to the same file; so you end up with a dated directory.

But in general, redirecting output is a lot more useful than redirecting input. That's simply because in many cases you won't be able to respond to prompts or warning messages from a program or command once it's been told *not* to expect input from the keyboard.

Similarly, though, there's a problem with redirected *output*; if the the command or program isn't outputting to the screen, you won't even be able to *see* the prompts or error messages.

Pipes and filters

Microsoft, which produces MS-DOS, is also responsible for the Xenix operating system. Xenix has become the most widely-used variant of Unix on smaller multi-user systems (and the IBM-compatible 'Apricots' now have Xenix as precisely that option).

Although it may not be obvious at first glance, Microsoft has progressively endowed MS-DOS with more and more of the features of Xenix. These include redirection and the practice of treating virtually everything as a file; and the Xenix connection has also given MS-DOS 'pipes' and 'filters'.

The piping feature of MS-DOS allows you to direct the output of one program or command so that it is used as input by another, effectively chaining a series of them together. Each program or command to be linked in this way is separated by the vertical bar, which on most systems appears as a broken vertical dash.

Piping is used most often with the three 'filter' commands:

- MORE displays data one screenful at a time
- SORT rearranges data
- FIND searches for specified characters

We'll see how piping works when we look at these in more detail. For now, be warned that MS-DOS stores the piped output from each program in a temporary file in the root directory of the default disk: so make sure there's some space on the disk for these files.

The MORE command

MORE displays the contents of a file (or the output from some other process) a screenful at a time. At the end of each screenful, you'll see the prompt ...

```
-- More --
```

Pressing Return produces the next screenful; pressing CTRL-C aborts the command and returns you to the MS-DOS prompt.

MORE can be used on its own, in which case you specify the input source by using redirection:

```
MORE < filename
```

This will TYPE the named file on to the screen, pausing after each screenful of information. You could also specify a redirected output, which in most cases would be the printer:

```
MORE < filename > PRN
```

Now the file will be printed with a 'more' pause at the end of what would've been a screenful of text.

MORE is primarily useful in displaying text. A big batch file, for instance, might involve a lot of on-screen explanation that you want to put before the user via a TYPE INFO.TXT command. If there's more than one screenful of text lines in INFO.TXT, they'd scroll past before the user has time to digest the information. But changing the command to ...

```
MORE INFO.TXT
```

... will guarantee that the readers miss nothing, and allows them to proceed at their own pace.

But MORE usually appears as a filter for other commands rather than as a command in its own right. A filter affects the otherwise standard output from some other MS-DOS command. For instance:

```
DIR | MORE
```

This will pause a directory listing at the end of each screenful (and so it's exactly the same effect as using DIR /P, though the end-of-screenful message is different).

The SORT command

SORT is used (would you believe) to sort information on a line-by-line basis. You can use it on quite large files, up to 63K bytes in length; you can sort in ascending or descending order, and you can specify the position along the line at which you want the sort to take place. Like MORE, the SORT command can be used on its own or as a filter operating with another MS-DOS command; and like MORE, it needs redirected input — and the output can also be redirected as well.

In its simplest form, SORT looks like this:

```
SORT < filename
```

This will sort the named file conventionally, on the basis of what's in the first position on the line, and display it on the screen. The sort works much as you'd expect, in this order:

- spaces come first
- dashes
- numbers (though only on the basis of the first numeral — so '333' will appear before '99')
- punctuation marks like colons
- letters (with case ignored)

You can sort in reverse order too, so that 'Z' comes before 'A' and '9' before '0':

```
SORT /R < filename
```

And you can have the sort done on a specified character position, rather than the first one, by specifying the position number. This for instance will sort the lines of your file in reverse order on the basis of what's at character position 16:

```
SORT /R /+16 < filename
```

As it stands, the command just displays its output on the screen. But you can redirect it, for instance to a disk file ...

```
SORT < B:INPUT.TXT > A:SORTED.TXT
```

Or to a printer:

```
SORT < B:NAMELIST > PRN
```

SORT is particularly handy when combined with other MS-DOS commands. This one is a classic:

```
DIR | SORT
```

It's being used as a filter for a DIRectory listing, and the resulting display will be a sorted directory. You can still use all the SORT parameters:

```
DIR B: | SORT /R /+10 > B:DIRLIST
```

This will sort the directory in descending order on the basis of whatever's at position 10 — which in a standard MS-DOS directory is the start of the filename extension. So you'll get a directory sorted by extension in reverse order, and it will be saved as a file on drive B called DIRLIST.

You might have to tart up the resulting text if you want to use it. The DIR command also produces a bit of lead-in text (volume name and a 'directory of drive B' header) plus the 'bytes free' footnote; these too will appear in the sorted directory listing — and SORT will have sorted them dutifully into the sequence of lines.

The FIND command

FIND will look through the nominated input to find occurrences of a specified character string. Again, it can be used as a stand-alone command, in which case the input will be a filename (in this case the name is not preceded by the input redirection symbol); or it can be used as a filter in conjunction with other MS-DOS commands.

To start with the simplest form:

```
FIND "string" filename
```

The 'search for' character string may be any group of letters, numbers, spaces and/or punctuation marks — provided they're enclosed in double quotes. And more than one filename may be provided, in which case the files are searched sequentially for any matches with the specified character string. For instance:

```
FIND " BAS " DIRLISTA DIRLISTB
```

(Note that the wildcard filename characters (asterisk and question mark) are *not* allowed, so you can't search a batch of files without specifying the names individually in full.) The two files DIRLISTA and DIRLISTB are probably directory listings of drive A and drive B produced by a DIR > DIRLISTA. This command would search them for any BAS references and output the matching lines from DIRLISTA first and DIRLISTB second. Incidentally, the blank spaces before and after BAS here will ensure that FIND won't pick up any filenames that happen to contain the string somewhere other than in an extension — so you won't get MSBASIC.COM in your output.

You can redirect the results of a FIND too:

```
FIND " BAS " DIRLISTA > PRN
FIND " BAS " DIRLISTA > BASLIST
```

The first of these will output the results of the search on your printer, the second will create a new file called BASLIST for them.

To include a quote in the character string, you have to enter two quote marks in succession — this will be interpreted as a single quote by FIND.

FIND isn't particularly clever. The command is very literal: the string in quotes must match *exactly* the string to be found. That applies to capitals, lower-case letters, and spaces. There isn't a match-any-pattern wildcard facility, either.

There are however several useful options that can be included in a FIND command:

- /V picks up all lines that do *not* contain the string
- /C displays the total number of matching occurrences in each file, *without* displaying the matching lines
- /N the line number of each line containing a match is shown to the left of the displayed line (meaning the relative line number — the number of lines from the start of the file).

As with MORE and SORT, FIND is probably most used as a filter:

```
DIR B: | FIND " BAS " | SORT > B:BASLIST
```

This takes the output from the DIR command and finds all the lines containing BAS. These are passed on to SORT which sorts them and outputs the results to the file BASLIST on drive B.

```
DIR | FIND "1-01-80" > PRN
```

Here you're printing a list of all files with that date-stamp — maybe to isolate those occurrences when you hadn't set the system date.

Putting it all together

So far, so good. You might be able to anticipate simple command-line uses for redirection, piping and the filterable commands; you should also see applications in batch files. But you can also put them to work in a more serious manner.

The classic application is a miniature database, where you can key in lines of ASCII-format text (using EDLIN, EDIT or your word processor). Each line constitutes one 'record', with a Return marking the end of the record. You can put a number of different items into that record by starting a new 'field' at a particular tab position; and as long as the new fields always start at the same place in the record, you can quite safely sort and select on any field.

The simplest and easiest example is also one of the most useful — a telephone index. Here's a few sample entries:

```
POTTER Simon      01-833 3501      journalist
AUDIOVISUAL LTD   01-911 4567      supplier
DAMSON GRAPHICS   01-222 1789      supplier
HYNDE Chrissie    010 331 454233   friend
PRINCE Leon       0272 450000      designer
DECKHAND Brian    01-833 3502      journalist
SUTCLIFFE Nancy   01-341 9919      illustrator
POTTER Alf        0822 444555      dad
APRICOT FILE      01-833 3501      employer
```

Here we have a 'database' file organised as follows:

Position	Content
0	surname or company name
17	phone number
33	comment

As it stands, this index — which for the sake of illustration has been saved on drive C under the less than original name PHONES.LST — isn't sorted at all. It just appears exactly as it was typed. So let's sort it alphabetically by surname or company name:

```
C>  SORT < PHONES.LST > PHONES.NEW
```

(I've saved it under a new name just to be on the safe side.) And now let's take a look at it:

```
C>  TYPE PHONES.NEW

    AUDIOVISUAL LTD   01-911 4567      supplier
    DAMSON GRAPHICS   01-222 1789      supplier
    DECKHAND Brian    01-833 3502      journalist
    HYNDE Chrissie    010 331 454233   friend
    APRICOT FILE      01-833 3501      employer
    POTTER Alf        0822 444555      dad
    POTTER Simon      01-833 3501      journalist
    PRINCE Leon       0272 450000      designer
    SUTCLIFFE Nancy   01-341 9919      illustrator
```

Alternatively you might have wanted them sorted in phone number sequence:

```
C>  SORT /+17 < PHONES.LST > PHONES.NEW
C>  TYPE PHONES.NEW

    SUTCLIFFE Nancy   01-341 9919      illustrator
    DAMSON GRAPHICS   01-222 1789      supplier
    APRICOT FILE      01-833 3501      employer
    POTTER Simon      01-833 3501      journalist
    DECKHAND Brian    01-833 3502      journalist
    AUDIOVISUAL LTD   01-911 4567      supplier
    HYNDE Chrissie    010 331 454233   friend
    PRINCE Leon       0272 450000      designer
    POTTER Alf        0822 444555      dad
```

Or in terms of the comment field:

```
C>  SORT /+33 < PHONES.LST > PHONES.NEW
C>  TYPE PHONES.NEW

    POTTER Alf        0822 444555      dad
    PRINCE Leon       0272 450000      designer
    SUTCLIFFE Nancy   01-341 9919      illustrator
    APRICOT FILE      01-833 3501      employer
    HYNDE Chrissie    010 331 454233   friend
    POTTER Simon      01-833 3501      journalist
    DECKHAND Brian    01-833 3501      journalist
    AUDIOVISUAL LTD   01-911 4567      supplier
    DAMSON GRAPHICS   01-222 1789      supplier
```

The MORE filter becomes relevant when you have a long list:

```
C>  TYPE PHONES.NEW | MORE
```

Or, more simply:

```
C>  MORE | PHONES.NEW
```

And finally, you use FIND to locate a particular record. For instance:

```
C>  FIND "POTTER" PHONES.NEW

    ---------- phones.new
    POTTER Simon      01-833 3501      journalist
    POTTER Alf        0822 444555      dad
```

Or to identify a particular phone number:

```
C>  FIND "222 1789" PHONES.NEW

    ---------- phones.new
    DAMSON GRAPHICS  01-222 1789      supplier
```

Or to pick out a group of people from the entries in the comment field:

```
C>  FIND "designer" PHONES.NEW

    ---------- phones.new
    PRINCE Leon       0272 450000      designer
```

As databases go, it may not be too fancy — but it's quick, easy, and efficacious. What more could you want? ◆

Device names

It may seem odd to the uninitiated, but a computer with MS-DOS treats the screen and printer as if they were files. This turns out to be somewhat useful since you can then carry out file-type activities such as COPY and TYPE while referring to these devices. The screen is called CON or CON:, which stands for console, when it is referenced as if it were a file. The printer usually goes by the filename PRN: or PRN.

These are the principal device names that MS-DOS can treat as 'files':

- CON — the 'console' — in practice, the screen and keyboard. It is used most frequently as an input device in the form COPY CON FILENAME, where anything you subsequently type at the keyboard will be saved under the filename when you press CTRL-Z (see below)
- PRN — an output device, almost always a parallel printer
- LST — same as PRN
- AUX — an output device, almost always a serial device on an RS232 port (eg a modem or plotter)
- NUL — a nonexistent dummy 'device' which can be used with commands that normally require *some* kind of device to be specified

To illustrate the flexibility all this provides, and incidentally to sneak in another example of redirection, here are three useful procedures that accomplish similar ends — both provide a typewriter-like way to get something on the printer without first creating a file on the disk. They're handy for labels, envelopes and the like. (The A> prompt is used for the examples, but it really makes no difference which drive you're on as long as you are in MS-DOS).

```
A> COPY CON PRN
```

This will let you type as many lines as you like direct to the printer. Use the Backspace key to move to the left and retype anything on the current line (and *only* the current line); press Return to move to a new line.

When you're finished, you have to tell MS-DOS to 'close the file' in the manner it expects. What it expects is a CTRL-Z and then Return. MS-DOS waits for the 'end of file' signal before it will return to its usual prompt.

At any time before printing, you may cancel what you are typing by pressing CTRL-C. This will get you back to the MS-DOS prompt without printing what is on the screen.

```
A> COPY CON B:TEST.BAT
```

This works in a similar fashion, except that anything you type on the screen will be saved in a newly-created disk file (called TEST.BAT in this case). Again, use the Backspace key to edit; press Return to move to a new line; and use CTRL-Z to finish. This is the quickest and easiest way of creating a short file on disk — including, for instance, the BLANK file used earlier with the DATE redirection example.

```
A> TYPE CON > PRN
```

The third example redirects the contents of device/filename CON to PRN. Since the contents of CON will be whatever you put on to the screen, it effectively produces an instant typewriter. It will let you type line by line, again using Backspace to edit and retype on the current line. Pressing Return will print the current line and move you on to a new line. You may cancel this at any time and get back to the prompt with the same CTRL-C combination.

Neither of these tricks will warn you of the end of the line, or prevent words from being chopped between two lines — but then neither do (pre-electronic) typewriters. Their advantage lies almost solely in the fact that they are quick and fairly efficient. ◆

Classified

Free Small Ads

Apricot Sales

■■■ APRICOT PC with twin double-density drives, 9in monochrome monitor, and 768KB memory. Complete with all the standard software, manuals, cables, 28 spare disks, and a complete SMART (v2.00) suite of software. This bundle cost one careful owner over £2,500 18 months ago; I can be persuaded to part with them for only £750. Call Peter any evening or weekends on 0990 28268.

Apricot Add-on Sales

■■■ MICRO FOCUS Level II Cobol and FORMS-2. We paid £1,365 plus VAT: offers invited. S Budd, Loopmead Ltd — 0903 65287.

■■■ RAM AND PRINTER — 512KB expansion memory with RAM disk software: £120 inc p&p and VAT. Also Star Gemini 15X printer — up to 233 columns on 15in paper; friction, roll and adjustable tractor feeds; Epson-type control codes; graphics too: £220 plus p&p. Phone Andy 0557 30624 (day) or 06445 244 (evening).

■■■ OMNIREADER optical character reader — inputs text directly into computer without typing. Includes software for Macintosh and Apricot. Price £50; contact Phil Croucher, 026 477 3739.

■■■ APRICOT Xi/PC 9in MONITOR — black, excellent condition, only four months old and little used. £50. Phone J Cundell, 0442 41221 (work) or 0525 376011 (home).

■■■ APRICOT INTERNAL MODEM with Communiqué; also Apricot-format dGRAPH and WordStar; also Epson MX-80 ribbons. Best realistic offer secures any item. Contact 065 122667.

Wants

■■■ WANTED: software for Apricot PC on single-sided disks for private buyer. K R Dodd, 57 Evelyn Drive, Hatch End, Pinner, Middlesex.

Services

■■■ DISK RECOVERY: problems with deleted files, scrambled hard disks, and the like? We can (probably) help — no fix, no fee! Call Roger Gann at TP Group on 01-833 3501.

■■■ FILE TRANSFER: we can convert files between Apricot single- and double-sided, IBM PC format, PC/AT format, and Sirius — cheap! Call Roger Gann at TP Group on 01-833 3501.

■■■ TURNKEY CONSULTANCY: we can recommend and/or implement systems using popular packages or custom-written software. We can also convert you from Apricot or Sirius to another species. Modest prices, guaranteed work. Call Roger Gann at TP Group on 01-833 3501.

Misc

■■■ COMMODORE CLEAR-OUT — FREEBIES! Come on, doesn't anyone want this stuff? A Spring-clean has thrown up a few items of stock relating to a previous venture in Commodore home computers. Specifically, we have a dozen or so copies of Dennis Jarrett's Complete Commodore 64 book and many cassette dust-covers. FREE to a good home — ie a non-profit cause like a school. Contact Roger Gann on 01-833 3501. ◆

The full list of Classified Ads is featured in the DEALS FOR READERS catalogue published every other month. This page summarises ads new this month and useful-sounding services.

Ads are free to all subscribers of APRICOT FILE.

Letters to the Editor

Readers' Writes

Gold in the hills

Brian Read's article *A cheap way into Telecom Gold* was not entirely fair on the subject of Telecom Gold costs. I send around 200 messages a month at 300 baud, receive a similar number, download another 10,000 words per week (mostly articles) at 1200 baud and send a few Telexes — all for under £30 a month. First-class post would be slower and more expensive.

The secret is to use the WPMAIL facility for batch mailing — ie send a whole day's messages, prepared off-line, in a single upload. Even at 300 baud, which is roughly 300 words a minute, ten average messages rarely take more than five or six minutes — and certainly not the 30 minutes Read supposes.

With multiple mailings using on-line address lists, the advantages of electronic mail become even more pronounced. The same ten letters can each be sent to multiple recipients for almost no extra cost.

Finally, Read evidently thinks his time is worth nothing. Personally I find not having to print out the letters separately, mess with paper, envelopes and stamps, and rush to the local box to make the last post a great time-saver which translates into significant savings.

Jack Schofield
Editor, Computer Guardian

PS The above letter was one of 12 uploaded at 300 baud in a total session time of six minutes or less.

Environmental problems

Looking for ways to unclutter my hard disk, I moved all the ACTIVITY files on to a floppy. Recently I copied them back again to enlighten a friend about this terrible suite of programs. With it loaded I summoned up SKETCH and DIARY, only to receive the message that I was 'out of environment space'. Would some kind expert please explain?

J E Bullin
Ford, Du Cane & Co, Rishangles, Suffolk

As it happens, Roger Gann is currently beavering away on the definitive article about this subject — should be ready for our next issue, maybe the one following. But in brief, the MS-DOS 'environment space' is an area of memory that holds the currently-set variables for COMMAND.COM — such as any PATHs that apply, and the format of the MS-DOS prompt. The environment space is not very large, and in this case it's likely that something set on boot-up leaves too little room for SKETCH and/or DIARY to run.

Review

Make more of WordPerfect

Ansible's WordPerfect utilities disk

By Mike Florey

WordPerfect is widely acclaimed as one of the best word processing packages around; almost certainly it's the best for use with the Apricot. And in an impressive act of faith Sentinel Software has released the latest version in Apricot format — see Paul Lavin's review in APRICOT FILE Release 2.7.

Problems

There is more to changing word processors than parting with the cash and sitting down with the on-disk tutorial. What about the last two or three years' worth of letters, reports, your CV and whatever else you have been writing? Sentinel is aware of the problem and ships a file conversion program which can convert to and from WordStar, MultiMate and IBM Document Content Architecture (DCA). But pity the poor Apricot owner who started word processing with the bundled SuperWriter package ...

And when you finally get to grips with WordPerfect, you'll discover that the Speller lexicon is an excessively large and poorly-Anglicised version of the American original. The Thesaurus is all-American, without even a token concession to English like 'colour'. The land that produced names like H W Fowler and C T Onions should have a law against this kind of abomination.

Finally, the macro facility is limited in practical use by the absence of any means of editing: keying errors are written into a macro as it is defined, along with any corrections made, and complicated or chained macros become very difficult to construct. Despite its undoubted power and versatility, even WordPerfect 4.2 has shortcomings.

Ansible Information, responsible for the well-established A.S.D.1 and A.S.D.2 disks for SuperWriter (available from DEALS FOR READERS), has produced the Ansible WordPerfect Disk to overcome some of these problems — and particularly to ease the path of those migrating from SuperWriter. The disk carries a macro editor, a genuine English lexicon, SuperWriter-to-WordPerfect file converter, a collection of macros, and a few handy utilities.

Macro Editor

This permits the creation and amendment of macros and uses a display modelled on the WordPerfect 'Reveal Codes' screen for keyboard commands. The function keys invoke the same commands and menus as would be the case in normal use of WordPerfect. The MicroScreen keys are not operative (use CONTROL with numeric keypad numbers). Because some keys are used to control the editor, the WordPerfect commands normally generated by those keys are obtained by preceding them with the Escape key.

In use it seems to work well and quickly. The main drawback is the need to be familiar with every WordPerfect command and menu structure; Ansible's macro editor would be much simpler to use if menu options could be displayed during editing as they are during macro definition within WordPerfect.

On the other hand, this would entail additional programming effort; and there is always the possibility that Sentinel might become possessive (or do I mean paranoid?) about its copyright position.

English spelling

Some 100K of the Ansible disk is taken up with AILEX.WP, a replacement spelling-check lexicon based on the word-list supplied to SuperWriter users on the A.S.D.1 disk. It is half the size of the official offering, runs faster (because of the reduced size), and is very comprehensive — though devoid of most technical jargon.

File conversion

SWP.COM is a versatile and easy-to-use program for converting SuperWriter document files to WordPerfect format. For reasonably straightforward files the conversion process is remarkably thorough; but it's not comprehensive — SuperWriter's more esoteric embedded formatting commands are stripped out without conversion. So are margins, tabs and the mail-merge commands.

Watch out for pitch and font numbers if you were in the habit of using proportional spacing. \PROP ON converts as 12-pitch, font 1, whereas WordPerfect tends to assume that font *3* is proportional. (This is likely to be significant for users of dot matrix printers and, possibly, laser page printers.)

A macro called SW.MAC tidies up the converted file and resets headers and footers to the current margins in readiness for further editing or printing. You should make any adjustments necessary to margins before running SW.MAC.

WordPerfect macros

Macros are one of the great strengths of WordPerfect (and of SuperWriter, for that matter.) There are several on the Ansible disk, but their usefulness depends very much on each user's needs: who is so disorganised as to need macros to switch all bold into underline and vice versa within a document? The VATIN and VATOUT macros were presumably created by Ansible for its own business use and might not be of relevance to anyone else.

Still, whatever your interests, one of the best ways of learning how to use the macro editor is to use it to experiment with the supplied macros -- preferably on a spare disk, unless you like living dangerously!

Files on the Disk

WM.COM	Macro editor; calls WM.OVL
SWP.COM	File conversion utility from SuperWriter to WordPerfect
SWDEMO.DOC	Demonstration SuperWriter file containing all the commands that can be converted by SWP.COM
SW.MAC	Tidies up files newly converted by SWP.COM
AILEX.WP	(Real) English lexicon for Speller — copy to working disk and rename as LEX.WP before use
TIM.COM	Puts day, date and time on the screen status line
TIMF.COM	Version of TIM.COM for F series and Portable
WPC.COM	Improves cursor control in WordPerfect version 4.1
WPF.COM	Overcomes some screen-handling problems when running WordPerfect 4.2 on the F series with early versions of the ROM BIOS. A Beta test release with program development apparently still continuing
INTFIX.COM	Corrects screen handling when running WordPerfect with PC/Xi BIOS 2.4 or earlier
CURSE.COM	Alters cursor type
ALTD.MAC	Macro to delete from cursor to end of text
ALTI.MAC	Switches in and out of italics
ALTJ.MAC	Another way of switching in and out of italics
ALTU.MAC	Converts current word to upper case
ALTL.MAC	Converts current word to lower case
ALTS.MAC	Saves current file without leaving document. The filename must already exist on the disk directory
BRET.MAC	Allows you to go to the bottom of a document, view or make changes, and return to original position. Calls UP.MAC, DOWN.MAC and DEMARK.MAC
TRET.MAC	As for BRET.MAC but takes you to the top of a document
BU.MAC	Changes all bold to underline. Calls BU1.MAC
UB.MAC	Changes all underline to bold. Calls UB1.MAC
BUB.MAC	Changes all bold to underline and all underline to bold — no comment! Calls BUB1.MAC to BUB7.MAC
UNB.MAC	Removes emboldening from a block of text within a bold section. For WordPerfect 4.2 only
UNU.MAC	Removes underlining from a block of text within an underlined section. For WordPerfect 4.2 only
VATIN.MAC	Sets up 'math' documents for VAT Input return. For WordPerfect 4.2 only
VATOUT.MAC	As for VATIN.MAC but for VAT Output
DB.MAC	Macro to change logged drive from A to B — best run on loading WordPerfect (WP/M-DB)
DBD.MAC	As DB.MAC but displays directory — use in the same way
ANSIBLE.CHR	An alternative screen font
WPAI.CHR	Another of Ansible's alternative display fonts
READ.ME	Copyright notice, documentation update and information on developments under way for other Ansible products

Also on the disk ...

The panel gives brief details of other macros, and also lists the utilities on the disk. A couple of them look handy: TIM.COM duplicates one of the features of Ansible's Acolyte disk (a parcel of utilities for Apricot users with and without SuperWriter — also available from DEALS FOR READERS). It puts the day of the week, date and time (in 12- or 24-hour format) on the status line.

And CURSE.COM is not a blasphemy generator, it permits you to select cursor shape (block or underline) and whether it is solid or flashing.

Documentation

This is to the usual high standards we expect from Ansible: logical, comprehensive and clearly written — though the A4 booklet isn't so convenient as an A5 ring binder. Those accustomed only to Apricot plc documentation standards have a pleasant surprise in store.

In conclusion

No attempt has been made to correct the deficiencies of the WordPerfect Thesaurus. But at £57.50 inclusive of carriage and VAT, this disk gives excellent value for money; it can be recommended strongly to anyone moving up from SuperWriter, those who want a decent spelling checker, and WordPerfect users who want to be able to edit macros. ◆

Ansible WordPerfect Disk

Description:	A clutch of enhancements and utilities for use with Apricot-format WordPerfect. Applies particularly to version 4.2, though most will work ok with 4.1.
Supplier:	Ansible Information
Address:	94 London Road Reading Berks RG1 5AU
Telephone:	0672 62576
Price:	£57.50, inc VAT and despatch. There are discounts for existing Ansible customers and special deals for those who buy WordPerfect from Ansible at the same time.
Configuration:	Any (non IBM-compatible) Apricot. Reviewed on an Apricot PC with twin 720KB drives and 768K memory, running with BIOS 3.1.
Conclusion:	An essential adjunct to WordPerfect — very good value.

Advanced SuperWriter

Part One: Printer Control

By Bob Smith

The trouble with working with a fussy bunch of people such as our users is that they are never satisfied. Take that printer control method I described in APRICOT FILE Release 2.4. "Not good enough", they cry. Why not? Well, to use it, they remind me, you have to be in MS-DOS — specifically, you have to be at the MS-DOS prompt.

Turns out our users generally need the facility when they're about to start printing from the middle of something else. So they have to exit from the something else, go to the printer menu, send the setup codes, and then go back into the something else to start their printing.

"Can't you do something about it?" they say. "Sorry", says I. "Well ... since I know you use SuperWriter a lot, I can help you out when using that."

Before I describe what I did with SuperWriter, let me explain why I had to say no to a generalised facility.

Multitasking

The problem is simple, the solution is not. What my Apricot enthusiasts need is to be able to do two tasks at the same time — or at least suspend one task to do another, then resume the first task at the same place it was left off. In conventional computing terms, what my users need is a multitasking environment.

So what about multitasking? Because it would need to run all our existing applications, it would have to be able to control and take care of the MS-DOS environment that our programs expect. The only commonplace systems that are currently available with this ability are Concurrent DOS from Digital Research and MS-Windows from Microsoft.

But both of these have snags. For a start, they are both sledgehammers to crack nuts as far as the printer control problem goes. They both take up an enormous amount of memory just to run themselves. In the case of MS-Windows, your software also has to be configured to use the MS-Windows environment; and without at least an 80286 processor (non-Xen Apricots use the earlier, less powerful 8086 chip) to run the operating system, the whole machine will be slowed down to a snail's pace. And for all this extra burden on the processor, it's only suspend-and-resume, not true multitasking.

Concurrent DOS *is* multitasking; it also has a better reputation for performance ... but a worse one for compatibility of MS-DOS application software. A lot of it *does* run without change; and in certain circumstances (where the micro is being used solely for development, for example, with an experienced user who is aware of the limitations) it can be tremendously productive. But for the general user, there is no certain fraughtless improvement in adopting this environment. It may turn out to be at the expense of not being able to run some of your favourite software, particularly the so-called 'badly-behaved' programs that bypass BIOS calls and address hardware directly, such as some do for screen display.

Demons

The non-conventional approach — which these days is probably the normal one — would be to have a program resident in memory all the time. It could then be activated by a certain keystroke. You would select the required setup codes from a menu as before, send them to the printer, then go straight back into the original program just where you left off. This kind of suspend-and-resume facility is all you need for a simple job like print controls.

In theory, I'm not against memory-resident programs (more properly called TSR — Terminate-and-Stay-Resident). Indeed, I've been tempted to try writing them myself. We ourselves usually have one up all the time, very occasionally even two. But that's the limit. If you read the same magazines as me, you also will be aware of the horror stories that are beginning to appear about the conflicting non-compatible TSRs that lock out your system, corrupt your other programs in memory and may even take a swipe at your disks! No wonder, the 'in' name for these rascals is 'demons'!

What I really want is one TSR that can do everything. But since my everything is bound to be different from other people's everything, how could such a program come about? The next best solution is a TSR that allows me easily to code and add the functions that make up my particular everything.

Better still (in theory) is software that allows me to turn ordinary programs into TSRs, such as appears to be the claim of Deskset with its Pop-Anything facility that appeared at the Which Computer Show. Intrigued by this possibility, I rang up Sagesoft to check the theory.

Pop-Anything

It turns out Pop-Anything doesn't actually turn ordinary programs into TSRs themselves. It works by calling up a second command processor, another copy of COMMAND.COM, which thus allows access to the MS-DOS prompt. From the prompt, you can run any other program just as if you had exited the current program and DOS was waiting for its next instruction.

To get back to the current program (the one you just left) you must first quit the program you're running (to get back to the prompt), then type 'EXIT' to unload the second COMMAND processor.

That way you should find yourself back in the original program.

Even though it's not quite what I'd hoped for, it still sounds a terrific function. But while Sagesoft claims it is not aware of any problems caused by the use of Pop-Anything, it is worth noting that it means there can be up to three programs resident in memory all at the same time — the original program, Deskset and the program called up by Pop-Anything. Deskset apart, it is unlikely that the other two were designed to co-exist together.

Before using the facility too widely, it would be prudent to test out the co-existence with at least your usual programs, and in as many different but likely combinations that you could expect to occur. And I would advise not using Pop-Anything to run any low-level utility software that might expect the whole of memory to itself, such as Norton Utilities.

The other class of potentially (but not necessarily) dangerous programs covers that software which grab memory as it needs it for processing data (or worse still, grabs all remaining memory when first loaded). Most modern spreadsheets and word processing software are in this category; and it would be a good test to load big models or documents so as to fill up memory deliberately to see what happens.

As well as programs, you can obviously issue any MS-DOS command while the second command processor is in place. You could, for instance, change the current directory and/or the current drive. Very usefully, you can keep the change in effect when you return to the main program: so you could use this facility for programs like WordStar that cannot recognise path names, those that require you to exit the program in order to edit files in other than the initial directory.

But beware of another potential trap. Programs like WordStar are only partly loaded into memory. The other parts (overlays) are only loaded when a function not in the base program is called for. If Deskset has moved the current directory, how does WordStar know where to find its overlay?

And what happens if you were in the middle of editing a file when you changed the current directory — particularly if the document is very large and only *part* was actually in memory? You might just have cut the thing in two, one half in the original current directory and the other in the new current directory! DPATH might prevent the overlay problem, but not the second.

Moral — you gotta know what you are doing or you can really mess up with this little firecracker!

Pop-Anything is not unique. WordPerfect and PC-Write both have this facility to exit temporarily to MS-DOS; and some languages, like MegaBasic, allow programs written in them to do the same. But Pop-Anything is the only commercial product I know for ordinary MS-DOS use that can give you suspend-and-resume at any and all times.

POP-DOS

While on the subject of Deskset, I ought also to correct an impression I gave in the last *Report from Bandage & Plaster* that the POP-DOS facility allowed access to the MS-DOS prompt for DOS commands to be run. As you can see from the description above, this function is provided through Pop-Anything. POP-DOS is in fact a module that allows some manipulation of directories and files such as copy, display, print, rename and delete of any file. You can list current directory contents in the MS-DOS manner and also in a format that displays the first 20 characters of each file; and by temporarily changing the current directory, you can list any other directory as well.

I like the sound of this module. It would allow me to go hard-disk-a-roaming at any time.

Another feature of POP-DOS is its typewriter mode. You use this for short pieces of text you want printed as you type (actually when you press Return), like addresses on envelopes or labels. Even better, you could use it to line up labels or other special stationery. Now *that* I really could use.

Finally, POP-DOS could solve my users' grumble about leaving SuperWriter to set up the printer (so would Pop-Anything, of course, since it allows access to any function available from the MS-DOS prompt — such as my Print Control Menu). POP-DOS can send sequences of escape codes (in decimal format and in the range 0-127) to the printer and so set up the printer any way you want — providing you can remember what escape sequences do what!

Truth will out

Deskset consists of separate modules that are also loaded separately so you can 'configure' it to your taste and memory capacity. Its memory requirements are somewhat fierce; the seven modules available from Sagesoft total 189K! (Thinks — don't I know an ex-SF-writer turned programmer from Reading who writes a column in this magazine and who gets real worried if his TSR programs get bigger than 7KB?)

Deep breath — here comes admission time. Owners of non-IBM-compatible Apricots can skip this section and read the rest of the article. As for you IBMers, you lucky lot, you alone can run this software and so you don't need the expertly crafted techniques that follow. Come to think of it, has anyone got an IBM version of SuperWriter? Let me know, I might want to get one myself ere long.

I did try — very, very hard — to persuade Sagesoft that it was worth putting Deskset on the original Apricot, but no go. Once upon a time, big companies fell over themselves in the rush to get software on the Apricot: was it so long ago? Well, who needs 'em. At least we can comfort ourselves with the thought that for original Apricot users we've resurrected the cottage industry that bred so many of these now-successful firms in the first place. And ten years from now, F1s will be the Morris Minors of today and Xis will be the MGBs, lovingly cared for (in cottages) everywhere.

Finally, I did say that perhaps I could code a TSR for print control myself. Well, maybe, but just to save a tiny bit of time and inconvenience — and that really only very occasionally. I have to apply the 80/20 rule which says that the effort ain't worth the result, folks.

Formatting from SuperWriter

I assume everyone knows that you can send escape sequences from SuperWriter to your printer via the OUT command? If you don't, it's quite simple to understand; OUT is one of those special formatting commands that you can place anywhere within a document and which tells SuperWriter to do something — in this case, to send set-up codes to your printer. It is actioned only when printing the document, and since it is an instruction to SuperWriter it is not itself printed.

The OUT command is enclosed by backward slashes (see description of CMD below for when it is not). That shows SuperWriter it is an instruction and not part of the text. The general format of the command is ...

\OUT n1,n2,n3, ... \

... where the \OUT is followed by a series of escape codes in the form of decimal numbers between 0 and 255. For instance, to change from 10 to 12 pitch for an Epson RX-80 you need to send ESC M; for that you would code the OUT as ...

\OUT 27,77\

You don't usually need the OUT command for the simpler formatting such as emboldening, underlining and italicising. These can be turned on by the ^Pb, ^Pu and ^Pi commands placed before the first word you want so formatted. A second ^Pb, ^Pu or ^Pi turns them off.

Sub- and superscripting can also be handled this way, with ^P> and ^P< respectively. Note that for these controls you must use the opposite control to turn off the effect; if you start superscripting with ^P<, you need ^P> to return to normal script.

With all of these, formatting is also turned off by a Return — if one appears before your second print control.

Apart from underlining, which seems to work on any printer, naturally you need a printer that supports the ^P effect you're trying to get. You also need to set the PID printer identification code in SW.DEF to allow SuperWriter to translate the ^P control into the appropriate escape sequence for your printer.

But if your printer is not one of those recognised by SuperWriter, you'll have to use the OUT command anyway instead of ^P to get these effects. Even if you have a recognised printer, you will still have to use the OUT command for any controls not handled by the ^P set if you wish to send them to your printer during printing — change of pitch, for example.

Keyed Section Files

So by including the appropriate controls as OUT commands in their documents, my grumpy users would not have to exit SuperWriter to set up the printer.

To make it easier, I added two facilities. I put a version of the PFM menu in the SuperWriter subdirectory so that it could be displayed on screen through access from the UTILITIES/LIBRARY function. (The XQT file called PFM takes you from your edit screen to this display and back again.)

The menu would then remind you of the name of the appropriate parameter for the desired print control. You can use this name to retrieve the correctly coded OUT command from the second facility I set up — a file of all the matching OUT commands for the PFM menu.

In SuperWriter jargon, this is a 'keyed section file'. It's just an ordinary document really, but it is laid out in a special way so that SuperWriter can recognise that it is split into chunks. Each section starts with and is identified by a 'key' which also serves to show where the previous section ends. Each key is preceded by a form-feed character (which appears as a dim exclamation mark) and ends with a Return.

Thus the 'key' is not restricted to a single word. Everything after the Return up to the next form feed (or end of document) is then the 'section'.

I called my keyed section file PFMLIB, and you can see from the example that the keys correspond to the print parameters from the PFM menu. The real keys then are P5, P6, DF, L, and so on; and the remainder of each key line, up to the Return, is really just a fuller description or comment.

Each section comprises an OUT command with the relevant printer escape sequences in decimal. (T — test print — is the exception to this rule; I'll explain this later.)

To get the OUT commands from PFMLIB into your document, you go to the Main Menu and select UTILITIES, then the LIBRARY option, then the Include Document option to bring up the following selections:

```
Specify Include file
Include next section
Restart Document
Include Keyed section
```

Pick the first of these to enter the name of the keyed section file PFMLIB, and follow up with the fourth option to enter the key of the section you want included in your document.

You can select and include as many OUT commands (sections) as you wish; but as you do, the internal pointer for PFMLIB will be moved down the file. If you then try to include a section that precedes the last one included, you may get a 'Not found' message. No bother, just select 'Restart Document' to start the search from the beginning again.

Both the 'Include next' and 'Include Keyed' options will display the section found on the

screen. You then accept or skip the section. And you don't have to specify a full key; a partial key simply matches with the next occurrence of those characters in the following keys. Note also that the keys are not pulled into your document, only the sections.

If you try to include from PFMLIB directly, by way of ESC I from the edit screen, you'll find that you are forced to accept at the minimum a full screen's worth of OUT commands together with their keys. That's ok, but you'll have to edit out everything you don't need.

You can browse the keyed section file (after a fashion) by selecting the 'Include next section' and then skipping the section displayed. Since this returns you to the option menu, you have to select 'Include next' again to see the next section in the file (surprise, surprise).

Printer Controls — running PF.XQT

Getting the appropriate OUT commands into your document under edit in this way probably sounds like a lot of hard work. Well, it's easier than it seems — but there are some drawbacks.

First, you have to know how to get to the 'Include Document' menu screen, then include the sections that you want. Second, unless you carefully edit the OUT commands you've just included, when you come to print you may find that there are extra unwanted blank lines in your printed output — they correspond to the Returns in the 'sections' and the paper movement caused by sending print controls themselves).

If you want to change pitch in the middle of the document, or something like that, you have to sort it out for yourself. On the other hand, if you just want to set up the printer once and for all before you do any printing, use PF.XQT. This is the SuperWriter equivalent to sending print controls from the Print Function Menu that I described in APRICOT FILE Release 2.4.

Here's how it works. You start in the document under edit; you've written all the text, polished it up several times, read it many more, you're as pleased as punch and you are now ready to print. You turn to the printer ... and your face drops as you see the last thing it printed was a spreadsheet in compressed mode at 8 lines per inch. But then you remember you can reset the printer *from within your document* by only a couple of keystrokes and PF.XQT.

So stay in your document and start up the PF.XQT (by ESC X). This takes you straight to the 'Include Document' menu and pauses for you to enter the keys for the OUT commands you wish to send. Now, select the OUT commands *in reverse order to that in which you wish to send them to the printer* — when the sections are included into the document, each successive OUT command pushes in front of the previous one.

So work out what controls you want *before* you select them. Until you get used to it, always select the Ttest section first (to print last) and then you will see the final effect of the OUT commands you picked. As before, it sounds more difficult on paper than it really is ...

When you've finished your selection, press '&' to let the XQT continue. PF then returns you to the edit document where you will see all those OUT commands inserted into your text. But they won't be there long. PF places block markers back and front of them, writes the block out to a temporary file called PFMOUT, then deletes the block just written from your document.

Next it goes to the 'Print Menu', selects PFMOUT as 'another Print File', prints it, returns to the 'Main Menu', selects 'UTILITIES', then deletes PFMOUT from the disk. Finally, PF returns to the original document.

Meanwhile, you just used up a page of paper to set the appropriate controls for what the printer is about to receive — your document. A hint: pin up near the screen a list of the OUT commands stored in PFMLIB, so that anyone can see both them and the order they are in the file. Best order is alphabetical, of course.

Printing PFMLIB

Try using SuperWriter to print PFMLIB direct. Two things happen. Every OUT command is sent to the printer in turn, so continually changing the set-up: and every form feed ejects a page of paper. At least the keys are printed (one per page)!

So you cannot print PFMLIB directly. Step forward another clutch of XQTs! Bring PFMLIB into the edit mode, then execute CMD.XQT, FFDEL.XQT and PTEMP.XQT in that order.

- CMD.XQT simply goes to the top of PFMLIB and inserts the formatting command CMD as \CMD$ — this tells SuperWriter that the character by which it will recognise any following commands will be a dollar sign, not the backslash. In other words, at a stroke we have turned the OUT commands into normal text.
- FFDEL.XQT works its way from top to bottom of PFMLIB, removing the form feeds. When it's completed its work, stop it by pressing ESC then X, then ESC again. This action will remove it completely so that you can start up the next XQT.
- PTEMP.XQT saves the modified PFMLIB under the name of PTEMP, prints it, deletes it, then quits SuperWriter.

So there you are. It sounds complicated, but it's all quite easy. I'm happy to say that my users at Bandage & Plaster love it — for the moment! ◆

PFM.XQT — display Print Function Menu

```
^[~          [go to 'Main Menu']
^K~          [up one — select 'UTILITIES']
~            [select 'LIBRARY' option]
~            [select 'Display document']
PFM.MNU~     [name of document to be displayed]
&            [suspend XQT while you look at print
             function menu — note that message at
             bottom of screen says "Press any key to
             continue"; ignore this, and continue by
             pressing '&' when you've seen enough to
             return to 'Library file access']
^[           [return to 'Utility selections']
^[           [return to 'Main Menu']
^[           [return to document under edit]
^Z           [stop XQT dead]
```

PF.XQT — send controls to printer

```
^[d/             [delete all block markers — in case
                 any exist]
^Q2/             [insert a block marker at cursor
                 position]
^[~              [go to 'Main Menu']
^K~              [up one — select 'UTILITIES']
~                [select 'LIBRARY' option]
^J~              [down one — select 'Include
                 document']
~                [select 'Specify Include file']
PFMLIB~          [name of file]
^K~              [up one — select 'Include Keyed
                 section']
&~               [pause the action — see text]
^[               [return to 'Library file access']
^[               [return to 'Utility selections']
^[               [return to 'Main Menu']
^[               [return to document under edit]
^Q2/             [insert second block marker — see
                 text]
^[cwPFMOUT~      [write block to file PFMOUT]
^dby             [delete block just written out]
^[~              [go to 'Main Menu']
^J^J             [down two — select 'Print document']
^K^K             [up two — select 'Secondary Print
                 functions']
^K^K             [up two — select 'Specify another
                 Print File]
PFMOUT~          [name of print file]
^[               [return to 'Printing options']
~                [select 'Format and PRINT document']
^[               [return to document under edit]
^[~              [go to 'Main Menu']
^K~              [up one — select 'UTILITIES']
~                [select 'LIBRARY' option]
^J^J             [down two — select 'Delete
                 document']
PFMOUT~          [name of file to be deleted]
Y                [confirm delete]
^[               [return to 'Utility selections']
^[               [return to 'Main Menu']
^[               [return to document under edit]
^Z               [stop XQT dead]
```

CMD.XQT — change command character to $

```
^T           [go to top of document]
\CMD$~       [insert this as text]
^T           [go back to top]
^Z           [stop XQT dead]
```

FFDEL.XQT — delete form feeds

```
^Q3f         [find next form feed]
^G           [then delete it]
^[xFFDEL~    [execute itself again]
```

PTEMP.XQT — print edited file without saving

```
^T           [go to top of document]
^[s          [go to 'Save your document']
^J           [down one — select 'Rename and save
             document']
PTEMP        [name of file to save]
~            [do the save and return to 'Main
             menu']
^J^J~        [down two — select 'PRINT document']
PTEMP~       [name of file to print]
~            [select 'Format and PRINT document']
^[           [return to 'Main Menu']
^K~          [up one — select 'UTILITIES']
~            [select 'LIBRARY' option]
^J^J         [down two — select 'Delete document']
PTEMP~       [name of file to be deleted]
Y            [confirm delete]
^[           [return to 'Utility selections']
^[           [return to 'Main Menu']
^J           [down one — select 'QUIT']
```

PFMLIB — keyed section file containing print controls

```
!P5     5 c.p.i.
\OUT 27,91,48,119\

!P6     6 c.p.i.
\OUT 27,91,49,119\

!DF     Default settings [P12+LM6+ST+DR]
\OUT 27,91,53,119,27,91,54,115,27,91,48,109,27,91,48,121\

!L      Letter settings [P12+LM6+MD+NL]
\OUT 27,91,53,119,27,91,54,115,27,91,49,50,109,27,91,49,121\

!T
         1         2         3         4         5
12345678901234567890123456789012345678901234567890
The quick brown fox went into the garden to sleep.
THE QUICK BROWN FOX WENT INTO THE GARDEN TO SLEEP.
```

Readers' Response

The SoapBox Analysis

What you think about three issues of APRICOT FILE

We've been running the SoapBox since Release 2.3, our December 1986 issue, in an attempt to generate some positive feedback about the content of APRICOT FILE. It's time to take stock of the input on the first three of those issues.

The disappointing news is the low level of response — less than three dozen for the three issues, and the bulk of them for the December 1986 SoapBox. That's why we haven't produced a summary before now. We wanted to get a sample that was statistically rather more valid.

We're inclined to take the view that non-response equates to non-cancellation of subscriptions — in other words, the bulk of readers seem generally happy with what we're doing. Still, it would be nice to have that verified ...

Overall impressions

And the SoapBox responses do confirm this impression. On average, our articles in the three issues scored **3.5** (with 5 as the best rating and 1 as the worst). Three of our regular items scored **4** or more — David Langford's *Disinformation Column*, the news pages, and *Readers' Writes*. Bob Smith's second *Bandage & Plaster* column (January) also hit **4.0**, and so did Lindsay Doyle's *Font Tricks* Workshop in February.

Even better from our viewpoint, nothing we've printed in those three issues did worse than **2.5**, which is the midpoint. The occasional *Diary* page, which we run only when we don't have anything more meaty to occupy half a page, scored **2.5**; but then it had a couple of fans who gave it **4**. And the February *Bandage & Plaster* also rated **2.5** — but there were only two SoapBox returns for that issue.

That's also one reason why the February issue is the only one to fall below the midpoint as a whole. Averaging the ratings for all articles in each issue, December came out best with **3.6**. Of course, it's a statistical fact that you're more likely to get an extreme result with fewer inputs.

The Soapbox: articles rated

◆ REGULAR ITEMS		3.8
Readers' Writes	4.3	
Disinformation	4.1	
News	4.1	
Doodles by David	3.7	
Free Small Ads	2.9	
Diary Dates	2.5	
◆ WORKSHOPS		3.2
Bandage & Plaster (Jan)	4.0	
Font tricks	4.0	
RAMdisks	3.5	
Printer control	3.5	
Bandage & Plaster (Dec)	3.4	
Turbo Pascal patch	3.1	
Handling dates	3.0	
Overseas mail spreadsheet	2.7	
Bandage & Plaster (Feb)	2.5	
◆ REVIEWS		3.2
Xi-10	3.7	
Books	3.5	
Workmate	3.5	
Xen-i	3.4	
MT290	3.0	
TopClass	2.8	
◆ MISC FEATURES		3.3
Mobile computing	3.5	
New Year Reflections	3.0	

The Soapbox: issue by issue

◆ OVERALL	
Responses	33
Article average score	3.5
Issue average score	3.2
◆ DECEMBER 1986	
Responses	25
Article average score	3.5
Issue average score	3.6
◆ JANUARY 1987	
Responses	6
Article average score	3.3
Issue average score	3.3
◆ FEBRUARY 1987	
Responses	2
Article average score	3.4
Issue average score	2.8

Bouquets and brickbats

Not all SoapBox respondents took the opportunity to comment on the newsletter and request specific articles, but remarks about APRICOT FILE were generally favourable. It's always gratifying to receive messages like "I would like to congratulate you all on an excellent and informative magazine" and "obviously a lot of care has been taken with this issue — well done".

Equally it's comforting to know that we're largely on the right track: "APRICOT FILE gives me a great number of tidbits that I probably wouldn't find elsewhere" is typical.

As for the columnists, "Langford and Doyle have the right style". In particular, David

Langford's *Disinformation* attracted universal favour: "always interesting ... very useful — often I get tips here that I don't find anywhere else ... full of gems — amazing what's been left out of the standard documentation ... a veritable goldmine of information, humorous but to the point ... knows his stuff, but human too ... excellent". In short, "more, please". And "we agree with the view that "Langford has the imagination that a good teacher needs in putting his subject across".

Doodles by David St.John Wallis polarised opinion rather more — "just what the doctor ordered" versus "a waste of space". The prevailing mood seems to be summed up by "usually useful, but presentation is overblown, wordy, self-consciously cute" or "varies wildly — occasionally essential, frequently useless"

Workshops generally produced a favourable response, and so did the reviews we ran in the period ("your reviews are independent and helpful; I particularly like the Summary panels at the end"). With both types of article readers seem prepared to put up with some subjects of little or no immediate relevance to their own needs on the grounds that our coverage is wide enough to provide something of use sooner or later.

Specifics

Some of the advice was frankly contradictory. A plea for " more information and fewer reviews" arrived in the same post as one for "more reviews, more analysis of new software and hardware". The general feeling seems to be that more practical hints and tips is what's required — or as one writer put it, "more reviews and workshops, less of small ads and the Davids".

Well, we're reluctant to drop the Davids in view of their consistently high score in the reader satisfaction stakes. But the free Small Ads page did collect a lot of adverse comment, largely because it "changes too little from issue to issue". From this issue we're taking the full list out of the magazine and putting it into the bimonthly DEALS FOR READERS broadsheet; only brand new arrivals and useful-sounding services will be featured in APRICOT FILE proper.

As for more Workshops and Reviews, both are being arranged. It helps us however if you can nominate specific subjects in both areas for us to cover.

For the future

We already have some ideas from the SoapBox returns into which we can get our editorial teeth.

- **More technical articles**: or, to put it another way, "more info on the quirks and foibles of the machine". For example, "details of how the hardware can be accessed by the software to produce non-transportable programs — timer controls, sound generator, screen memory hardware". Another respondent suggests a piece on persuading the CPU that it is reading single- and not double-sided disks, and a workshop on how to add system tracks irrespective of BIOS version. Converting CBM Pet files to Apricot format came up twice, and another one to be requested a couple of times was more information about the .KB files. We're looking for authors on all of these topics.

- **More on assembler** was a popular request. In the next few issues we'll be writing about MS-DOS function calls and the difference between EXE and COM files. We're also putting together a DEBUG tutorial.

- **More for the beginner**: or, as one respondent put it, "more stuff for morons". Another writer: "more often than not the technical tips are a bit above me". There definitely seems to be a need for rather more of an elementary approach in some areas, and at the risk of offending some of our more advanced subscribers we intend to feature a section for beginners in APRICOT FILE.

- **More on MS-DOS** — we'll be carrying more on DOS commands in Simon Potter's occasional *Understanding MS-DOS* series. There'll be more patches for DOS utilities too, and we'll have a résumé of DOS error messages in next month's newsletter.

- **More on comms** was another popular request — especially more down-to-earth stuff for the unsophisticated. To some extent this is the bailiwick of our newly-launched sister publication COMMS FILE, but the editors do talk to each other ... A Kermit workshop was mentioned by a couple of people, another respondent asked for "cookbooks on comms, bulletin boards, modems, etc". Expect more comms coverage from APRICOT FILE.

- **More Workshops on specific packages** — dBASE and SuperCalc were mentioned in particular, and we'll also be commissioning articles on CardBox and DataEase. Any other packages we should be looking at?

- **More for the programmer** with requests for a Turbo Pascal tutorial, Turbo workshops, and "Basic hints and tips". Since we tend to be mainstream users of ready-to-go packages and MS-DOS, we don't have much programming knowledge in the office: so anyone who's interested in writing such articles is invited to call the Editor.

- **Vertical-market group tests** — a number of people wanted reviews and/or workshops in particular areas, with statistics, project scheduling, and company financial analysis being the favourites. To do this kind of thing

properly would consume several pages of the newsletter and we're not sure about the appeal: opinions, please.

- **Copy protection, security, and backup** questions would obviously interest many subscribers. Articles are on their way ...

Apricot antagonism?

One comment that puzzled us slightly was "you must shake off the old 16 BIT COMPUTING anti-Apricot bias". We don't feel particularly biased against Apricot plc, and we've always tried to be even-handed about the company.

As users, we like the computers in general ... but we dislike some of the particular quirks they exhibit, especially when the bugs and irritants show evidence of sloppy work in design and implementation.

As industry commentators, we can understand Apricot's move to larger and more powerful IBM-compatible systems; and we've even tipped Apricot shares if you want to risk a flutter on the side. (Incidentally, we recommended Apricot at 56p — as we go to press the shares are 104p).

As long-time Apricot watchers, we have been impressed with some things (the adaptation of many of the best aspects of the Sirius, the pretty full specification of the PC and Xi, the physical design of the computers) and depressed by others — the blizzard of BIOS variants, the operation of the promised 'five years' guaranteed support' when Apricot dropped the 8086 machines, the lack of direct support for the in-the-box packages like SuperWriter, SuperPlanner (remember that?) and Concurrent CP/M. ◆

Respond!

We urgently need your input. It's your newsletter, after all; we can try to match its content and style to what you want to read. To encourage rather more response, particular in terms of attracting specific subjects for us to cover, your SoapBox forms can now be returned to our FREEPOST address (no stamp required in the UK). Please use this facility — it helps us to help you.

The SoapBox — Release 2.8

Tell us what you do and don't like about this issue. Just complete the analysis panel and add any notes — what you'd like to see in the future, whether you want more or less of a particular author, or any remarks you'd care to make; then tear off the form or copy it, and send it to our FREEPOST address.

APRICOT FILE, TP Group, FREEPOST, PO Box 509, London N1 1BR

Article	< Interesting				Uninteresting >	Comments
Columns:						
Doodles by David	☐	☐	☐	☐	☐	
Bandage and Plaster	☐	☐	☐	☐	☐	
Workshops and Technotes:						
All about RS-232-— pt 2	☐	☐	☐	☐	☐	
Piping and Redirection	☐	☐	☐	☐	☐	
Reviews						
SignWriter	☐	☐	☐	☐	☐	
Cachebox	☐	☐	☐	☐	☐	
Ansible WordPerfect	☐	☐	☐	☐	☐	
Features						
News of Apricot	☐	☐	☐	☐	☐	
Comms World	☐	☐	☐	☐	☐	
Regulars:						
Diary Dates	☐	☐	☐	☐	☐	
Backpage News	☐	☐	☐	☐	☐	

Review

The Writing on the Sign

A look at Signwriter

By Terence Green

Signwriter is a program for producing signs and notices on a dot matrix printer when you don't want to go to the trouble or the expense of having the work typeset and printed.

It comes from Wight Scientific on a couple of 3.5in disks, with a manual and a couple of sheets for 'getting started'. Within a pretty short time — minutes, rather than hours or days — you should be able to produce simple notices using the program's built-in fonts, and with a little more application you can design and print fonts of your own.

You can then specify character height and position, calculate print positions and output your own design to an Epson-compatible printer. The program offers lots of scope for flexibility.

On the blocks

Once you've got the package out of the box and copied the files from the master disk on to your system, just follow the instruction sheets on 'Getting Started with Signwriter' and 'Signwriter for the Apricot Computer' — and within minutes you should be able to print out a sign that's perfectly adequate for a window display, pasting on the loo door at church socials, or sending anonymously through the post to upset people you don't like (the people at Signwriter only suggest the first option).

That's my theory, anyway. As with everything in life, practice makes perfect. It was after a few dry runs that I began to produce reasonable lettering, certainly suitable for, say, headlining leaflets you intend to photocopy.

The characters produced rely on an 80x80 matrix (due to be upgraded to 160x160) which produces passable clarity in circle arcs over 10 millimetres in height — the bigger the letters the clearer the arc outlines are. It isn't typesetting clarity; if you intend printing your leaflet there's really no need for this program. But compared to lining up dry-transfer lettering like Letraset, it's a doddle.

So, once I had convinced myself that the program justified its subtitle of "instant display lettering", I went back to the beginning and started afresh. The 'Getting Started' sheet refers you to the README file on Disk 1, and actually suggests plunging in from there without reading the manual at all. This struck me as good advice since hands-on experience beats manual-reading any day.

First, though, there was *some* information on the Apricot sheet which needed to be looked at. Predictably the sheet deals with aspects of the program which are peculiar to the Apricot, as opposed to the IBM for which the literature was produced.

- There are the predictable keyboard differences
- You won't be able to run the on-screen character designer (DESIGN.EXE) without BIOS 3.1, because you need to have BIOS 3.1 Extended Graphics loaded.
- If you *haven't* got BIOS 3.1, there is some helpful advice on getting Apricot plc to update you. (Anyhow, there is an alternative way of creating fonts if you are desperate to do so; that we'll go into later.)

So, with the sheets read and inwardly digested I turned to the README file on disk. This contains some information about advanced character design and printer settings.

README also contains Supplementary information from Wight Scientific "as Signwriter is constantly upgraded in small steps" and includes notes about "known faults and weaknesses". This up-front attitude appears in all the literature, and phone calls to the programmers are positively welcomed — very refreshing.

Up and running

This done I prepared a system diskette containing DESIGN.EXE and SIGNIN.EXE from the second Signwriter disk, along with SIGNOUT, FONTCALC and the ancillary files.

SIGNIN is for entering lines which are to be printed, including characters and shapes not on the keyboard which are present in the supplied font, or in a font you create. There are prompts for Horizontal or Vertical printing (obviously necessary if you want to print a row of very large characters on 80-column printers), and for spacing and justification or centering.

You choose the character size (measured in millimetres) and where the sign is to be placed on the paper. Signwriter calculates character spacing and likely line length, and beeps for action if your sign won't fit. You get the option to edit it, or change character size or spacing.

There is a limit of 70 characters to a line; over 50 characters, each input produces a warning beep to discourage you. The output program finds the amount of calculation involved in long lines too much! This can be worked around using sequential vertical prints.

Advanced features are available. For instance, you can read in a text file line by line. This works well until you come across a blank line which Signwriter takes as the 'end of input' signal.

Once a sign is complete it can be printed immediately; or it could be stored to disk, either in the default file SIGN or under another name.

Choosing to print will lead straight into SIGNOUT which can also be invoked direct from the A> prompt, in which case it will print out whatever is contained in the SIGN file. There are hidden options in SIGNOUT which are detailed in the manual. Why they should be hidden is unclear because they offer useful alternatives to straightforward output. There is also a 'quick print' facility which creates the sign with one pass instead of the normal three: this is handy for trial runs while you finalise your layout.

Options and possibilities

SIGN files can be printed to disk and run off later. In this case all the calculations which are normally done during printing will be done during the print-to-disk. This will result in quicker printing when you finally get round to it, because the calculations will be done already; but printing to disk can take some time if the file is very large, and stop your creative flow!

There is an option to reduce the sign size by a chosen ratio, and another to repeat the sign a given number of times. The cut-down option is a useful adjunct when working on layout, but beware the repetition, which will be continuous, with only a line feed between repeats.

You can also produce T-shirt iron-on designs using a Mirror switch in SIGNOUT which reverses the output for printing on to special paper. All good stuff.

A change of character

If you want to design your own characters, the DESIGN program provides you with an on-screen facility that's menu-driven and relatively straightforward. It helps to rough out a character on paper first; that isn't strictly necessary, though, for the instructions in the manual are sufficient to enable you to go straight into DESIGN and get along with the help of its menus. The main problem occurs with defining circle arcs, but practice makes almost perfect!

In order to create characters, both on screen and in print, you have to create an outline which can be *filled*. To get outlines it is necessary to create enclosed structures with inner and outer boundaries, so the more complex your designs, the less likely you are to succeed. For practical examples look at the outline characters which Signwriter provides in its SYMBOLS font.

You need BIOS 3.1 for this, but even without it you *can* still create fonts. Designing without DESIGN has to be done on paper (which takes time, but once again experience will speed matters) and entered using a word processor: or you can use the program FONTFIX.BAS supplied. This is described in the manual as being "relatively crude" and not as crashproof or user-friendly as other Signwriter programs. Take the authors at their word again; they are not overstating matters — FONTFIX is relatively useless and definitely *not* crashproof. But you do have the Basic source code to experiment with.

Following DESIGN, a program called FONTCALC must be invoked which will automatically check your changes and produce a new SIGNFONT and SPACINGS file. This allows SIGNOUT to work relatively quickly while retaining flexibility in the files containing details of your sign and the font from which characters are drawn.

All your sign output and font files (not the .EXE) files can be word-processed to edit them once you understand the principles explained in the manual.

The printer settings file can likewise be treated to suit your printer/application, although most matrix printers these days are Epson-compatible and will need no alteration to the SETTINGS file (my Kaga Taxan worked without any difficulty). Signwriter offers help with printer settings, and promises specific settings files in future for alternatives to Epson printers.

Documentation

As mentioned, the producers of Signwriter are very frank about the quality of the program, to the extent of advising you what to do if your update contains a disastrous bug.

The documentation constantly encourages you to contact the programmers — with suggestions, queries, problems, new characters you have designed. There are even offers of royalty payments for new fonts if they succeed in selling separately.

The README file on disk, which contains later additions to the photocopied manual, says this is a relatively early version. Buying Signwriter entitles you to one free update (succeeding updates cost £5), and as the program is "constantly upgraded in small steps" you should choose your time carefully. If you do take up the offer of constant friendliness with the programmers you will know when it's a good time, but on the other hand five quid isn't going to break your bank if you find yourself needing further, unfree updates.

Last words

I feel ambivalent about this program. I can see how it could be very useful to someone who had a specific purpose in mind, in which case it's worth costing Signwriter against other methods after making sure that it would suit your purpose.

But when the producers are falling over backwards to make it plain they haven't finished it yet, £80 plus VAT to join a user group devoted to one program suite — albeit a potentially very useful program — seems a bit steep. ◆

Signwriter

Description:	Instant display lettering by computer
Supplier:	Wight Scientific
Address:	44 Roan Street London SE10 9JT
Telephone:	01-858 2699
Price:	£80 plus VAT (10-day money-back guarantee). Includes one free update and free telephone support; each subsequent update is £5
Configuration	All Apricots except IBM-compatible Xen-i
Conclusion:	What the program does it does well (did I write that?). The manual is full of background information; but is confusingly presented, and badly needs a rewrite. If it cost half as much it would be a good buy. As it stands, Signwriter needs retouching.

RS-232 on the Apricot

Part Two — Does handshaking really make you go blind?

By Stuart Bunt

Last month I described how the RS-232 standard was arrived at and the minimum configuration which might permit data transfer between your computer and the peripheral:

Minimum configuration

	DTE [eg Apricot]			DCE [eg modem]
TXD	2	transmit data -------	receive data	2
RXD	3	receive data -------	transmit data	3
COMMON	7	----------		7

This is a minimal configuration for data communications which may work. But it doesn't allow for any conversation between the apparatus — to explain, for instance, that a printer's buffer is full. As most computers can send data far faster than a printer can print it this may lead to lost data if the computer merrily goes on sending data at 9600 baud while your tired old Epson chugs on at 40 characters per second.

The most commonly used of the other pins are numbers 20 and 6. These lines usually indicate that power is on at the DTE and DCE ends, via pin 20 and pin 6 respectively.

Line 20 is called DTR or 'Data Terminal Ready' on the DTE end, and it is usually indicated by the presence of a high voltage on line 20 — typically anything greater than 3 volts. When the DCE detects this high voltage on its pin 20, it will do its stuff.

Line 6 works in a similar way but it is the DCE which pulls the line high (meaning it puts a high voltage on the line to indicate it is ready to go). This is sometimes called the DSR signal or Data Set Ready (a 'data set' is an old-style Americanism for 'modem').

Pins 4 and 5 perform similar functions but indicate that the DTE wants to send information (RTS, Request To Send) and that the DCE is ready to receive it (CTS or Clear To Send). This is useful for indicating that the buffer of a printer is temporarily full, for instance; the printer (assuming it is configured as a DCE device) would pull the CTS line low, and this prevents the DTE from sending data — even though the DTE may be showing on line 4 that it wishes to send data.

One step up

	DTE [eg Apricot]			DCE [eg modem]
TXD	2	transmit data -------	receive data	2
RXD	3	receive data -------	transmit data	3
RTS	4	ready to send ----------		4
CTS	5	----------	ready to receive	5
DSR	6	ready? ----------	ready!	6
COMMON	7	----------		7
DTR	20	ready! ----------	ready?	20

One last foible: line 8 is sometimes used to listen for a 'carrier' signal on the DCE and is known as the DCD or Data Carrier Detect. This usually applies to modems, which have to wait for a carrier tone from the other end of the line before they can start communicating. But some other devices may need this line to be high before they'll work: in that case, a high voltage is usually forced on to line 8 by shorting it (cross-wiring) to another line like DTR which always has voltage.

Problem: DCE to DTE

It may be necessary to connect a DCE device (such as an IBM PC) to a printer (which is usually configured as a DTE). This would seem to be straightforward: a DTE to a DCE — isn't that what the RS-232 interface was designed for?

But in this connection all the data flow will go *from* the DCE *to* the DTE. In the RS-232 standard there is no way that the DTE device — the printer, in this case — can be quizzed by the DCE (the computer) about whether it is ready to receive data.

So to make most printers work, the hardware manufacturers are forced to improvise. Once the RS-232 standard has been abandoned there is little incentive in keeping the other pin assignments either. A pin has to be assigned for handshaking and often the problem of interfacing boils down to trying to find out which one has been hijacked by the manufacturer.

It would seem to make sense to use the Request To Send line (RTS, line 4) as the DTE printer is requesting data. An alternative is to use Data Terminal Ready (DTR, line 20); but some manufacturers avoid all clashes with the standard by picking a pin which is not defined at all in the classic RS-232 definition. If line 4 or 20 is used as the handshaking line there may be no need for line 5 (CTS, clear to send from the DCE). And of course if the printer does not send any data there is no need for the Received Data line on the computer (line 3).

All this can result in some rather odd cable configurations — in this hypothetical case, only lines 2, 7 and 4 are connected (to 2, 7 and 4 at the other end).

This is all well and good: but what if both your devices are configured as DTEs? What if one has

no input to pin 6 while the other is expecting something on line 6 ... and so on?

Never fear, there are some nice characteristics of the RS-232 standard which will help us here. If two devices are RS-232-compatible, they can be connected without starting a meltdown. (That's the theory, anyhow ... as we noted last month, if you short line 9 on an F series Apricot you'll find that pretty blue smoke curls out of the back of the machine as a small resistor in the line gives up the ghost!) Anyhow, in basic RS-232 the voltage levels will all be standard for on and off.

The DB-25 connector is also pretty well standard (this is the usual flattened-oval D-shaped connector with 25 pins or pinholes arranged in two rows). It should therefore at least be possible to push the connectors together with the occasional use of a gender bender (and I do not mean a smart gynæcologist with a nice line in trips to Morocco — a gender changer is an adapter that lets you connect a male plug to a male socket).

What's important is that you give both ends of the connection all the signals they require to work while trying to maintain as much of the reciprocal handshaking as possible (so that, for example, the DTE doesn't go on sending information to a switched-off DCE).

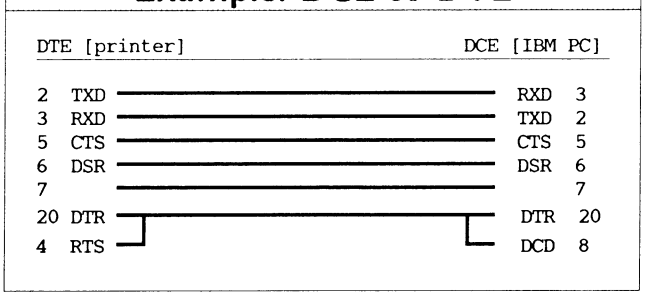
This is best described by example. Say you have a printer to connect to an IBM microcomputer. The IBM serial interface is configured as a DCE while many printers are configured as DTE (remember the DCE/DTE convention was designed to cope with modems and dumb terminals, it never said anything about printers). Now let's look at the most important lines, 2 to 8 and 20.

As the two ends are a DCE/DTE pair, you can connect lines 2, 3 and 7 to each other. This *may* work — but only if there is no handshaking at all. If the DTE has a standard RS-232 interface it may be expecting line 8 to be high; so what if the micro does not put out any such signal on line 8 (after all, it's not a modem)?

The answer is to trick the printer by connecting its own pin 8 to pin 20 which is always high as long as the printer is switched on. This sort of incestuous shenanigins is ok in theory, as the RS-232 should be immune to any voltages on any of its own pins.

What about line 4? The printer never sends any data to the micro, so it would never pull high line 4 (RTS, request to send); and yet the micro may not send any data to the printer *until* this line is high. Again, the solution is to short pin 4 to line 20 which is in turn pulled high by the DTE printer. Lines 5 and 6 can be connected to each other and the system should now run.

Example: DCE to DTE



Experimentation extras

All this connecting of odd wires one to another is made a lot easier if you invest in a multi-purpose cable. A cable like this has switches on each end next to the DB-25 connector which enable most of the common connections or fixes to be tried without you needing to touch a soldering iron.

The Serial Master Cable from **DNCS** is a good modestly-priced example (call **0706 67567** for DNCS in the North and Midlands, **01-368 1221** in the South). At £23.50, it's not a lot more than a normal custom-made RS-232 cable; and with it you can always impress your friends by making their serial interfaces work.

There are other alternatives such as a 'patchbox', a gismo which plugs into the line and has small wires which can be inserted into various holes in the connector to make any set of connections you wish. This is very fiddly, however, and a patchbox like this from DNCS costs £19.50; so why not buy the cable instead? That way, at least you have a cable as well ...

Apricot serial interface

Pin no.	Name	Function
1	GND	Frame ground
2	TXD	Transmit Data
3	RXD	Receive Data
4	RTS	Request To Send
5	CTS	Clear To Send
6	DSR	Data Set Ready
7		0V [Signal Ground]
8	DCD	Data Carrier Detect
9		+12V [via FS4 — warning: do NOT short out on F series machines]
10	...	[not connected]
11		-12V [via FS5 — note: pins 9 and 11 are NOT connected on PC/Xi]
12	...	[not connected]
13	...	[not connected]
14	...	[not connected]
15		External Transmit Clock [input]
16	...	[not connected]
17		External Receive Clock [input — note: NOT connected on PC/Xi)
18	...	[not connected]
19	...	[not connected]
20	DTR	Data Terminal Ready
21	...	[not connected]
22	...	[not connected]
23	...	[not connected]
24		Transmit Clock [output]
25	...	[not connected]

If you can't discover all the pin assignments on your computer or peripheral, or you don't believe

what the dealer or manufacturer is telling you, you can make or buy a line tester. This has small LEDs which light up when there is voltage on a line; with that you can tell which lines are used by each piece of apparatus and when.

You can pay hundreds of pounds for a really sophisticated line tester, but there are less expensive alternatives — the cheapest I have seen is the Data Tracker, again from DNCS, at £19.95. This has indicators for lines 2 to 6, 8, 11, 19 and 20; you won't need any more for most of the applications you're likely to come across.

A simple tester can be built yourself by connecting a low-current LED to the common ground on pin 7 and connecting the other leg of the LED to the line to be tested. As these are diodes they will only light when the voltage is applied to the correct leg of the LED: in this way it can show if the voltage is positive or negative.

If you want to be really flash (flash, geddit?) you could connect two LEDs in parallel but reversed, perhaps one red and one green; then one or the other will light to indicate is positive or negative voltage. These diodes can be purchased in packets of 10 for about £3.50 for low-current LEDs or 78p for high-current LEDs; the former should cause less interference to data transmission. Put them in series with a 470 ohm resistor to limit the current flow through the diodes.

Armed with your circuit tester and multiple-switched cable, you may go on to find that your printer/terminal connection requires only three or four wires to work. Usually these will be TXD, RXD, Common ground, and a handshaking line or two. Obviously this is going to make your cable quite cheap as you will only need four or five wires. Finding out which are the handshaking lines and when they are active is the hardest part; but once the connection is running the various lines can be shorted out until the machine stops, thereby identifying the handshake line.

Software handshaking

When all this hardware handshaking is complete there is still the little problem of software compatibility.

Thus far, all the descriptions have been about hardware control and handshaking at the level of voltages on individual wires. Many programs however also include *software* handshaking, where the actual content of the data stream itself (usually on line 2 or 3) controls the peripheral and the interchange of data between the two machines.

Typically, for example, a word processing program will send out a CONTROL-M code at the end of each line. This will tell the printer to start a new line; the printer may then send another control code back to the computer telling it that it has printed the line and is waiting for the next line to be sent.

This is the only method of controlling the data stream between remote terminals and modems — only a few lines may be connected between a modem and the remote mainframe, so all the control has to be in the data stream.

Because communication over such a long distance (and particularly over British Telecom lines) is prone to error, error-checking is often built into the data stream as well. At the simplest this consists of a parity checking bit: all the bits which make up one data block are automatically summed, and a 'parity bit' that is sent along with the block is set to 1 if the total is odd, 0 if the total is even. If the parity does not match at the other end, the terminal asks for the byte to be issued again.

The parity setting can be odd, even or none, depending on whether the software adds one bit when the total bits in the word are even or odd or if it doesn't bother to check for parity at all. (Most bulletin boards don't bother with parity: I suppose it saves on transmission time having to send one less bit with each byte. Besides, it also means equipment which can't check parity at all may still call up the bulletin board).

In its most secure and complicated form, error-checking can consist of 'packets' of data being interspersed with checking bits which allow the software to check the integrity of the data packet. The packet might also be sent back to the originating machine which can then check it against the original it transmitted; if the data returned doesn't match what was sent, the machine will send the byte again. The well-known public domain communication program Kermit works in this way (*it's on* DEALS FOR READERS *disk number AP021, price £10 — Ed*).

Setting up

The most important software parameters to set are baud rate, parity, number of stop bits and number of bits per word. **Baud rate** is the transmission speed, equivalent to 'bits per second'; for example, 300 baud means 300 bits per second (which, when extra bits for stop, start and parity are included, approximates to 300/11 or roughly 27 ASCII characters a second). The **stop bits**, which can be one, two or none, mark the end of a block of data transmitted. The **length of the word** transmitted is usually 7 or 8 bits.

All of these must be identical for the two devices. Hopefully the required setting for these can all be found from the manual of the peripheral, from DIP switches set inside the machine, or at worst, from the manufacturer's technical support service.

Most computers come with some software to set up these parameters — for instance, WCONFIG.EXE in a machine with Microsoft Windows: or SETUP.EXE in one with GEM as a front end. The variations are so enormous I am afraid I cannot

give much more help; as a last resort, try consulting the manual.

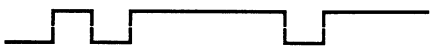
The parameters at the other end may be fixed at one setting, so that you have to change the microcomputer to match. Often you will get some choice, with the parameters generally set by DIP switches inconveniently located deep in the bowels of the peripheral. Getting at these frequently requires removal of the whole casing before they can even be examined, let alone changed.

Once you have gone to the trouble of finding out how these switches are set and have used the manual to find out what they mean, write it all down. Better still, stick a label on the side of the printer with this information so that if even you have to go through this again you will not have to open the casing again.

If you aren't sure what is required by a given machine or piece of software, start by trying 300 baud, no parity, one stop bit and an 8 bit word. Even if you get garbage at first, changing the various parameters one by one should eventually lead to the correct conformation.

A typical serial transmission

In this example the letter B (ASCII 42 Hex, 66 decimal or 01000010 binary) is sent with one stop bit, even parity, 8 bits per word and the binary 1s positive. The order is start bit (always positive), the eight data bits (01000010), one parity bit (set to 0 because the sum of data bits is even), and one stop bit. So the full data block for this one character is 10100001000:



1 0 1 0 0 0 0 1 0 0 0

The start bit shows when the data starts so that the initial 0 can be picked up. The eight data bits follow (with the least significant bit first). Even parity is specified and the parity bit that comes next is therefore 0, so that the total bits add up to an even number, in this case 2. One stop bit (which always has the opposite sign to the start bit) then indicates the end of the transmission.

Note that 1 is indicated by a positive voltage; this is called 'positive logic', but 'negative true logic' can be found in some systems. In that case the start bit would be a 0.

If all this fails, consult your dealer — who will then tell you he doesn't know how to connect RS-232 interfaces either ... Good interfacing! ◆

My bible for RS-232 interfacing problems is *The RS-232 Solution* by Joe Campbell, published by Sybex. This can be obtained in the UK from a variety of specialist computer bookshops — try Computer Books of 74 High Street, Rickmansworth, Herts on 0923 779129, who have it at £12.95.

Diary Dates

◆◆◆ MAY 1987 ◆◆◆

12 Desktop Publishing for Managers *London.* One-day conference/seminar for £199. Ordinarily we wouldn't feature this kind of event, but the line-up of speakers looks good — our regular contributor **Kathy Lang**, the author of the package we use (NEWSWRITER) **Friedman Wagner-Dobler**, and the excellent market analyst **Andy Tribute**. Speakers too from Aldus, CCA, Apple and Apricot — all very PageMaker-oriented. *Organiser: EMAP, tel 01-608 1161*

11-17 Flanders Technology International *Ghent.* Catch-all robotics-to-biotechnology show including computers and comms. First three days are for 'science and industry professionals' only, the last three are open to all. *Organiser: Flanders Expo, tel [Belgium] 91.22.60.31*

12-14 Infobase *Frankfurt.* New show promoting on-line databases and information service. *Organiser: Frankfurt Messe, tel [West Germany] 069 7575 477*

24-26 Automan *Birmingham.* Fourth running of this show for automated manufacturing. *Organiser: Cahners, tel 01-891 5051*

20-22 European Unix User Show *London.* Another chance to see what all the fuss is about, though last year's offering was a bit specialised and distinctly aimed at larger users. *Organiser: EMAP Exhibitions, tel 01-608 1161*

20-23 Microcomputer *Frankfurt.* Exhibition for office micros and small business computing generally. *Organiser: Frankfurt Messe, tel [West Germany] 069 7575 477*

27-29 Computer North *Manchester.* All-purpose business computer show at the impressive G-Mex centre. *Organiser: Cahners, tel 061-832 4242*

◆◆◆ JUNE 1987 ◆◆◆

1-4 Comdex/Spring *Atlanta.* The smaller (but still enormous) and infinitely more palatable of the two annual US Comdex shows. Las Vegas in the Autumn tends to get more product launches, but as a barometer of the American computer business this too is unmissable. *Organiser: Interface Group, tel (USA) 617 449-6600*

2-5 Office 87 *London.* Features office furniture and supplies but includes 'office machines' too. *Organiser: BED Exhibitions, tel 01-647 1001*

16-18 Comdex/Europe *Nice.* The European version of the big US Comdex shows is now in its fifth running. Last year it was dire, but the South of France in June was pleasant ... *Organiser: Interface Group, tel (USA) 617 449-6600*

16-18 Networks 87 *London.* Quite an interesting event last year, though geared to large user of networking products and service. *Organiser: Online, tel 01-868 4466*

Doodles by David

Tips and Tricks

from the working day

By David St.John Wallis

Having reached my thirteenth Doodle without the sack (*yet! Ed*) I actually forgot to write it this month. So if this lot passes the Editorial D Notice it'll probably be shorter than usual or it'll test NEWSWRITER's copy-fitting algorithms. Well, it is jolly late and the fifth is now a tenth. (Americans, or watchers of The Lost Weekend, will understand this Salingeresque remark ... or not).

The great date debate

My miniscule contribution to Lent business: while thousands of programmers are flexing their stirrups on this one, an arguably subtler manoeuvre is to RENname your AUTOEXEC.BAT as 1.BAT; it's not too much effort to press nine keys at the start of a session. *Now* you know why Dennis decided to call this column *Tricks and Tips*.

Of speed

Punters throughout the land who purchased Apricots which trot along at a stately 4.67MHz have been heard to mutter darkly that all this megahertz stuff is a ploy, if not a con.

The same chaps tend to write me huge dismissives about databases too. But here's one reason for speed: quicker spreadsheeting.

Lotus, the company which failed to make 1-2-3 version 2 available to Brits with Apricots, did so for Continental buyers. Oh, the disgrace. We tested our totally fenced copy using an elementary model described in a recentish edition of a glossy collection of advertisements published monthly (it shall remain nameless, but there are chimpanzees on the cover of most issues). This benchmark checked the speed with which the program sussed a sum like A1+B1*C1/D1, replicated down the spreadsheet as far as it would go.

1-2-3 version 1a replicated further, probably because it's not such a *big* program as version 2 (and that's probably why it did take longer to work out the result).

Here are the speed test numbers (for my own benchmark, incidentally, not *PCW*'s).

MS-DOS version	2.11	3.1	3.1
Lotus 1-2-3 version	1a	1a	2
Speed in seconds	169.1	124.1	62.8
File size	M1102	M980	M838
Number of cells	14326	12740	10894
Time per calculation	0.011803	0.009740	0.005764

Depressing the calculate key will demonstrate that the time per calculation of Lotus 2 is nearly half that of 1a; and overall speed is nearly three times quicker.

But to revel in these presents you have to be using MS-DOS 3.1, the Softclone IBM emulator, 800400eg as your memory resident graphic jockey, an 8087 co-processor, *and* a Teutonic keyboard layout .... which (for instance) substitutes the 'Y' for Yes as 'Z' for Ja.

The whole point about speed in spreadsheets is that if you've got a lot of iterations in a spreadsheet, you're either a very bad modeller or a jolly good one.

But if you have speed, you can get back to more serious things in life — like, well, grilling your tax return or smiling at clouds. Explanations of all these concepts begin on page 436 of my seminal tome on the subject, published by Color Climax GmbH — a bargain at only 80 Deutchmarks in the third kiosk on the right leaving St Pauli towards Klosterwall.

Power of the press

I put a free ad in APRICOT FILE last month for a modem: it sold on the very first day. For the asking price. That Lord Unilever, God bless his mittened pinkies, must have got it wrong. Got something to sell? List it in these hallowed pages. Better yet, they'll advertise it for months after you've sold it. (*It's your own fault for not telling us — Ed.*)

Let's try this for size. One Carron Capri 4 two-oven cooker with a griddle yet: a song at only £220. (We just got an Aga for the new MusePlatz.)

Dat ol' duplicate file problem

Subtitled 'Impossible no' with a few upsidedown question marks well beyond the capability of NEWSWRITER. (*Wrong, actually — Ed ¿¿¿*)

Take a look at this directory listing:

```
Volume ID for drive D: is Aprifile
Directory of  D:\

APFILE10 V1      6656   8/02/87  13:49
APFILE11 V1      6144  14/02/87  11:26
APFILE12 V1      2688  24/03/87  15:16
APFILE13 V1      6528  28/04/87   2:36
APFILE13 V1      6528  28/04/87   2:36 [!]
```

My friend Sandy says the one with the sexy square brackets can't happen (my friend Jill, to whom I am married, says "Pump out the cellar again you lazy sod" or words to that effect). Explanations on the back of a pound note gets you an evaluation ScannerS disk which you can even pay for even if you read this far. Or not.

Tip

Wow! At last a *tip* — from a column entitled *Tips and Tricks*. Will these wonders never cease?

As the regular reader will have observed, I am concerned with speed. This is because computers have let me do things more rapidly than I could do them before. The Apricot, however, is not speedy by present-day standards ... but you can make it go faster.

One: you can use FAST from DEALS FOR READERS. This makes things seem faster. You can also use QuiK from OmrC — gosh, what a difference! (*Or, for that matter,* HARE *from Ansible — Ed*).

Two: you can also load a memory-resident program called MicroCache. While this costs a princely £195 — actually far too much for a utility — who wants utilities when what we need is applications?

Three: you can fit an NEC V30 microprocessor replacement if you don't mind a few squawks from the loudspeaker. Or you could even try an 8087 numeric co-processor; I have a feeling that this makes lots of things go faster — irrespective of whether or not they are using the right registers etc.

FORMAT foibles

For a chap who after seven years' practice can now do the Rubik cube on one side in only three months, you may feel that anything I have to say about computers is of less interest than ambergris or the price of passion in Singapore.

But I must advise you not to use your TOOLS utility to format a hard disk. — you will for certain onanise your label sectors, an exercise guaranteed not to to make front-page headlines in *Forum*. Apply in a plain brown wrapper with a fully-addressed postage stamp and we might, I say *might*, send you FORMAT.EXE or even the wonderful 'factory formatters' (FORMAT *is also on* DEALS FOR READERS *disk AP020 — Ed*)

Of Apricot ...

You would think that a company with a desktop publishing system to sell would be jumping about when a punter came along with a need for an 80MB server and five terminals. Not so, Apricot plc.

That just about sums it up doesn't it? Public Liability Company, I mean. Here is a tiny dealer trying to earn a crust and supporting Apricots without even the merest flicker of recognition — over four years, for God's sake. And support from Apricot Public Liability Company came there none ... ◆

Write for us!

We're always interested in articles and other submissions for publication — particularly product reviews, 'how-to' articles, technical tips and amplifications on specific packages.

If you have something substantial you'd like to say, contact the Editor first to talk about it. In general, we like to see at least a synopsis before committing ourselves to a biggish piece.

Articles can be submitted in two ways. You can use **electronic mail**; we're on Micromail, Telecom Gold, The Source, and Mercury Link (formerly EasyLink). If you submit something on e-mail, phone us to say there's something on the way: use a line width between 60 and 80 characters: and append a note if you've had to change any non-transmittable control characters and the like.

Or send the article on **disk** — we can take Apricot, Victor/Sirius, IBM PC and PC/AT disks. And we can handle common spreadsheet and most WP formats, though we do prefer a minimum of formatting (eg avoid proportional spacing) and it's safest to use any ASCII-type or 'non-system' format that your word processor permits.

Electronic mail and disk transfers can become garbled, so if you can manage it we do prefer an accompanying hardcopy too.

We edit everything we receive — for grammar, for our own 'House Style', and for the format requirements of APRICOT FILE. We try not to change the sense or the conclusions, however, and we do attempt to preserve an author's own writing style. In most cases we send authors a pre-publication draft for checking and approval before we go to print.

Material for publication should be accompanied where appropriate by a self-addressed and correctly stamped envelope or mailer. We'll do our best, but otherwise we cannot be responsible for the return of unsolicited manuscripts, disks, and suchlike.

Unless otherwise agreed, editorial submissions will be accepted on an 'all rights' basis so that the copyright on all contents of APRICOT FILE is held by the Publishers or by the Publishers and Authors jointly. On the other hand, that's just to protect you and us: if you want to retain copyright, just say so.

The Editor is **Dennis Jarrett**, and you can reach him on **01-833 3501** or via our e-mail addresses — Telecom Gold **81:JET767** (from overseas **10081:JET767**), Mercury Link **19011170**, or The Source **BDE566**. Our postal address is:

TP GROUP
PO Box 509
London N1 1YL
England

Someone out there must have something against us. Remember the mailshot to **'Dennis Ferrett'**? Here's the pre-registration label **Roger Gann** received for the **PC User Show**:

Mr R J Chann
Director
T P GROUP
P O Box 509
London
N1

WordPerfect advances ...

Software distributor **Softsel** produces a weekly list of its best-selling products, and while the Hot List for the UK doesn't quite have the industry-barometer status of the US version it certainly represents a decent indicator of sales success. As regular readers will know, several of our contributors are mightily impressed by **WordPerfect** on the Apricot. That view on the word processor from **Sentinel Software** appears to be shared: it has topped the Softsel Hot List for 15 consecutive weeks.

... Apricot slips ...

The absence of a mass market for **Apricot** products and the end of the cut-price stock clearance saw the company's share of the UK market slip to **6.2** per cent in February, according to market researchers **Romtec**. **Olivetti** displaces Apricot for third place behind **Amstrad** (28.3 per cent) and **IBM** (23.9 per cent).

Romtec has also produced a report on distributors, which gave **Northamber** top slot in 1986 with sales to dealers of **£21.7 million**. **P&P** (which began as Pete and Pam, two South London teachers who took an empty suitcase to the States and bought all the Apple software they could find for resale here) was ranked second with £18 million, ahead of £13 million for **MBS** — which, as it happens, has just reported **profits of £937,000** on a **turnover of £76 million** for the year to December 1986 (it lost £3.4 million on sales of £67 million a year ago).

Romtec says the prospects for dealers are excellent, citing expectations of growth better than 50 per cent during 1987; the dealers they service anticipate only 22 per cent increase in sales.

... Brikat stumbles

After 18 months of rapid growth, Kettering-based **Brikat** has reported a halving of profits for the half-year to January. By comparison with the 1986 midpoint, the USM-quoted firm had profits of **£240,000** on sales 11 per cent higher at **£6.5 million**. From the financial base of its highly successful **Pegasus** accounts packages, Brikat started expanding into retail outlets during 1985, and last year bought out the troubled Poole database designer **SoSoft** — the latter was responsible for the **Tomorrow's Office** file manager, and Brikat repackaged that as a Pegasus product. But now SoSoft has been shut down, all but three of the retail outlets have been closed or sold, and the fledgling subsidiaries in South Africa and the States have been bought by local interests. Chief executive **Johnnie Johnson**, the former BOS Software director who replaced long-time MD **Colin Stanley** in December, reckons the drain on profits has now been arrested.

Southdata consulting

Superfile is a flexible and well-liked database manager with some novel features, and it probably deserves a higher profile than it gets. But as is the case with many sophisticated packages, it can take some effort to get the most from it. With some **3,000 sales** in the past five years, Superfile's progenitor **Southdata** is taking a sensible step towards increasing its appeal — a consultancy service for Superfile implementation that can include turnkey systems supply. Southdata's unusually literate MD **Peter Laurie** quotes Kutuzov from *War and Peace*: "Train hard, fight easy".

New trade body for micros

Some 65 people from all parts of the small-computer business turned up at the inaugural meeting of the **British Micro Federation** late in March. The BMF aims to produce a code of practice, reliable statistics, a PR exercise, promotion for the FAST anti-piracy programme, and liaison with Government policy-makers. The committee elected by the meeting illustrates what we see as the uncomfortably wide range of this body — it includes representatives from **Microsoft**, **Atari**, games-producer **Mastertronic**, business software distributor **Softsel**, and leisure computing distributor **Lightning**. Past attempts to find a commonality of interests for the micro business have all foundered; it will be interesting to watch the progress of this one ...

Epson goes laser

Epson has put a name and a price on its long-expected laser printer. The **GQ-3500** has a pretty aggressive RRP of **£1,795**; that buys HP LaserJet Plus emulation as well as compatibility with Epson's matrix printers.

At **six pages per minute**, it isn't as fast as some of the competitors. But it looks very neat, and it has some excellent features — it includes automatic face-down **collation** (many more expensive laser printers still output face-up, so that a long document comes out with pages in the reverse order), it can use **envelopes** and comp slips (also quite rare), it can print **'landscape'** (lengthways on, rather than across, the A4 sheet), extra **fonts** come on a credit-card-size plug-in cartridge rather than the bulky HP/Canon-style package. What's more, that price buys you a true **graphics** capability for digitised images and the like (the 'Plus' bit of the LaserJet Plus) — though the 640KB buffer provided as standard probably won't be enough memory for that, and you'll pay extra to take the GQ-3500 up to 1.5MB.

Still, "Epson's answer to the mass-market generic laser" looks a good buy ... Dealers should have it from this month. ◆

BT charges ahead

According to a recent survey by **National Utility Services**, Britain has Europe's highest charges for local telephone calls (and higher than North American rates as well). On the other hand, Britain also showed up as having the *lowest* charges for trunk calls. Since BT's rates were approved by the telecomms watchdog Oftel, we're stuck with them. So cut out those order-by-phone pizzas and enjoy your next long-distance call ...