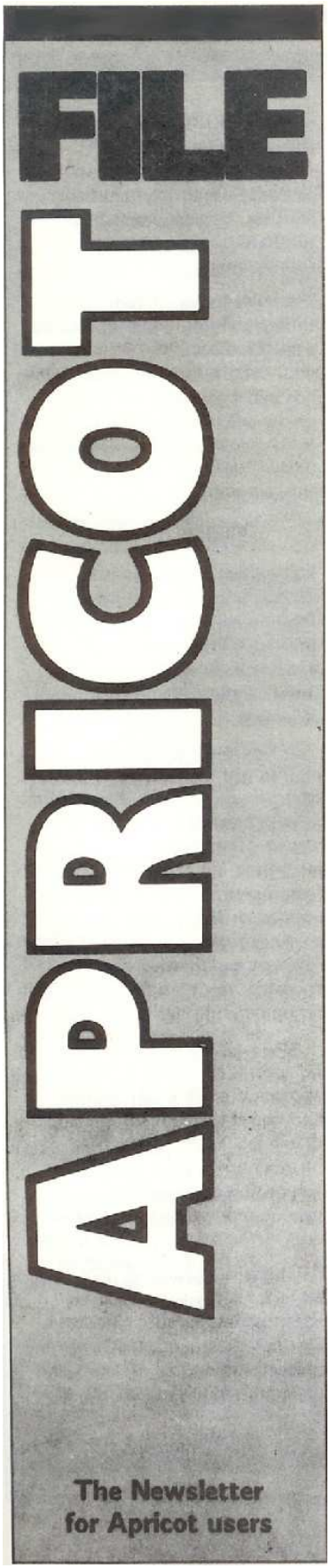
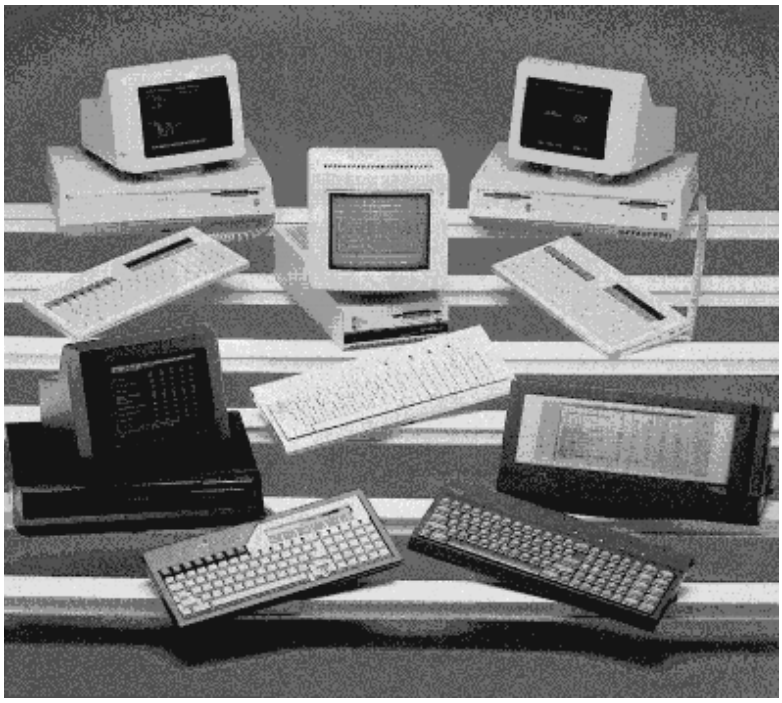


Apricot File 2_12



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 2.12

September 1987

FILE

APRICOT

The Newsletter for Apricot users

Update your Apricot!

Apricot's PC-286 conversion: the Chameleon that changed its spots

By Garret Keogh

Convert your Apricot PC or Xi into an IBM AT-compatible? Yes, a mere eight months after APRICOT FILE told you about the 'Apricot by another name', Apricot's project Chameleon is finally in the company's price list for shipment "immediately".

We may have correctly judged the time it would take to deliver, but we didn't get it quite right (well, we can't all be perfect). With a hard disk and Hercules-compatible monochrome video as standard and networking facilities as optional, the PC-286 is a much more impressive machine than we predicted.

For those who don't know, the PC-286 is an upgrade for users of the old-style Apricot PC and Xi that makes it IBM-compatible — and performance-wise it's a superior kind of compatibility, for you get a 286 processor and other internals that make your Apricot the equivalent of an AT.

The upgrade comes from Apricot itself. Basically your entire PC or Xi (system unit, keyboard, screen) is despatched to Apricot Computer Maintenance. There its insides are ripped out, the motherboard is replaced, things are added to and removed from your keyboard, they mess around with your monitor as well ... and for an all-inclusive price between £700 and £1,000 you end up with an AT in your Apricot PC casing.

What Apricot gains is your continuing custom for its now almost complete IBM-compatibility. The original launch of the PC-286 upgrade used phrases like "ongoing development of the user base"; the reality seems to be that it wants to hang on to those Apricot-family users, but only on Apricot's own terms.

That's not to say you would be getting a bad deal. Users are moving to the industry standard all the time; more important, so are the software houses. Here is a practical and not uneconomical way to make that move. Just looking at the PC-286 you can see what a powerful and impressive machine it is.

First appearances

The real changes in the old-style Apricot take place inside the box. There are substantial modifications however to both the monitor and the keyboard.

Physically, the keyboard is the same — same size, same number of keys. The difference is that a total of 40 keytops are replaced, along with their meanings. The most striking difference is the eight numbered function keys instead of the Apricot-labelled function keys.

CONTINUED ON PAGE 2

In this issue

The Disinformation Column — On Strange Communications . 6
The Workshop: WordPerfect address list builder 9
Batch File Recipes — Disk calatogue, File finder 9
Comms on the Apricot — Part Two: Communiqué reviewed . 14
Footnote: A Partial History of BIOS Bugs 19
DEBUG for Beginners — Part One: First Steps 20
The Q&A Clinic — three pages of queries and answers . . . 27
Doodles by David — the occasionally essential column 30
Writers in this issue 31
The Workshop: SuperWriter's bugs and bodges 32
SoapBox 35

Apricot's PC-286 upgrade

CONTINUED FROM PAGE ONE

Two extra function keys have also been added above the number pad — IBM-compatible software tends to use the ten function keys on IBM-compatible keyboards, so Apricot needed a couple more.

Basically the righthand side of the keyboard changes totally. The keypad is completely moved in one keytop space; the Apricot's ESC key is displaced by an ALT key, the new ESC key is situated above the number pad, and IBM-type PgUp and PgDown keys have appeared.

Those familiar with an AT keyboard anyway will have difficulty with the crowding on the newly modified keyboard. Regular users of the old-style Apricots will probably have problems with the new position of the ESC key and some of the additional keys. The Apricot keyboard has always been a good one, however, with a nice feel and a sensible layout; this should make the crossover that bit easier.

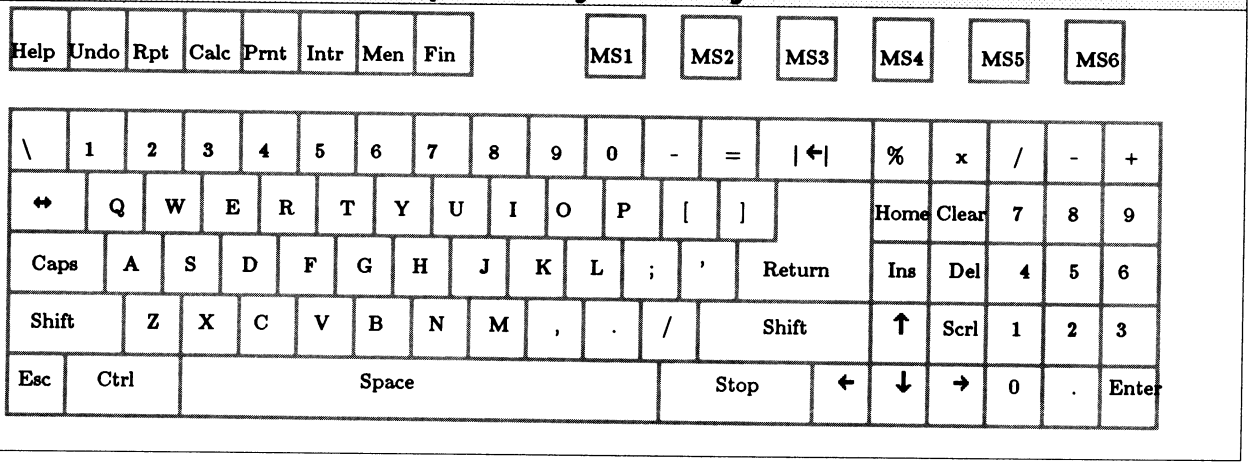
An optional keyboard in Xen-i style is available, and there's a full-scale ATE layout on offer too. But without AT-type keyboard sockets, the actual process of connecting this seems difficult to comprehend unless you stick sort of adaptor thingy on to the back. The Xen-i MicroScreen keyboard adds an extra £179, the 101-key ATE keyboard (no MicroScreen) is £99.

Unless you pay extra for a change of display, the 'new' monitor is the same 9in display with the same controls. Screen size is the only place you lose out, if at all. The PC-286 comes with a screen of 720x348 resolution, compared with 800x400 on the original Apricot; that means you get a border of unusable blank space around the edge of the screen.

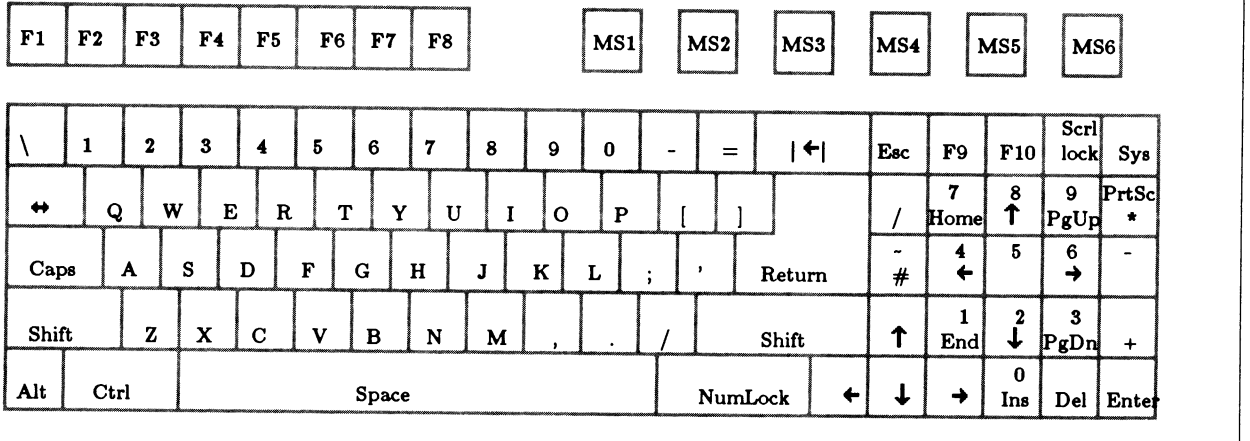
This is not as off-putting as it might sound and is only really noticeable on first viewing. You do gain Hercules-compatible monochrome video as standard due to the presence of the Paradise Systems chip on the motherboard — that emulates different screen and graphic types. PC users who are used to the non-graphics on their machines will be quite pleased with this step.

Optional 12in monitors are available at extra cost — monochrome green-on-black, Apricot's 'paper white' display, and hi-res EGA-compatible colour.

The Apricot keyboard layout ...



... and the PC-286 keyboard



The motherboard

This is where the real changes take place. The latest surface-mount technology gives you the Intel 80286, several custom chips and eight one-megabit DRAM memory chips in an impressively small space — so the board comes with a megabyte of memory there, and there are sockets for a small daughter-board that will take another 1MB.

It is only with this surface-mount technology that so much can be fitted inside the relatively small size of the PC casing. Surface-mounting is the kind of toy that only the big boys can play with — it prevents cloning and copying, but more important it also means the motherboards can be churned off the production line quickly. It's expensive for the manufacturer, but the results can be worth it.

The shoe-horning means you do lose the Apricot's internal expansion slots; but the external XP-i box is reportedly available at last — that will hold up to four full-size IBM cards and retails at £199.

Due to the presence of the 286, support is also possible for the OS/2 operating system announced with IBM's PS/2. This could be important in terms of expansion and/or upgrade to the possible future industry-standard; OS/2 will provide true multi-tasking at long last, it will retain MS-DOS compatibility, it will need a lot of memory (hence the provision of so much on-board RAM and an easy-fit daughter board), and Apricot should be able to ship it early next year.

A LAN controller chip is also available on the motherboard. This is a feature that is standard on the Xen-i range — which now uses a virtually identical system board — but it's only an option on the PC-286.

At the back of the CPU all the I/O sockets stay the same: same connections, same positions. Mercifully, the serial and parallel ports on the motherboard use PC/XI-style connectors rather than the IBM AT's nine-pin DIN-type socket.

Retaining the Apricot's setup of serial and parallel ports makes the changeover much more easy as you don't have to mess around buying new cables and gender-benders to make your system work; this removes one of the doubts we aired in our January issue, Release 2.4.

The only significant change is for the monitor socket which changes from a male to female connector. When your old monitor is used with the modifications by Apricot, a rather dodgy-looking adaptor is added to run the display.

Software

The PC-286 comes with MS-DOS 3.2, Utilities and GW-BASIC. The BIOS version is 3.06.

An optional extra is Windows (version 1.03), the latest implementation with full use of the function keys. This works quite well on the PC-286 making good use of both the computational speed and the new keyboard — it also utilises the existing MicroScreen rather well.

For those worried about having to go and spend a lot of money on new IBM-compatible software Apricot has a scheme that might soften the financial blow — for certain packages, you can upgrade your current software to the latest IBM-compatible version for a set fee. For example, to upgrade your Apricot version of WordPerfect to IBM-type WordPerfect 4.2 costs £80.

Compatibility

The PC-286 uses 3.5in disks. You can reformat Apricot-type disks as 1.2MB IBM-compatible; but you can also read your existing Apricot disks regardless of how they were formatted or whether they're single- or double-sided. This is a surprising feat and again wipes out the worries we aired in our original report in the January issue of APRICOT FILE.

So all your valuable data can be read and saved in IBM format. For business users this is a mighty plus.

As for IBM-compatibility, the PC-286 looks as close as you can get. When I tried it all the major application programs and utilities ran perfectly on it first time with no exceptions — we tried the likes of WordStar 4, SideKick, HomeBase, Reflex, ProComm, Norton Utilities and our DTP package NEWSWRITER. Basically the PC-286 is a Xen-i in a small box, and is as compatible as its big brother.

True, the small screen size takes some getting used when you're used to seeing popular programs like Lotus and SideKick running on a normal IBM-compatible type screen; but this will obviously cause no problems for consistent PC users.

Speed comparisons

	IBM PC/XT	Apricot PC/Xi	Apricot Xen-i	IBM AT	Apricot PC-286
Computing index	1.0	1.1	9.8	7.7	9.7
Disk index	1.0	na	2.5	3.3	1.7
Performance index	1.0	na	7.3	6.2	7.0

[The SI utility won't run on the Apricot PC/Xi to test the hard disk.]

When I ran the Norton Utilities SI System Information program, I originally got processor speeds that were considerably lower than those from the Xen-i. It turned out the PC-286 had come from the factory set for 8MHz rather than

10MHz; a carefully-located DIP switch on the motherboard brought it up to scratch. The speed can also be altered by software via an interesting suffix to the MODE command.

The table below compares the PC-286 with its most immediate competitors. Most of the deficiency in overall speed by comparison with other 80286-based machines is due to Apricot's use of the old-style disk drives.

Conclusions

It's interesting to look at the relationship between the PC-286 and the Apricot Xen models. All the low-end Xen machines now have almost exactly the same motherboard as the 286; it appears that the fact the motherboard could fit in the existing PC casing was a fortuitous coincidence — hence the upgrade offer.

There *are* differences between the motherboards that are worth noting, such as the fact that on the Xen machines the LAN chip is standard while it's only an option on the PC-286. The PC-286 board is, surprisingly, also slightly larger with the connector for the hard and floppy disk drives being on the motherboard itself.

The possible drawbacks we mentioned in our January issue, concerning the reading of Apricot-format disks and the alteration of I/O sockets, have been corrected. Does someone in authority at Apricot actually *listen* to what we say?

The PC-286 upgrade produces a fast and elegant machine. It's interesting to see that the review edition of the machine I was given was a PC that had definitely been through the wars; yet as a PC-286 it worked perfectly well. This is good tidings for those whose Apricot's have seen better days — at least you know get a fast and reliable machine.

While £900 might seem cheap for a 1MB hard-disk AT-compatible, it is a large sum for a step that might not be the right one — and a step you might not need to take anyhow. But if you want to get back into the mainstream — if you want the latest software (which does generally mean the best), if you want to be part of the world's largest user base (with all the clout that implies), and if you want to get on to a decent upgrade path with guarantees of future compatibility (such as Apricot is offering), the PC-286 is an investment worth thinking about.

In short — if you have decided you really would benefit from going IBM-compatible (and it is a major decision), this is the way to do it. You must remember however that for that price you are not only gaining an AT, you're *losing* your existing machine. If you think this is a worthwhile cost to pay, you won't be disappointed with the PC-286. ◆

PC-286 upgrade prices

FROM:	TO:
Apricot PC twin 315KB floppy drives	PC-286 20MB hard disk **£999.00**
Apricot PC twin 728KB floppy drive	PC-286 20MB hard disk **£899.00**
Apricot Xi 5MB hard disk, one 315KB floppy drive	PC-286 5MB hard disk **£849.00**
Apricot Xi 5MB hard disk, one 315KB floppy drive	PC-286 20MB hard disk **£1,099.00**
Apricot Xi 5MB hard disk, one 728KB floppy drive	PC-286 5MB hard disk **£699.00**
Apricot Xi 5MB hard disk, one 728KB floppy drive	PC-286 20MB hard disk **£999.00**
Apricot Xi 10MB hard disk, one 315KB floppy drive	PC-286 10MB hard disk **£849.00**
Apricot Xi 10MB hard disk, one 315KB floppy drive	PC-286 20MB hard disk **£1,099.00**
Apricot Xi 10MB hard disk, one 728KB floppy drive	PC-286 10MB hard disk **£699.00**
Apricot Xi 10MB hard disk, one 728KB floppy drive	PC-286 20MB hard disk **£999.00**
Apricot Xi 20MB hard disk, one 728KB floppy drive	PC-286 20MB hard disk **£699.00**

[All PC-286 models come with MS-DOS 3.2, 1MB memory and one 720k double-sided floppy drive.]

Stop me and buy one

We are currently negotiating the terms on which we'll be able to offer the PC-286 upgrade ourselves through the DEALS FOR READERS service. By the time you read this, we should have things firmed up — details in our next issue, hopefully.

Optional extras

MONITORS	
12in monochrome green 720x350	£199.00
12in monochrome paper white 720x350	£349.00
12in EGA colour 640x350, with EGA card (requires XP-i box)	£1,098.00
KEYBOARDS	
Xen-i MicroScreen keyboard	£179.00
Xen-i 101 key ATE keyboard	£99.00
MISC	
1MB memory	£399.00
LAN chip	£99.00
XP-i expansion box	£199.00
External IBM-compatible 5.25in floppy drive	£299.00
Windows with Apricot Mouse	£199.00

PC-286

Description: Apricot PC or Xi upgrade. Provides replacement motherboard with 10MHz 80286 processor and 1MB memory; internal 3.5in hard disk drive of 5, 10 or 20MB plus single IBM-compatible 720Kb 3.5in floppy disk drive; Hercules-compatible monochrome video on motherboard with replacement monitor in same housing; replacement 90-key keyboard with all AT keys and functions in same housing; MS-DOS 3.2.

Supplier: Apricot Computers plc via dealers

Price: From £699 to £999, plus optional extras — see table

Conclusion: An AT-compatible at around £900 is still as good a bargain as it was in January; there's still no sign of a sub-£1,000 AT-compatible with a hard disk on the market. If you're going to make the move to IBM-compatibility, we think the PC-286 offers a practical solution to what is usually a complex and confusing move — compatibility on an already familiar machine is a good solution.

It's that time again ...

This bumper issue marks the end of the second year of publication for APRICOT FILE. Next month will see the start of Release 3 — and that too will be a substantial number: the DEBUG series continues, a Turbo Pascal primer gets under way, we have some good Workshops in the pipeline, and subscribers will get the Complete Index to Release 2 (and this time we won't we cutting back on editorial to cope with it).

We're still the world's only independent publication outside the user groups which is dedicated to the 'original' Apricots, and we're quite proud of our longevity.

To check that we're delivering the goods, though, and to make sure that we continue to do so, we need your feedback. At the back of this issue you'll find a *SoapBox* form on which you can rate the newsletter's performance over the past year; please take a moment to express your opinion — it helps us to help you, as they say, and the FREEPOST licence means it won't even cost you the price of a stamp.

Of course, the other thing we need is your re-subscription. The majority of our subscribers have taken the newsletter from the start of the publication year, so for most of you this will be the last issue you receive — unless you let us have your renewal instructions.

The 1987-88 rates for new APRICOT FILE subscriptions will be rising by ten per cent. But for existing subscribers, we're holding the present rates:

BRITAIN	£50.00
IRELAND	£60.00
REST OF EUROPE	£60.00
ELSEWHERE (air mail)	£65.00

If you want to renew your subscription, just drop us a line with your name (or your company's) and if possible a note of your Subscriber Reference Number — that appears on the envelope in which this issue arrived.

There are three ways to pay:

- cheques, Eurocheques, and money or postal orders *in Sterling* made payable to 'TP GROUP'.
- Girobank transfer — our National Girobank account number is **59 028 1909**.
- Credit card — we accept **Access/Mastercharge, American Express** and **Diners Club**. Sadly we do not take Barclaycard, Visa or Connect.

Sometimes it feels as though the rest of the world has forgotten about the Apricot user. With APRICOT FILE, you are not alone. Can you afford *not* to renew?

The Disinformation Column

On strange communications

By David Langford

Honestly, I don't go out looking for cruelly ironic stories about famous computer firms. It was one of our customers who wrote in after the nasty morning when he discovered his local dealer was about to go bankrupt with his Xi monitor stranded in their stockroom. The Liquidator sternly informed him that he could reclaim it only by making a personal appearance before 12 noon, ie within about two minutes. Now read on:

"Having outrun the taxi, I arrived at the door just in time to step aside to allow two removal men past. They were carrying a large fibreglass thingy on which was inscribed: 'AWARD. APRICOT DEALER OF THE YEAR'. That said it all."

Speaking of our favourite firm, you probably won't have heard about the obscure new bug reported in Apricot MS-DOS 3.2. This operating system may work pretty well with most conventional software, but it causes Apricot repairmen's brains to crash! We have a harrowing report from an Xi owner who received a personal visit from one of these roving Apricot chaps. At sight of the fatal words 'MS-DOS 3.2' on the boot-up screen, it seems that your typical repairman froths at the mouth and snaps, "You're not authorised to have that operating system on this machine!" Then he reformats the hard disk with MS-DOS 2.11, BIOS 3.1.

Ansible Information is wondering whether it's worth circumventing this rare bug by patching MS-DOS 3.2 so that it camouflages itself as MS-DOS 2.11. Alternatively, bolt the door and be ready to defend your computer to the death if Apricot maintenance people are reported in your vicinity.

Most surreal of all are the conversations several people have recently been having with famous Sorcim, alias famous Computer Associates ...

"Hello, Computer Associates?"

"Yes, can I help you?"

"I've seen this advertisement for the vastly improved SuperWriter 2, asking me to call this number ..."

"No. No, you've called the wrong number. That number was put in by mistake. You want to get in touch with Sorcim, not us."

"But you *are* Sorcim."

(Long pause) "Er, yes, that's, um, true, and of course we'll be glad to help you, but it's nothing to do with us. The advertisement was a mistake."

"Look, I've been using SuperWriter on an Apricot for years and I'm very interested in this program — I want to know more."

(Triumphantly:) "Ah, of course it's only available for the IBM PC."

"Yes, it's my new IBM that I want it for. Could I have some details?"

"I'm sorry, this program is nothing to do with us. That isn't really our number in the advertisement. You want to talk to Computer Associates."

"But you *are* Computer Associates."

(Long pause.) "Er, well, ye-e-es, but ..."

And so on, forever.

With a strange device

Every so often I hear of someone coming to grief through forgetting about the ways of MS-DOS device names. Newcomers are usually surprised by the neat way in which 'devices' like the console and printer are accessible through DOS nicknames as though they were ordinary files. Experienced users will have picked up the trick of using COPY with device names for short cuts — for instance:

- COPY FILENAME PRN will print out a (plain ASCII) file without requiring you to run the word processor, but the file had better be short if your printer isn't loaded with continuous stationery
- COPY CON FILENAME allows you to type little files (like batch files or an amended CONFIG.SYS) straight to disk, ending input with CONTROL-Z followed by Return
- COPY CON PRN is handy for addressing envelopes: direct output from keyboard to printer, though nothing's printed until (again) you hit CONTROL-Z. This is where you find out whether the tab characters which so conveniently centre the address on the screen are properly interpreted by the printer. Note that COPY CON XXX will copy from the keyboard to XXX, whatever XXX may be, but COPY XXX CON will send stuff from XXX to the display.

Unfortunately these special device names like CON for console and PRN for printer don't just go away when you're not thinking about them. All sorts of disasters lie in wait when you accidentally use device names as filenames — they stay device names, even if you've added an extension.

So you've written your SuperWriter article about AIDS prevention, and wittily retitled it CON.DOM, and you activate the macro which saves the text and exits SuperWriter ... the result is that a messy version of what you've written is smeared over the

screen, while no file is created at all. This is in fact an accidental use of the technique suggested by Jost Riedel in the Release 2.11 *Readers' Writes* column: writing to CON as though it were a file, to send awkward characters like Escape to the screen despite GW-Basic's reluctance to do so directly.

Titling a file PRN (or LPT1 or LPT2, device names which allow the distinction between two printer ports) is liable to do much the same, except that the mess winds up at the printer.

The same dangers await with CLOCK, the built-in clock (inadvertently trying to read this by opening a SuperWriter file called CLOCK with any extension is likely to hang your Apricot, and TYPE CLOCK is deeply dangerous too), the communications devices AUX, COM1 and COM2, and the MicroScreen alias MSCREEN.

The MicroScreen is particularly weird in its behaviour. COPY FILENAME MSCREEN or COPY CON MSCREEN (end with CONTROL-Z) allow you to label it without using the escape sequences which 'officially' turn it on ... although the MicroScreen clock may overwrite part of your text unless first turned off. (If the characters sent to MSCREEN begin with ESC E, it will be cleared and the cursor homed in the MicroScreen without affecting the main screen.) It then becomes possible to give the MS-DOS command TYPE MSCREEN, which more often than not will list what you've written to the MicroScreen, over and over again, possibly forever. This also works on machines like the F2 and F10 which don't even have a MicroScreen. (I idly — though not very seriously — wondered whether one could save and restore MicroScreen labelling on the Xi, with batch file commands like COPY MSCREEN STORE.DOC and, later, COPY STORE.DOC MSCREEN. The results varied from interesting to disastrous.)

Of course the null device, NUL, is provided in MS-DOS as an intentional means of throwing away unwanted output. So inventing a filename like NUL.DOC for saving word-processed text really is asking for it. Nobody has ever found a use for NUL besides suppressing the odd unwanted screen message: COPY THISFILE THATFILE > NUL should omit the '1 file(s) copied' message, though naturally this fails if you're using a screen accelerator program which disables the right-arrow 'redirection' operator.

Memory-resident problems

Memory-resident utilities are all the rage, and fairly soon I'm going to have to sweat through a detailed tutorial column explaining how to write simple ones in assembler. (They're a bit too fiddly to offer as tiny COM files for creation with DEBUG.) But with these lurking demon programs come a variety of new perils; here are some points to watch.

- Is a given program memory-resident? Some programs may modify the Apricot's workings without actually leaving parts of themselves installed — for example, the interrupt repair utility I described in APRICOT FILE Release 2.5. To check whether FRED.COM is memory-resident, use DEBUG as described in my Release 2.7 column; see whether the starting segment of the available memory changes when you load FRED.
- Do you regularly install and remove several such programs? If they're well-designed they shouldn't interfere with each other's workings. But when they are installed 'on top of' each other it's wisest to remove them in reverse order of installation to ensure you get back all the tied-up memory. For example, after running FRED.COM (which usefully causes a loud censorious beep whenever you type a rude word) and then BLOGGS.COM (which usefully lets you switch a flashing cursor on and off from within your word processor), you decide you want to write some pretty rude things about Sentinel Software Ltd and therefore wish to 'de-install' FRED. Since BLOGGS.COM is installed 'on top of' FRED.COM, removing only FRED.COM won't make the space it used available. But if you remove BLOGGS, and then FRED, and then re-install BLOGGS, you'll maximise the usable memory.
- Following on from that, do you regularly use MANAGER, ACTIVITY or GEM to run other programs? If in any of these cases you load a memory-resident program, it's liable to slot itself in on top of MANAGER and (as it were) lock the lid in place. Depending on circumstances, this can lead to disaster. Those little memory-resident devils are most safely loaded either (a) before anything else, via AUTOEXEC.BAT; or (b) using batch files which load them just before the program that needs them and remove them immediately afterwards.

Escape Codes revisited

I've heard from readers who are having trouble with some of the Escape sequences from my recent demi-semi-definitive article. Most frequently, there appear to be bugs when using Turbo Pascal to invoke sequences like ESC n or ESC $ which load the keyboard buffer (with current cursor co-ordinates and the character at the current cursor position, respectively).

Here's an example of a non-working Turbo function sent from that part of Australia whose heart-stopping initials are A.C.T:

```
Function CurrentChar:Char;
begin
  Write(#27'$');
  Read(Kbd,CurrentChar)
end;
```

This has a nasty habit of finding nothing in the keyboard buffer. The problem is actually a peculiarity of Turbo Pascal itself, which does not properly set up a type-ahead buffer unless you've set the {$C-} compiler directive (that is documented as suppressing CONTROL-C breaks and incidentally speeding screen output).

I find it best to put {$C-} at the head of the Turbo program (after the PROGRAM name declaration and initial comments, but before any definitions for Type, Var, Const and so on). Once or twice it's apparently failed to work when put further down in the source code.

ESCTEST

```
Program ESCTEST;
{Read from keyboard & output straight to screen
to test ESC sequences}
{$C-}  {Force creation of type-ahead buffer}
Const
  Escape = ^[;              {Define ESC constant}
  ResetScreen = ^['z';      {And ditto for complete
                            screen reset}
  LastWasESC: Boolean = False;  {Flag used to
                                spot two ESCs in
                                succession}
Var
  Key: Char;                {This will contain the
                            current keystroke}
begin
  Write(ResetScreen);       {Start tidily}
  Repeat
  Read(Kbd,Key);            {Transfer from
                            keyboard...}
  Write(Key);               {...to screen.}
  If Key=Escape then begin      {Has someone just
                                hit ESC?}
    If LastWasESC then begin    {Has ESC been typed
                                twice running?}
    Write(ResetScreen);         {If so, tidy up...}
    Halt                        {...and finish.}
    end
    else LastWasESC:= True      {Otherwise, just
                                remember this ESC}
  end
  else
    LastWasESC:= False;         {Clear the flag if
                                the latest key
                                wasn't ESC}
  until False;              {go on forever: there's a
                            sloppy internal HALT
                            instruction}
end.                        {end program}
```

Escape tester

When researching the escape sequences article I used this simple Turbo Pascal program to check various effects. It merely reads exactly what you type at the keyboard and — unlike MS-DOS or a word processor — sends exactly those characters to the screen. It's handy for experimenting and getting the feel of how escape sequences control the screen.

So to check that ESC ? turns on all the MicroScreen lights, type ESC and a question mark. To turn on reverse video hit ESC followed by 'p' (and turn it off again with ESC q). To leave the program altogether, just press ESC twice in succession.

SCAVEN bug

Gateway User Group has not only reported but cured a quite important problem with SCAVEN.COM (from DEALS FOR READERS disk AP003). It seems that this utility correctly spots bad sectors on Apricot disks, but does not correctly mark them as bad (since it fails to allow for the peculiar Apricot arrangement of non-allocatable disk sectors used for the directory and file allocation tables). While you'll always be safe if you nervously throw away every disk on which bad sectors are reported, you cannot rely on any disk whose bad sectors have been 'mapped out' by SCAVEN. Duff sectors may still be roaming freely, just waiting to nobble your favourite files.

Mark Fisher's corrected version DISKCHK.COM sorts out the problems, and any regular user of SCAVEN.COM should get hold of a copy pronto. By the way, a double-sided disk which seems unusable owing to bad sectors in the required directory and FAT areas may work ok when formatted single-sided and, perhaps, cleaned up with DISKCHK.

(We'll be putting DISKCHK.COM *on to a future* DEALS FOR READERS *disk. In the meantime, anyone with a copy of AP003 is welcome to return the disk to us with a stamped self-addressed envelope for deletion of* SCAVEN *and insertion of the New Improved model — Ed.)*

A toast to longevity

Good grief, this is Release 2.12 — has APRICOT FILE really been going for two whole years? It seems like much longer *(cheeky sod — Ed)*. The old 'real' Apricots still seem to be holding up fairly well; but there's a slow trickle of discouraging news about worn-out or loudly rattling disk drives, disintegrating MicroScreens, fading monitors, clogged keyboards, once-proud users who've eagerly or reluctantly made the fatal leap to Imperial Babbage Machine compatibility. Year by year we grow fewer, clinging to our fond memories like Battle of Britain veterans: how many of the Few will still be around to celebrate this magazine's third anniversary in 1988, I wonder? Anyway; here's to us. ◆

WordPerfect Address List Builder

A free-style address automator

By Lindsay Doyle

Here's a suite of macros (XQT files to SuperWriter fans) to automate an address-list searcher and envelope-addresser — like the one for SuperWriter in APRICOT FILE Release 1.8 but much simpler, thanks to WordPerfect's advanced features.

I would never have abandoned SuperWriter and taken up WordPerfect if it hadn't been for the recommendations of my daughter (who writes science fiction), David Langford (who writes science fiction), and Christopher Priest (who writes science fiction).

The latter two are also responsible for Ansible Information, a nifty little firm about whom readers have heard much good. They have supported and extended SuperWriter for many years after it was abandoned by its parents. Now, advancing boldly into the 20th century, they also offer the best price for WordPerfect; and they supply programs for converting SuperWriter files to WordPerfect format and for editing macros, neither of which are available elsewhere as far as I know. Ansible's product support is nonpareil, a word which is in my *Oxford English Dictionary* but has not yet been installed in the WordPerfect dictionary.

Macros are to WordPerfect what XQT files are to SuperWriter, to wit files of command (and/or text) keystrokes which substitute for manual entry whenever an oft-repeated series of keystrokes is required. They have the file type .MAC.

WordPerfect provides for the generation of such files, but not for examining and editing them. This weakness is overcome by Ansible's macro editor, WM.COM, which is one of many goodies on the Ansible WordPerfect Disk reviewed in APRICOT FILE Release 1.11).

I use macros for (among other things) generating standard letterheads with date and salutation, writing "Yours sincerely, Lindsay Doyle" with two keystrokes, selecting an address from an address list and inserting it in the text of a letter, and addressing envelopes. This article describes the last two functions.

An identical structure can be used to retrieve 'boilerplate' paragraphs from a data file for making up contracts, dunning letters, product specifications, or any other documents where pieces of standard text are used repetitively.

If you have never dared use macros, here is your chance to try them out without fear of failure. If you use them every day, you may nevertheless find some useful hints in what is to follow. You may have different formatting requirements, but this should not prevent you from adapting my approach.

Requirements

This is the set of specifications I devised for addressing letters and envelopes:

1. The address list is to be free-style — meaning that each address should be free to contain as many lines and as many characters per line as required.
2. Each address is to be free to contain a telephone number and additional information (''Barby and Bingo'', ''use back door'', ''vicious dog'', ''cheques bounce'', and the like) which can be viewed without being printed.
3. No keywords are to be used — the system should be able to retrieve any address on any word contained within it.
4. Design a first macro to retrieve a selected address and place it under the letterhead or anywhere in a letter which has already been started.
5. Design a second macro to retrieve an address either from the address file or from the text of a letter, and to print it on an envelope at a programmed position. (Being able to retrieve from text means that when I have written a letter to someone who is not in my address file I don't have to retype the address on the envelope.)

The address list

The address list is simply a WordPerfect file containing your addresses. As I have several such lists, I call my correspondence address file C-AD.

Each address is entered line by line without blank lines. A blank line is entered to separate it from the optional telephone number and other data, and another blank line separates this from the next address.

Although the addresses need not be entered in any particular order as far as the macro is

concerned, it is convenient for manual editing to enter them in alphabetical order of last names.

The address list can be examined without using the macros, for instance to retrieve a telephone number, by using the WordPerfect search function ^+_*keyword*_^+.

Transcribing macros

In the text that follows I have used the symbols which appear in the Ansible macro editor WM.COM. These are not always exactly the same as used by WordPerfect (for technical reasons which are beyond your ability to understand so I won't try to explain). But then you'll never see them in Wordperfect except while you're writing the macro. But then you can't edit macros in WordPerfect, so you really need WM.COM. Right?

Basically, [Block] means push the <BLOCK> button, [Srch>] means search in a forward direction, and so forth. Where you see three dots in a macro followed by a new line of text, it means I wanted to continue on the next line so as to identify separate functions which would otherwise be run together. There is no Return in the macro at that point. So when entering these macros, don't type the dots and don't use the Return key except when the text says [RETURN].

The address search macro

I have given the address search macro a Control name, ALTA.MAC, which means that it can be called up by holding down the CONTROL key while pressing 'A', instead of having to press the macro key (REPEAT on the Apricot) and then spell out the name of the macro file. This feature of WordPerfect is not very well described in the manual, but is covered clearly and succinctly by Ansible.

ALTA.MAC has two subsidiary macros, NOTFOUND.MAC and FOUND.MAC. They tell it what to do in either of the two cases — that the desired string is not found or that it is found. Take my word for it, the lines which call them have to be positioned as shown. See the Ansible manual for a clear explanation and examples.

ALTA.MAC

[Macro] NOTFOUND [RETURN]	If string was not found, go to macro NOTFOUND.MAC to end in an orderly manner.
[Switch] [Retrieve] A:C-AD [RETURN]	Otherwise call up the screen for Doc 2 ... ... and load the address file C-AD from disk A.
[Slow]<0>	Make macro actions visible.
[Srch>] [Macro Pause]	Pause while user enters search string.
[Srch>] [Macro Pause]	Search for string. Pause while user accepts selection.
[Slow][Off]	Make macro actions invisible again.
[<Srch] [RETURN] [RETURN] [<Srch]	Search for the beginning of the address — ie two RETURNs next to each other, indicating a blank line.
[Block] [Srch>] [Srch>]	Search for two RETURNs in the other direction, the end of the address. You don't need to enter the RETURNs again.
[Move] 2 ...	Copy address block into temporary storage.
[Macro] FOUND [RETURN]	If you have reached this point, the string was found; so finish accordingly.

NOTFOUND.MAC

[Exit]	Leave Doc 2 ...
N ...	... without saving it.
N ...	Do not leave WordPerfect but clear Doc 2.
[Switch]	Return to Doc 1.

FOUND.MAC

[Exit] NN[Switch]	Leave Doc 2.
[Move] 5	Put selected address text in body of letter at cursor position.

The envelope macro

I have called the envelope-addressing macro NV.MAC, but you might want to give this one a Control name too. It makes use of the two-document feature and column formatting ability of WordPerfect. Column formatting comes into play because I want to position the addresses on my envelopes indented to the right rather than left-justified; this facility could be adapted for the previous macro as well, if desired.

NV.MAC

```
[Block]              Block the address (from cursor
[Srch>]              to blank line). Note: macro ^A
[RETURN]             left the cursor at the
[RETURN]             beginning of the address.

[Srch>]              Copy address from Doc 1
[Move]               into temporary storage.
2 ...

[Switch]             Switch to Doc 2 and clear it.
[Exit]               Too bad if you had anything
NN ...               there!

[Math/Cols]          Set two-column mode, second
4 [RETURN]           column starting at position 25
[RETURN]             (or your choice). Note that
2 [RETURN]           end of first column must also
                     be adjusted to be less.

[>]                  Cursor right.
20[<]                Cursor left.
[V]                  Cursor down.
25[Exit]

3[HardPg]            Turn on column feature and go
                     to column 2.

[Move]               Print the address at
5 ...                column 2, Doc 2.

[Macro Pause]        Wait for user to align
[RETURN]             envelope in printer and press
                     RETURN.

[Print]              Print the envelope.
1 ...

[Math/Cols]          Turn off columns feature.
3 ...

[Exit]               Clear Doc 2
NY                   and return to Doc 1.
```

Using the suite

I assume that you have generated an address file according to the simple rules described above and have successfully transcribed the four macros to wherever you want them to exist in your system — for myself, all my WordPerfect command files and macros are in a subdirectory on hard disk; all my correspondence files are written to floppy.

■■ Calling ^A

Start with a potential letter on the screen. Place the cursor where you want the address interpolated and do ^A. The letter will disappear temporarily and you will see the text of the first part of the address list.

The status line will display the message 'SRCH', prompting you to enter the string you wish to search on. Do so and press Return — don't use ^+, as you would ordinarily for a search, because in fact you're returning from a macro pause to the macro.

The cursor will curse to the first occurrence of the selected string and the beeper will beep. Note that the cursor doesn't have to be at the beginning of the address, as it may have found the desired string in the middle of a line (for instance, it might find 'Smith' in the line 'John Q. Smith').

If you are pleased with the selection, press Return; the text of the address will be entered in your letter, which will then reappear. If you don't want the first occurrence of 'Smith', press ^+ twice and the macro will step to the next occurrence (eg 'Mary Smith') and so on ('Xavier Smith', 'Yolanda Smith') until you press Return or until there are no more examples of the selected string. All this happens in a flash, is much simpler than it sounds, and soon becomes second nature.

■■ Calling NV

You should now have a text on the screen which includes the address followed by a blank line. (Actually it can be any paragraph followed by a blank line, if you can think up some use for printing it separately.) Place the cursor at the beginning of the paragraph and call the macro — press the REPEAT key, followed by 'N', 'V' and Return.

The macro will appear to do nothing, though it is invisibly carrying out all the busy little tasks we have assigned to it. It will beep, however, which is your signal to install the envelope in the printer and press Return again when ready. (You don't know how to get an envelope into your Epson printer? See the discussion below ...)

The macro now prints and disgorges the envelope, while the letter remains on screen for your delectation.

Epson printers and envelopes

The first job is to get the print head over to the left. If it's not there already, you may have to turn the printer off and on again to cause it to assume the position.

Put a curl in your envelope by pulling it over the edge of the table once or twice. Then pull the paper scale or bail forward and insert the envelope from the rear — under the continuous stationery, which does not have to be removed. Keep the envelope to the right and roll it round and up into view until about an inch protrudes above the scale.

Close the scale and then position the envelope to the left against the paper drive. Engage the release lever (push the far lever away from you) and set the head adjustment to max (pull the near lever toward you). These precautions will ensure that the envelope does not get smudged from inadvertent contact with the ribbon and that the 'out of paper' detector is satisfied that paper is present.

When printing the envelope is completed, reset the two levers. These actions, so tedious to describe, are very simple to carry out and soon become second nature.

Good (automated address) hunting! ◆

Batch File Recipes:

Disk catalogue, file finder

By David Gaunt

These two batch files, which I use frequently, illustrate the flexibility of the MS-DOS batch file system. One produces a catalogue of all files on a disk, the other helps locate a file from among a pile of floppy disks! Both make use of utilities obtained from the DEALS FOR READERS Public Domain collection, though CAT.BAT can be modified to work without them.

CAT.BAT

Directories are valuable for organising files on floppy disks as well as hard disks. But a drawback is that it's impossible to obtain a complete listing of all files on the disk without having to type in a complex command sequence.

CAT.BAT is a batch file which produces a catalogue of all files on a disk. The syntax is ...

CAT <drive:> <directory(s)>

If no directories are specified, CAT.BAT produces a list of files on the disk by reading the directory parameters from another batch file called CATDIR.BAT. Each disk has its own unique CATDIR.BAT file containing a list of directories located on the disk; when a new subdirectory is created, only the CATDIR.BAT has to be updated.

Lines 3 and 4 check that a drive is specified and that CATDIR.BAT exists; if not, appropriate error messages are displayed. Line 5 stores the specified drive in the environment (so this batch file may not work with MS-DOS 3.1).

CATDIR.BAT must have the following format:

%1 %2 \ \directory1 \directory2 ... etc

What's happening will be clear if you run CAT.BAT with ECHO ON. If you enter the command CAT A:, line 6 would look like this:

IF @==@ A:CATDIR CAT A:

In other words, when no directories are specified CATDIR is called and the two parameters %1 and %2 become CAT and A: respectively. CAT.BAT is then recalled from CATDIR.BAT but now has a drive parameter and several directory parameters. The loop in lines 8 to 14 will report the contents of each directory specified; if you don't have the public domain utility SDIR.COM (on DEALS FOR READERS disk AP002), line 11 can be replaced with something like ...

DIR %DRIVE%%1 /P

The usefulness of the batch file can be increased if line 11 is replaced by ...

SDIR %DRIVE%%1 /P > PRN

This will list the disk catalogue at the printer. If you use this with a hard disk, make sure your printer has plenty of paper!

As you can see, this batch file makes extensive use of chaining. An easy and very useful way to terminate a long batch file is to chain it to a batch file containing only a Return and a line feed (^z). For example, when the last directory has been listed on the screen the batch file chains to CRLF.BAT (line 9) and stops abruptly — much quicker than using a label at the end of the file.

CAT.BAT contains several escape codes to liven up the display (escape codes can be entered into batch files via EDLIN, by pressing ESC and then an uppercase 'R' followed by the appropriate code). Line 10 is particularly helpful — it persuades the status line to show which directory is being listed on the screen, a feature particularly useful when directories fill more than one screen.

CAT.BAT

Line numbers are included for reference — don't enter them when keying in this batch file.

```
1   ECHO OFF
2   ECHO ^[(^[(
3   IF %1@==@ GOTO LABEL1
4   IF NOT EXIST %1CATDIR.BAT GOTO LABEL2
5   SET DRIVE=%1
6   IF %2@==@ %1CATDIR %0 %1
7   SHIFT
8   :LOOP
9   IF %1@==@ CRLF
10  ECHO ^[j^[x1^[;^[)^[p DIRECTORY OF %DRIVE%%1
    ^[q^[(^[K^[y1^[k
11  SDIR %DRIVE%%1
12  ECHO
13  PAUSE
14  FOR %%@ IN ( CLS SHIFT IF) DO %%@ NOT %1@==@
    GOTO LOOP
15  CRLF
16  :LABEL1
17  ECHO ^G^[p THE SYNTAX IS CAT [DRIVE:]
    [DIRECTORY(S)] ^[q
18  ECHO ^G^[p DIRECTORY(S) ARE OPTIONAL ^[q
19  CRLF
20  :LABEL2
21  ECHO ^G^[p NO FILE OF DIRECTORIES
    (CATDIR.BAT) FOUND ON DRIVE %1 ^[q
```

SEEK.BAT

Recognise this problem? You know the file name, and you *know* it's in that pile of disks, but which disk ...? SEEK.BAT allows any number of disks to be searched for specified files. The syntax is ...

SEEK FILESPEC1 [FILESPEC2 ... FILESPEC3 ...]

Wild cards and multiple file searches are supported.

SEEK.BAT

Line numbers are included for reference — don't enter them when keying in this batch file.

```
1   ECHO OFF
2   ECHO ^[z^[(
3   IF %1@==@ GOTO LABEL1
4   A:
5   CD\
6   :SEEK
7   ECHO ^G^[p INSERT DISK TO BE SEEKED IN DRIVE
    B: ^[q
8   ECHO
9   PAUSE
10  ECHO ^[z^[(
11  ECHO ^[p THE FOLLOWING FILES WERE FOUND ON
    DISK IN DRIVE B:^[q
12  WHEREIS B:%1 %2 %3 %4 %5 %6 %7 %8 %9
13  ASK ^G^[p DO YOU WISH TO SEARCH ANOTHER DISK?
    ^[q
14  IF NOT ERRORLEVEL 1 GOTO SEEK
15  CRLF
16  :LABEL1
17  ECHO ^G^[p THE SYNTAX IS SEEK FILESPEC1 ..2
    ..3 ^[q
```

Line 2 clears the screen and line 3 checks for a filespec. If one has not been supplied, the batch file will go to LABEL1 (lines 16 and 17) and prompt for the valid syntax. Lines 4 and 5 set the disk to its root directory to ensure availability of the utilities used; SEEK.BAT requires CRLF.BAT and the DEALS FOR READERS utilities ASK.COM and WHEREIS.COM.

The loop in lines 6 to 14 prompts for the target disk to be placed in drive B, reports which of the specified files have been found (if any), and asks whether you want to search another disk. If yes, the batch file loops; otherwise it chains to CRLF.BAT and ends. ◆

Contribute!

Why not share your own useful utilities? Let us have your own home-brewed batch files, and we'll write them up for this *Recipe* series. Just us a line at APRICOT FILE, PO Box 509, London N1 1YL.

About the newsletter

Our aim is to provide a complete editorial and information service to users of Apricot computers.

Except in the normal course of day-to-day contact, sub sales and the like, we are not affiliated with or supported by Apricot plc or any of its dealers. Nor for that matter are we sponsored by or directly associated with any user group or any other such organisation, though in principle our goals and guiding intentions are very similar to the declared aims of many such bodies.

The Publisher does not necessarily agree with the views expressed by contributors, and assumes no responsibility for errors of reproduction or interpretation in the subject matter of this newsletter or from any results arising therefrom.

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]
Easylink	**19011170**
The Source	**BDE566**
Telex	**265871 MONREF G**
	quote reference 81:JET767

The People

Editor	**Dennis Jarrett**
Production	**Fiona Carey**
Write	**Garret Keogh**
Subscriptions	**Terrance Duggan**
DFR/Technical	**Roger Gann**

The Guarantee

The annual subscription to APRICOT FILE buys 12 monthly issues as follows:

Britain	£50.00
Rest of Europe	£60.00
Elsewhere (air mail)	£65.00

If having subscribed you come to feel that APRICOT FILE does not meet your requirements or that it does not match our claims, we will gladly cancel your commitment to the magazine and refund the unused portion of your subscription.

■■■ ISSN 0268-6066 ■■■

Checking out comms

Part Two of a series:

The Apricot Internal modem plus Communiqué

By Mike Florey

An internal modem should be cheaper than an external equivalent: there is no need for a separate box, and as the computer provides power there's no need for a power supply. Because there is no modem cable to an RS-232 port, a major source of headaches is removed; and overall the complete installation is much neater.

On the other hand, a spare expansion slot is needed; and resetting the modem involves rebooting the computer. But to reset an external modem after it has dialled the same number four times unsuccessfully may require it to be switched off and on again — is that more or less inconvenient than rebooting?

You pays your money ...

The Communiqué package

Apricot's Communiqué was launched in late 1984 as an internal modem and comms package for the PC and Xi machines.

The modem and software are available separately if required, but it's unlikely that many have been sold that way: the software will work only with the modem, and the incessant changes to the modem device driver have defeated most attempts by software houses to support the modem — so Communiqué is just about the *only* comms package that will work with it.

The software was originally copy-protected with emphasis placed on the impossibility of illicit access to expensive on-line services by anybody not in possession of an original system disk. The author of this bright idea unfortunately overlooked the ease with which passwords could be 'stolen' by simply TYPEing the file ACCESS.PAR; that contains all the legitimate owner's user IDs and passwords in ASCII format.

Sanity returned with the introduction of a non-protected generic version (version G2.0) following the launches of the F series and the Xens. The distribution disks contain files for all models and the correct selection is copied to the user's working disk by some clever batch files.

Both the modem and software have suffered a long and chequered history of design changes — the early releases should never have been inflicted on the market. The most up-to-date configuration has the following release numbers:

Modem firmware:	version V2.1
Modem driver:	MODEMAPR.SYS
	version Vf3.1.2
	dated 6 November 1986
Communiqué software:	version Vc2.20

Early production models of the modem (those built prior to Spring 1985) had a tendency to drop the line about six seconds after detecting a carrier tone that was weakish but perfectly usable. Upgrading to version V2.1 of the modem firmware cures this problem completely; but bear it in mind if you're buying secondhand.

Modem performance

In general, though, on a subjective assessment the modem's transmission performance leaves much to be desired: it still gives up too easily on calls which have a noise-free but not particularly strong modem tone. As you might expect, it's rather better at the slower and less demanding V21 (300/300bps) data rate.

Prestel access has proved trouble-free from my home (about two miles from the West Wickham telephone exchange) and from the office in central London.

I have no trouble calling any **PSS** access node at 1200/75bps from home. This has been less successful from the office since we moved and took our phone numbers with us (very convenient, but the extra cable length doesn't do much for performance). The office fax machine (cheap and well worn) does rather better over the same lines.

Calls from home to the Midland Bank home-banking computer (a **viewdata** service) on the opposite side of the London call area are marginal, with a high error rate in the 75bps direction and occasional complete failure to respond to modem tone.

The software

The absolute minimum hardware configuration you need for Communiqué is 256KB memory, RAM BIOS version 3.1, and a single-sided disk drive. The *practical* minimum demands twin double-sided drives; a hard disk gives a big improvement in the time taken to load program files.

Apart from a few reservations, the software is a much happier story than the modem. There is a lot of it and it is heavily overlaid to fit within the 256K memory constraints of early Apricots. It's written in a mixture of Assembler and compiled Basic, which achieves Apricot's aim of portability between machines in its range; the downside is that this severely compromises the speed at which data to or from the line can be handled.

My calculations based on downloading a largish ASCII file at 1200/75bps suggest that the software can manage about 500 bits per second flat out; this drops to about 350bps using EPAD error protection (EPAD is an extended error-checking protocol). On the Gateway Apricots, Communiqué is reputed to be even slower.

This is disappointing. According to Apricot, the 80286 and 80386 based machines (including the 'Apricot Architecture' Xen) can handle data at the full line speed — not much of a comfort to those of us with 8086s.

The whole thing is menu-driven and includes a facility to run other software from within Communiqué (only relevant for the built-in text editor unless you have a hard disk). Up to 40 services can be accommodated on the connection menu.

In everyday use Communiqué is simplicity itself: press '1' for the Connection Menu, key the number of the required service followed by Return, and sit back while the software sails through the complexities of dialling and logging on. The truly squeamish can arrange for the screen to blank during log-on, but then you could be left staring at an empty screen if something goes wrong.

Intelligent use is made of the MicroScreen; and, with a little practice, the on-line Help and simple procedures render the Handbook virtually redundant. Viewdata and Teletype modes are supported together with partial implementations of DEC VT52 and VT100 protocols. It is possible to switch between modes while on-line; the process is rather cumbersome, however, and it uses up space on the Connection Menu.

Control files permit automation of regularly-performed functions. But they are not interactive, and I've found them to be of limited use. Keep them short and simple for best results.

Files may be sent from disk or stored to disk ('dumped', in Communiqué parlance). Communiqué cannot understand MS-DOS, on the other hand, so subdirectory references are taboo. The screen can be echoed to a printer but this is a bad habit — it will usually lengthen connect time, painful when using expensive on-line services.

One sad omission is some form of clock to provide a reminder of how much a session is costing. A partial remedy is available in the TIM.COM clock program shipped on Ansible's ACOLYTE disk (available from DEALS FOR READERS at £57.50). This does have the drawback that it blots out part of the status line in viewdata mode; and ACOLYTE itself clashes in memory with part of Communiqué — perhaps it's the modem driver, or it may be a key assignment problem.

Setup

Setting up comms software is inclined to be a tricky business for the non-technical — and many dealers seems to be even less capable. At least Apricot has tried to make the process painless by providing a comprehensive selection of examples which can be adapted to your own use.

A parameters file called ACCESS.PAR contains information for each service listed on the Connection Menu: the telephone number, User ID and Password (both optional), sign-on, attachment and encryption file names, and some printer details. Sign-on files have a .SGN extension and contain the sequence of messages and responses required to log on to a service. Attachment files have a .ATC extension and contain details of data bits, parity, stop bits and the like; a number of the attachment files provided are actually duplicates under different file names.

Your greatest difficulty is likely to lie in creation of sign-on files for services not already catered for. In many cases it is possible to adapt one of the examples provided; but take a copy first using the MS-DOS COPY command — there is no facility to read in a file, edit, and then save under another name.

A sign-on file comprises a logical sequence of messages to be awaited from the distant end and the responses to be sent. The user ID and password stored in the parameters file for a given service may be inserted in the corresponding sign-on file by the use of [I] and [P] respectively.

It's all a bit too dogmatic for my liking: the ability to accept both upper and lower case as valid would save a lot of frustration. Screen presentation, though, is excellent — a new line for each message or response, and none of the weird control codes so prevalent in comms software.

Viewdata mode

Most viewdata services assume that the user has a colour screen. It can be frustrating to be told that the trains highlighted in red will not run — worse still if you have the conventional green screen ...

Seriously, if your main interest is viewdata the Apricot PC is the wrong piece of hardware to use.

The F series machines do support colour, but the software will run even more slowly as the CPU has more to do.

Note that the Prestel 'Welcome Page' changed earlier this year. As a result, the Prestel sign-on file in Communiqué needs to be changed:

```
[SNGLE]      set screen to fill width with one
             Prestel page
identity     wait for Prestel to ask for your
             user identity
[I][P]       send user ID then Password
```

Perhaps sign-on files aren't so difficult after all!

Once safely logged on to Prestel, you can set about the serious business of tracking the fortunes of your shares or whatever. Pages can be printed as ordinary ASCII text while on-line; the software automatically strips out the viewdata escape sequences used to control colour, double height, flashing characters and mosaic graphics.

Early versions of Communiqué supported flashing characters but the feature never worked properly and was deleted from version G2.0 onwards. This will be of no consequence to professional users, but their kids might lament its absence from Micronet's electronic playground ...

Communiqué has no facility to download CET-encoded software from Prestel or viewdata-standard bulletin boards. But, to the best of my knowledge, there is nothing available to download this format; so the omission probably doesn't matter too much. The truly determined might wish to download Beeb software for conversion to GW-BASIC with Apricot's B-TRAN utility; we have used B-Tran in the office successfully but that was on a relatively well-behaved program. Most games software (what else is there on Micronet?) uses chunks of Beeb assembler, which screws up B-TRAN. *(And see our downbeat review of* B-TRAN *in* APRICOT FILE *Release 1.12 — Ed.)*

The absence of a viewdata message frame editor makes the mailbox facility on Prestel slow and tiresome to use. Pages may be dumped to disk, either to a new file or appended to an existing one.

As the software versions have progressed, most of the viewdata routines have been rewritten in assembler. As a result, the screen display keeps up with the line data rate much better than in Teletype mode.

Viewdata Page Manager

This facility enables viewdata pages stored on disk to be displayed, printed, copied to another file or converted to text files for use with other software. File conversion may apply to the entire file or selected pages only.

Pages may also be selectively deleted, but a bug has been lurking here since the beginning of Communiqué: deleting pages does not reduce the file size.

Overall, the Page Manager is a useful training aid in a business environment in that it allows demonstration of the method of use and facilities of a viewdata service without the cost of being on-line.

Teletype mode

This is the mode I use most frequently; and, subject to the speed limitations of the software, it works well. Sign-on files to access Telecom Gold via PSS are supplied and all that is needed is to ensure that the last two digits of the Network User Address correspond to your Telecom gold system number. For MicroLink on system 72 it is as follows:

```
[blank line]    send Return
[T]             [wait a bit]
[blank line]    another Return
[T]             [wait a bit]
A2              tell PSS your terminal type
NUI?            PSS asks for your Network User
                Identity
NTLGOLD901TYM   NUI for Telecom Gold Closed User
                Group
ADD?            PSS asks what address it should
                connect to
A21920100472    NUA of Telecom Gold system 72
Sign On         Gold asks you to send user ID
ID [I]          send ID [space] User ID
Password:       Gold asks you for your password
[P]             send password
```

Some comms software expects you to bash that lot in manually — and the best of luck! The time delays [T] are defined on the last screen of the AMEND SIGN-ON setup routine. If something does go wrong during sign-on, Communiqué will try to recover the situation by carrying on from the last point it was able to recognise; or, as a last resort, it will attempt to start from the beginning of the sign-on file again.

Apricot has claimed that Communiqué handles MultiStream EPAD error-protected access to PSS since version G1.42; in fact the modem driver has only worked properly with EPAD since version Vf3.1.2 (6 November 1986). Even with this version, Communiqué is not compatible with all EPAD error-checking methods. Fortunately the error-check procedure required (vertical and horizontal parity checking of data blocks) is the EPAD default, which simplifies filling in the application form for a MultiStream password.

Beware, though, if you intend to use more than one type of comms system to access the same EPAD account: if they are equally dogmatic about accepting only one error-check, but it happens to be a different one, life will become very frustrating. More on EPAD error-checking and how to change the parameters on-line will follow in a later article.

When accessing any service via MultiStream using its EPAD error-correction, NUAs should be sent in their full international format. For instance, to access Telecom Gold System 85 via EPAD you need this:

```
[blank line]
ECF
[I]
NUA :
23421920100485
Sign On
ID ABC123
Password:
[P]
```

A couple of fiddles have been used here to overcome limitations in Communiqué. The details held in ACCESS.PAR allow for only two access parameters, normally your user ID and password; initial access to EPAD requires a password, then Telecom Gold wants to know your ID and Gold password. I have used the 'user ID' field to hold the EPAD password and the 'password' field to hold the Telecom Gold password. The Telecom Gold user ID (mailbox number) is not likely to change frequently and may be included as a line in the sign-on file without inconvenience.

The space between 'NUA' and the colon is essential and underlines the need for absolute precision when creating sign-on files. 23421920100485 is the full NUA for Telecom Gold System 85.

'ECF' is one of the CCITT X3 PAD commands — it is preceded by a ^P that does not display on screen but will appear in a file dumped to disk. ^P is the Data Link Escape character and does for comms what ESC does for printers; on no account should any file sent via EPAD or any other route into PSS contain X3 PAD commands unless they are intentional. Traumatic things can happen: for instance, ^P CLR will disconnect you without further ado and is the command used to terminate a call made over a leased data line. It's unfortunate that some word processors have appropriated ^P for use as a print control marker.

Within the limitations of these constraints, which are general and not an exclusive feature of Communiqué, files may be sent from or dumped to disk at will.

Caution is needed when using the 'Append to existing file' facility in Teletype mode, however. The appended text seems to start after the previous 'end of file' marker, which is left in the file where appended sections join.

Bulletin boards within the range of the modem may be accessed. The parameters screen needs only the BBS name and telephone number, and the sign-on file requires only a 'Y' to say that lines are terminated by a Return. The attachment file should show 8 data bits, 1 stop bit and no parity; host echo is normal and many BBSs allow line feeds to be negotiated at the time of registration.

Manual dialling cannot be used, which is an irritation when sampling BBSs to see which ones should be included permanently on the Connection Menu. And Communiqué cannot handle XMODEM or KERMIT error-correction protocols, which effectively rules it out for the serious bulletin board user looking to down- or upload files.

Communiqué has an encryption facility to convert prohibited characters into something acceptable. In principle this should permit word processor and even binary files to be sent through PSS, but only to another Apricot with Communiqué. I had a singular lack of success sending encrypted binary files through Telecom Gold; this may have been finger trouble on my part.

The encryption method for binary files appears to have close similarities with that used by Swadelands School to send compressed weather satellite pictures to MicroLink's Weatherlink feature.

Documentation

The original user manual for Communiqué was a disaster — no more than a print of some of the so-called Help screens. Later versions are much improved, though over half the wordage is taken up with descriptions of how to use Telecom Gold and EasyLink (now called Mercury Link 7500).

At least the documentation is now sufficiently comprehensive to allow the more adventurous to experiment with services not on the initial Apricot-provided list. All things considered, it's pretty good by Apricot standards.

Last words

For use with European CCITT-standard data networks, Communiqué has a head start over anything sired in the States — and many comms packages have their origins on that side of the Atlantic.

Cable hassles are nonexistent, of course, and it's easy to get going — especially for Prestel. Above

all, there are no compatibility problems: the software has been specifically written for Apricot computers and the internal modem.

Apricot has aimed the package squarely at the business user and features required by the bulletin board user have been ignored almost entirely. But its main drawbacks are the performance of the modem and the sheer bulk and sluggishness of the software. Provided you don't make a habit of accessing services with high connect-time charges, the speed limitations may be less important than ease of use and neatness of the complete installation. After all, you can read a report or make a cup of coffee while it chugs away.

Despite the many critical comments in this review I have used versions G1.32 onwards of Communiqué successfully for the past 18 months (though rather less successfully for the preceding four months with version G1.09). Right now I am looking for a replacement; but finding a package that works so much better that it justifies the outlay of personal funds is not proving easy ...

The package reviewed here has reached the end of its development; it is pointless to invent a wish list in the hope that Apricot will correct some of the deficiencies. The future for Communiqué lies with Apricots that run Windows, apparently including the Apricot-architecture Xen. The Windows modem interface enables the software to address the modem as COM2 in the conventional manner; it follows that the Windows version of Communiqué can drive external modems, and to that end Hayes-compatibility has been incorporated. I suppose the Xen-i 386 might just be quick enough to keep up with a 2400bps modem, but Apricot is playing safe and limiting the Mk 3 internal modem to 1200bps.

As a secondhand buy for the 8086 Apricots, Communiqué *is* worth considering — provided you can take advantage of its strengths without suffering too many of the weaknesses. But at a recommended retail price of £395 plus VAT for the modem and software, it's a rip-off by any reckoning.

Apricot is putting its current production effort into the IBM-compatible range. It is unlikely that any more true-Apricot internal modems will be made or that further development of the software will take place. The price will apparently be used as a marketing tool to regulate demand, ensuring that stocks remain available for a reasonable period but without leaving unsold stock on the shelf when demand eventually peters out. That's the theory according to Apricot, anyway.

Since Communiqué was first launched, the choice of alternatives has widened considerably with competing products offering a better performance, price or both. You don't *need* the internal modem; and if you're using an external RS-232-connected modem, there are many software options apart from Communiqué. So think carefully before buying. ◆

FOR

- \+ Easy to learn and use
- \+ Neat installation
- \+ No modem cable problems (no modem cable)
- \+ Works properly with packet data networks
- \+ Supports EPAD error-protection protocols
- \+ Comprehensive facilities in teletype mode
- \+ Adequate modem performance for Prestel and PSS access
- \+ Full compatibility assured

AGAINST

- — Overpriced
- — Modem performance
- — Slow
- — Limited support of viewdata facilities
- — Cannot handle XMODEM or KERMIT protocols
- — No clock
- — Cannot work with subdirectories

Communiqué

Description:	Internal modem (takes one slot) and async comms package.
Supplier:	Apricot Software via dealers
Address:	Apricot Software Shenton House Dudley Road Halesowen West Midlands B63 3NT
Telephone:	021-501 2284
Price:	£395.00 plus VAT
Configuration:	Any Apricot (though IBM-compatibles need the PX adaptor)
Conclusion:	Pros and cons — major plus is ease of installation and guaranteed compatibility; major drawbacks are poor modem performance, size of software, omission of many up-to-date facilities. Could be a good *secondhand* buy — if you can live with the deficiencies.

Historical Footnote

Bug Hunting

A partial guide to BIOS bugs

By Damien Thorne

If there's ever a fire at Apricot's Advanced Technology subsidiary, it would take a couple of days to rewrite the BIOS ... but an eternity to put all the bugs back in.

In this piece we take a (fairly) lighthearted look at some of the bugs which have appeared in Apricot BIOS releases. We'll look at what Apricot did to fix them, and what else got mucked up in that process.

Paperless printers

When the very first Apricot machines were shipped to a few select Beta-test sites, a number of errors kept rearing their ugly heads. The first was on the printer.

The BIOS once utilised the 'Paper Out' line in the printer port, which is set by the printer. When the printer has paper, the line is set as 'logic high' — meaning the printer has current flowing through it back to the computer. If the printer runs out of paper, it will stop the current going back to the PC.

At Apricot, the paper-out detection was tested and found to be fully working; but not all printer cables have a cable containing this line. In fact, very few do. And if a cable that does not have this line is used, the Apricot thinks that the printer is out of paper. So it tells MS-DOS. MS-DOS will then print 'Out of Paper Error — Abort, Retry, Ignore?'. This will also appear on the printer — which probably *is* loaded with paper — if the CONTROL-P switch is set!

Click clunk

This bug was fixed simply by disabling the paper-out detection in the BIOS. Another early weirdo was less serious but more intractable.

It's the 'click-click' syndrome. To recreate it, you'll need a twin-floppy Apricot PC. Boot it up and make sure there are readable disks in both drives. Enter PATH A:;B: at the prompt, type some random text and press Return. Both disk drives will be accessed one after the other — producing the 'click-click' effect; that's quite normal, because with the PATH statement we're telling MS-DOS to look on both disk drives for the program name which is the random text.

The first Apricot machines were shipped with the MS-DOS PATH set up as above, so lots of users were getting 'click-click' when they mistyped a program name.

The problem was not solved until MicroSoft was called in. A lot of people looked very silly.

Serial saga

The serial port driver is like an enormous pit: you pour your data in, and depending on the BIOS version it will either send it down the serial line (often at random baud rates) or it will completely forget everything. There are so many bugs in the serial driver that it would take an eternity to explain them all, but we've selected the best ones for your delectation.

First, the anomaly of setting baud rates. In early versions of the Apricot, the serial clock was slightly inaccurate; this meant that Apricots would talk to each other reliably, but they couldn't talk to other machines. This problem was soon solved, only to completely muck up some baud rate values (they varied). This was fixed, only to cause problems with the bits-per byte setting ... And *that* problem still exists; the BIOS doesn't mask out the unwanted bits, so problems can be caused when reading data from the serial port.

Silver screen

No part of the BIOS is entirely free from bugs, but the screen driver is a notorious problem area.

Doyens of escape sequences will know about ESC 'Y' to position the cursor on the screen. In very early versions, the line and column numbers it took were relative to 31 decimal rather than 32 as on the Sirius; so programs ported from the Sirius produced some strange screen displays. This problem also took æons to solve.

More MicroScreen

My favourite bug concerned a communications package written by an Apricot user. The software was very good; but after exactly 99 blocks of data had been transmitted, the 100th block would invariably be corrupted.

This problem was investigated by the whole BIOS team at Apricot for three weeks without success. Then one day, somebody using the package noticed that the program was echoing on the MicroScreen the number of blocks being sent; when the count got above 99, the MicroScreen was scrolling up because the extra character in the number string went off the end of the last line.

It turned out that while the MicroScreen was being scrolled, the serial chip (which also controls the keyboard) was getting a bit overworked: it could either scroll the MicroScreen *or* receive data ... but not both.

This bug has never been fixed: the program's author was told to move his block numbers on the MicroScreen back a few spaces, and the problem magically went away. ◆

The DEBUG Tutorial

DEBUG for Beginners

Part One: First steps

By Simon Potter

Many of the programs and utilities you use have bugs, idiosyncrasies, awkwardnesses and inappropriate operations. Well, they can be amended easily with DEBUG. If you don't have something like the DEALS FOR READERS program UNDELETE, you'll find that DEBUG is also handy for recovering 'lost' files on disk. And if you want to try your hand at writing tiny COM programs (as featured in David Langford's *Disinformation Column*), DEBUG is the easiest way of writing assembly code.

Assembly code? Yes — to get the best from DEBUG, you're going to have to learn something about assembly language. Don't worry about that — it won't be too hard! And if your appetite for assembler is whetted by the brief introduction here, watch out for our follow-on 'teach yourself assembler' series.

Uses for DEBUG

- adapt, customise and/or debug existing programs
- write your own assembler utilities
- produce multi-choice menus, pop-up Help screens, and other aids
- develop assembler routines for use from programs written in Turbo Pascal, Basic, or whatever
- examine and temporarily modify the contents of memory
- examine and permanently modify the contents of a disk — for instance, to undelete accidentally-erased files

In essence, DEBUG is just an editing program — exactly like EDLIN or any word processor. But as the name implies, DEBUG is designed as a programmer's tool: it is primarily intended for use with 'command' and 'executable' files — MS-DOS programs with the suffixes COM and EXE, respectively — rather than *text* files. DEBUG *can* be used to edit text files, but a word processor is so much easier; and it *can* be used just to alter the contents of individual storage locations in memory or on a disk — but we'll come to them later.

If you don't have a current version of the Apricot-format DEBUG, by the way, you'll find it on DEALS FOR READERS disk AP002.

About COM and EXE files

Back to basics: COM and EXE files can be run simply by typing the name of the file (minus the suffix, and in some cases with one or more parameters) at an MS-DOS prompt. For instance, to run the CHKDSK.COM disk-checking utility you enter ...

```
A> CHKDSK
```

Those COM and EXE files are written in assembly language in order to interface directly with the MS-DOS operating system and the hardware. The alternative is to use a programming language, in which case the resulting program may need to be run through an interpreter like MSBASIC.COM; the interpreter, which you'll note is itself an assembly-language program, acts as a halfway house and quite literally 'interprets' the contents of your program into a form that MS-DOS and the computer can understand. That's why you run a Basic program called, say, TESTPROG.BAS with a command like

```
A> MSBASIC TESTPROG
```

The other kind of programming with a language involves a compiler, a COM or EXE file which recodes your program into assembler format so that it can be run directly. If you have the Basic compiler BASCOM.EXE you could run it on TESTPROG.BAS and produce a program called TESTPROG.COM.

Hex basics

So we're going to talk about assembler programming. You will also need a touch of facility with hexadecimal numbers, because the Apricot (or more specifically the Intel 8086 microprocessor) handles data 16 bits at a time (usually).

If we'd been born with sixteen fingers rather than ten, we would already be using hexadecimal notation instead of our current commitment to the number 10. Instead of using ten unique digits (0 to 9) in arithmetic, we would have had sixteen — and we'd probably have counted our sixteen fingers something like 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F. Anyhow, those are the digits used in hexadecimal notation to provide sixteen unique digits; and just as we can use the ten digits of decimal notation to construct numbers, so we can use those sixteen to build hexadecimal numbers.

The computer works with hexadecimal notation. So as far as DEBUG and assembler programming are concerned, you'll have to use hex too. Getting into the habit of translating from decimal to hexadecimal can be a mite troublesome, but it's more a matter of understanding why it needs to be done than worrying about the mechanics of how to do it.

And there are a number of short-cuts. You can start thinking in hex; you can have the computer do the conversion maths for you; or you can refer to the table over the page.

Hexadecimal-to-decimal conversion

DIGIT IN HEXADECIMAL NUMBER									
5 [leftmost]		4		3		2		1 [rightmost]	
hex	dec	hex	dec	hex	dec	hex	dec	hex	dec
0	0	0	0	0	0	0	0	0	0
1	65,536	1	4,096	1	256	1	16	1	1
2	131,072	2	8,192	2	512	2	32	2	2
3	196,608	3	12,288	3	768	3	48	3	3
4	262,144	4	16,384	4	1,024	4	64	4	4
5	327,680	5	20,480	5	1,280	5	80	5	5
6	393,216	6	24,576	6	1,536	6	96	6	6
7	458,752	7	28,672	7	1,792	7	112	7	7
8	524,288	8	32,768	8	2,048	8	128	8	8
9	589,824	9	36,864	9	2,304	9	144	9	9
A	655,360	A	40,960	A	2,560	A	160	A	10
B	720,896	B	45,056	B	2,816	B	176	B	11
C	786,432	C	49,152	C	3,072	C	192	C	12
D	851,968	D	53,248	D	3,328	D	208	D	13
E	917,504	E	57,344	E	3,584	E	224	E	14
F	983,040	F	61,440	F	3,840	F	240	F	15

The chart works for converting hexadecimal numbers of up to five digits into their decimal equivalent. To use it, locate the decimal values corresponding to the hexadecimal digit in each position of the hex number and then add the decimals. For instance, the decimal equivalent of F2FA is 62,212:

fourth digit:	F hex	=	61,440 decimal
third digit:	2 hex	=	512 decimal
second digit:	F hex	=	240 decimal
first digit:	A hex	=	10 decimal
SUM			62,212 decimal

Similarly, FFFF hex is 61,440 + 3,840 + 240 + 15 = 65,535. To find the decimal value of 66 hex, we'd add 96 + 6 = 102.

The Rules

The results of using DEBUG on a file can be swift and dramatic. So here are two key Rules of Using DEBUG:

1. **Never DEBUG the only example of a file. Always work on a copy of your file.**

2. **Make sure it's not the only copy you've taken of that file. Always work on an extra copy.**

There's another safety precaution you should take — create a working disk containing DEBUG and the file(s) you're working on. That way you don't risk screwing up anything you really need. So that you can run these exercises independently of anything else, it also makes sense to start with a disk that has a copy of the operating system on it; do that by formatting with FORMAT /S (if you have the FORMAT utility — it's on DEALS FOR READERS disk AP020), or use the SYS utility after formatting.

Let's get under way. We're first going to demonstrate some of the DEBUG commands, and for that we'll be DEBUGging a file called MENU.TXT. This is a simple text file that's part of my AUTOEXEC start-up routine — you'll find something similar on DEALS FOR READERS disk AP001, and the principle is outlined in my Batch File Cookbook in APRICOT FILE Release 2.7; but for now, all we need is something easy to play with and quick to enter — if you want to join in, see the panel for the contents of MENU.TXT.

The example file

MENU.TXT is a simple text file. You can use a word processor and save it in ASCII format (SYSTEM mode, if you're using SuperWriter); or you can enter it with EDLIN.

```
<Space> <Return>
<Return>
<Return>
<Return>
<Return>
<Return>
<Return>
<Return>
<Return>
<16 Spaces> 0. <6 Spaces> TIDY-ALL <Return>
<16 Spaces> 1. <6 Spaces> CHKDSK C: <Return>
<16 Spaces> 2. <6 Spaces> CHKDSK A: <Return>
<Space> <Return>
<16 Spaces> 3. <6 Spaces> NEWSWRITER <Return>
<16 Spaces> 4. <6 Spaces> SUPERCALC 3 <Return>
<16 Spaces> 5. <6 Spaces> CARDBOX <Return>
<Return>
<Return>
<Return>
```

Getting started

There are two ways to start DEBUG with a file; both produce the same result, and either can be used. You can include the file's name in the command line:

```
A> DEBUG MENU.TXT
-
```

(We're assuming that your DEBUG working disk, complete with the target file MENU.TXT, is on drive A.) Or you can load DEBUG first and call up the file once DEBUG is running:

```
A> DEBUG
   -N MENU.TXT
   -L
```

With either method, you'll get the DEBUG prompt — a hyphen — appearing on the next line. In the second case, you're following the prompt with the Name command to specify a file for DEBUGging; and then the Load command loads that file from disk into memory.

Don't worry if you have any problems when using DEBUG; no changes you make to a file will be permanently saved unless you very specifically tell DEBUG to Write (or save) the amended version, and you can enter the DEBUG command **Q** (for Quit) at any time to return you to the MS-DOS prompt.

See what you've got

So now we have loaded DEBUG and a file for it work on. At the hyphen prompt, enter 'D'; that's the Dump command. The resulting display should look something like the panel at the foot of the page.

The **D** command (which, like all DEBUG commands, can be entered as upper- or lower-case) displays 128 bytes of memory — the contents of 128 memory locations, in other words. If you hadn't specified any program when you loaded DEBUG, you'd get the first 128 bytes from the start of memory; otherwise you get 128 bytes from wherever the program starts in memory. The first time you use **D**, this will be the first 128 bytes of the program; subsequently it's 128 bytes from the last address displayed.

Great, but what's that forest of numbers? Well, there's three columns. At the start of each line you've got a reference-point, a memory address in hex notation. The centre section tells you what's in the sixteen memory locations starting from that address — again, it's all in hexadecimal. On the right you have the sixteen ASCII-code equivalents for those hex numbers; a space shows up here as a dot, otherwise you're likely to recognise characters and words.

Don't worry if you don't have the same numbers in the lefthand column when you come to try DEBUG for yourself with MENU.TXT. The first four numbers in the start-location references here indicate a 'segment' address, and that will vary from one Apricot to another depending on the amount of memory installed, the precise configuration of your system, and where MS-DOS has actually decided to put MENU.TXT in physical memory.

The second four numbers are 'offsets' within the program, and these should be the same on your use of DEBUG. We'll cover this in the next instalment, but for the sake of getting started with DEBUG let's simplify things: the segment address indicates whereabouts in physical memory the file MENU.TXT is located, the offsets show the location of individual bytes of MENU.TXT *from* the start of the segment.

Because MENU.TXT won't change its length or the relative position of its constituent bytes, the second block of four hex numbers — the offset — should be the same on your screen as on ours. The first four, the segment, might well be different if MS-DOS has decided to put MENU.TXT somewhere else in memory.

The middle section shows sixteen two-digit hex numbers per line. Each of these is a byte and corresponds to an ASCII code of some kind. The hyphen in the centre is there just to help you to keep track of where you are; you'll actually have to work out the address of each byte by adding from the reference location in the lefthand column — so the fifth hex number along the first line is at location 0105, the tenth is at 010A, and so on. (Note that you're adding to the offset: the segment address stays the same. The full, verbose description of the location of the fifth byte in MENU.TXT would be 0473:0105 — offset location 0105 hex in segment 0473. But since the segment part is likely to change from one Apricot to another, and since the computer can be left to choose that anyhow, the offset portion is all we really need.)

As for the rightmost column, you'll observe that it's a fairly comprehensible rendering of the text in MENU.TXT. That's why we suggested you start your DEBUGging with this particular file; it's easier to see the relationship between hexadecimal numbers and the ASCII or 'English' letters and words you typed.

```
-D
6ECD:0100  20 0D 0A 0D 0A 0D 0A 0D-0A 0D 0A 0D 0A 0D 0A 0D    ................
6ECD:0110  0A 0D 0A 20 20 20 20 20-20 20 20 20 20 20 20 20   ...
6ECD:0120  20 20 20 30 2E 20 20 20-20 20 20 54 49 44 59 2D      0.       TIDY-
6ECD:0130  41 4C 4C 0D 0A 20 20 20-20 20 20 20 20 20 20 20   ALL..
6ECD:0140  20 20 20 20 20 31 2E 20-20 20 20 20 20 43 48 4B        1.       CHK
6ECD:0150  44 53 4B 20 43 3A 0D 0A-20 20 20 20 20 20 20 20   DSK C:..
6ECD:0160  20 20 20 20 20 20 20 20-32 2E 20 20 20 20 20 20            2.
6ECD:0170  43 48 4B 44 53 4B 20 41-3A 0D 0A 20 0D 0A 20 20   CHKDSK A:.. ..
-Q
```

If you had loaded a COM or EXE file, you probably wouldn't see very much understandable text in this column — because such files employ assembler code mnemonics and hex numbers, which don't always translate into recognisable ASCII codes. As it is, you can see that 44 hex is a capital 'D'; 42 is a capital 'B'; 79 is a lower-case 'y' ... and so on. 20 is a space, and DEBUG displays a dot when there isn't a printable ASCII equivalent of a hex number.

Ways with Dump

You can ask to see the contents of memory starting at any location. For example:

```
-D 0101
```

DEBUG would display eight lines of sixteen locations apiece starting at offset 0101. (You don't actually need a space after the **D** command — or indeed after any DEBUG command. We've standardised on its inclusion for the sake of clarity.)

When the **D**-for-Dump command is used without parameters, or with a specific start address, it displays 128 bytes (eight lines) at a time. You can ask for more or less than that, however.

```
-D 0100 FFFF
```

That would produce a display of 65,535 bytes starting at 0100 (FFFF in hex is 65,535 in decimal.)

And you can view a specific range of addresses with one of two methods:

```
-D 0104,0108
```

Here the start and end of the range of addresses is given, separated by a comma. This would **Dump** only the five lines between 0104 and 0108 inclusive.

The alternative is to specify the start address and a number of bytes to be dumped:

```
-D 0104 L 5
```

That would have the same effect. The **L** signifies 'Length'; it must be there, and it's followed by a hex number. (Again, you don't really need spaces around the 'L' — we have them for clarity.)

So the full syntax for **Dump** is:

```
D [start address or range] [number of lines]
```

... where everything is in hex; the two parameters are optional; a range consists of two address offsets separated by a comma; the number of lines to be dumped is preceded by an 'L'; and a number of lines can only be specified if a start address or range of addresses is given.

Some people prefer just to hit a 'D', study what's on the screen, and then enter 'D' again when they're ready for another eight lines. But don't be afraid to ask for a lot of lines and scroll through them; you can always use CONTROL-S to pause the display and CONTROL-C to abort it, and D 100 will return you to your starting point.

Similarly, you can use CONTROL-P to print out the results of a **Dump** with a minimum of keystrokes.

Changing memory contents

So you can see the hex numbers that make up your program, and if it's a text file you'll also be able to see the ASCII characters that correspond to those hex numbers.

And now for some editing: you can change the hex numbers and save the new version of the file. For that we need the **Enter** command — and a table of hex-ASCII equivalents, since this command is used to overwrite the present contents of particular memory locations with a new hex value.

The simplest form of this command has 'E' followed by an address offset:

```
-E 100
```

(You can omit leading zeros from addresses used with any DEBUG command.) The resulting display will look something like this:

```
-E 100
6ECD:0100  20.
```

DEBUG puts on to the screen the current contents of that address, adds a dot, and leaves the cursor immediately after it. You're being invited to type in something to replace the existing contents of this address; you can enter a two-digit hex number representing the replacement value, or you can press the space bar to leave that one as it is and move on to the next byte.

If you want to leave it, tap the space bar. Now the screen looks like this:

```
-E 100
6ECD:0100  20. 0D.
```

0D is the hex code for a Return. Just for the sake of example, let's change this one. Enter '44' (that's hex for a capital 'D'); press the space bar again and you'll move on to the next byte in the sequence, with the Return code changed to a 'D'.

```
-D 100

6ECD:0100  20 44 0A 0D 0A 0D 0A 0D-0A 0D 0A 0D 0A 0D 0A 0D    .D.............
```

As you can see from the righthand column, the non-printing dot that formerly represented the Carriage Return has changed to a 'D'; and the second byte along the centre column is 44 rather than 0D.

Searching

The Search command is a quick way to locate a particular address you want to change. The syntax is ...

```
S [range] [search-for item]
```

The range is the start and end addresses (in hex, naturally) for the search; separate them by a comma or a space, and have another comma or space before the search-for item. That parameter can be a hex value or a character string (one or more characters); in the latter case, enclose the string by single or double quotes. For instance:

```
-S 100 120 20
7A54:0100
7A54:0113
7A54:0114
7A54:0115
7A54:0116
7A54:0117
7A54:0118
7A54:0119
```

That command searches addresses 100 to 120 inclusive for the occurrence of a hex code 20, which happens to be an ASCII space. As you can see, DEBUG will display all locations in which it found that number. Once you have them, you can use D-for-Display to see the relevant entries.

Here's an example of a search for a character string:

```
-S 100 200 'CHK'
7A54:014D
7A54:0170
```

You'll observe that DEBUG comes back with the location at which the string starts. Note that for DEBUG commands and hex addresses you can use upper or lower case; but in a search-for string, you have to be exact — enter the text you're looking for *precisely* as it was used in the file, or DEBUG will never find it. The program will return a blank prompt if it can't locate the searched-for item, and that applies if it does not locate an *exact* match for your request.

Hex and character data can be mixed in the same request; this is perfectly valid ...

```
-S 100 FFFF 12 34 'abc' 56
```

It requests a search from 100 through 64K bytes for the sequence of '12 34 a b c 56' occurring in that order.

Be warned — just because DEBUG finds a match, it doesn't mean that the data located is what you're looking for. If you look for 21 hex, you might find several occurrences where that data is used as an ASCII code (it's the code for an exclamation mark); but there again it might be the data for a MS-DOS interrupt call (Interrupt no. 21 is used a lot) or part of another Assembler statement. So use D-for-Display to check.

Saving the edit

At this point, only the copy of the file that's sitting in memory has been modified. If you were working on a program, you could test your modifications from within DEBUG: the G-for-Go command will attempt to execute the changed program and DEBUG will tell you if it terminated normally or had problems.

Since we're DEBUGging a text file, that doesn't apply. If you want to save the revised version on disk, you'll need the Write command; this saves the file under the same name as you loaded. Write starts the save from wherever you happen to be within your DEBUGging of the file, so make sure you're back at the start of the file — the easiest way to do that is a D 0100 command:

```
-D 100 L 1
6ECD:0100  20

-W
Writing 00EF bytes
```

(Asking DEBUG to Dump only one byte is a quick way to get back to the start.) The Write command is usually used like this, without parameters, but for advanced applications you can specify how much of the file is to be saved and at what address the saving should start. You always get the confirming message saying how many bytes have been saved.

To avoid overwriting the existing file, specify a different name (and/or a different drive identifier, and/or a different directory path) before using Write:

```
-D 100 L 1
6ECD:0100  20
-N C:\TEMP\NEWFILE
-W
Writing 00EF bytes
```

Instant hex arithmetic

Finally for this instalment, the Good News. You aren't too happy with handling hex? Don't worry — there's a hex arithmetic command. If you enter H 19F 10A at the DEBUG prompt, the program will add the two numbers together — and it will also subtract them for you before displaying the results. In this case the results are 02A9 (sum) and 0095 (difference). Handy, no?

So the H command has just two parameters — the two numbers to be added and subtracted — and

DEBUG's response is the sum and difference of the numbers, in that order. The numbers can be one to four hexadecimal digits long. The addition and subtraction are unsigned.

Note however that the **H** command won't handle any carry beyond the fourth hex digit:

```
-H 5 6
 000B FFFF
```

Here we're adding 0005 and 0006. The sum is 000B as shown, but the difference is -1: since there is no carry, we get FFFF.

Last words

That'll do for now. You should be able to handle hex, even if you don't understand it; and you should be able to appreciate why you need to use hex, even if you don't appreciate why DEBUG uses it. You should be able to use DEBUG to load a text file, to make byte-by-byte changes, and to save the revised version. Next month: fancier editing and an introduction to how MS-DOS handles memory. ◆

DEBUG aide memoire

These notes summarise the key rules about entering DEBUG commands and parameters; you can see how they apply in the discussion of Dump, Search and Hex, but they also apply to all other DEBUG commands and parameters.

COMMAND KEYWORDS

Command keywords and the L-for-Length parameter can be entered as capitals or lower-case letters. Throughout this series we include a following space for legibility, but it's not strictly necessary.

SEPARATORS

Command keywords and parameters can be separated by spaces or commas — or both. (Again, that's just for legibility; unless the command involves a hex number, you don't actually need any separators at all.)

INPUT ERRORS

Errors in command input are indicated by the message ^Error and the return of DEBUG's hyphen prompt.

BYTE VALUES

Byte-value parameters are one or two hex digits.

ADDRESSES

Address parameters with commands can be in one of three forms:

- a hexadecimal offset value
- a segment and offset in hex and in that order, separated by a colon
- a segment register name and a hex offset in that order, separated by a colon

Leading zeros in offset addresses can be omitted.

ADDRESS RANGES

There are two ways of specifying an address range:

- specify the start address and a length. The address is followed by a space or comma, the letter L, another space or comma, and a hex value indicating the number of bytes from that address
- specify the start and end addresses, separating them by a space or comma.

ASCII-Hexadecimal equivalents

Non-printing codes

The first 32 codes in the ASCII set are non-printable control characters which are used to send commands to an output device — usually a video monitor or printer, though some of these codes are specific to data comms.

Note: The ^ character indicates an ESCape code, usually produced by holding down CONTROL while typing the character.

Hex code	Decimal	ASCII
00	0	Null
01	1	^A SOH [Start of Heading]
02	2	^B STX [Start of Text]
03	3	^C ETX [End of Text]
04	4	^D EOT [End of Transmit]
05	5	^E ENQ [Enquiry]

06	6	^F	ACK [Acknowledge]
07	7	^G	BEL [Bell]
08	8	^H	BS [Backspace]
09	9	^I	HT [Horizontal Tab]
0A	10	^J	LF [LineFeed]
0B	11	^K	VT [Vertical Tab]
0C	12	^L	FF [Form Feed]
0D	13	^M	CR [Carriage Return]
0E	14	^N	SO [Shift Out]
0F	15	^O	SI [Shift In]
10	16	^P	DLE [Data Line Escape]
11	17	^Q	DC1 [Device Control 1]
12	18	^R	DC2 [Device Control 2]
13	19	^S	DC3 [Device Control 3]
14	20	^T	DC4 [Device Control 4]
15	21	^U	NAK [Negative Acknowledge]
16	22	^V	SYN [Synchronous idle]
17	23	^W	ETB [End of Transmit Block]
18	24	^X	CAN [Cancel]
19	25	^Y	EM [End of Medium]
1A	26	^Z	SUB [Substitute]
1B	27	^[	ESC [Escape]
1C	28	^\	FS [File Separator]
1D	29	^]	GS [Group Separator]
1E	30	^^	RS [Record Separator]
1F	31	^_	US [Unit Separator]

Printing codes

The remaining ASCII codes are printable. For the purposes of reference, the rest of this list is arranged in character sequence. Compare the symbols that appear on the Apricot's screen with those produced by your printer — not all printers will print these characters as the ASCII committee meant them to be displayed.

ASCII	Hex	Decimal	ASCII	Hex	Decimal	ASCII	Hex	Decimal
space	20	32	@	40	64	‘	60	96
!	21	33	A	41	65	a	61	97
"	22	34	B	42	66	b	62	98
#	23	35	C	43	67	c	63	99
$	24	36	D	44	68	d	64	100
%	25	37	E	45	69	e	65	101
&	26	38	F	46	70	f	66	102
'	27	39	G	47	71	g	67	103
(	28	40	H	48	72	h	68	104
)	29	41	I	49	73	i	69	105
*	2A	42	J	4A	74	j	6A	106
+	2B	43	K	4B	75	k	6B	107
,	2C	44	L	4C	76	l	6C	108
-	2D	45	M	4D	77	m	6D	109
.	2E	46	N	4E	78	n	6E	110
/	2F	47	O	4F	79	o	6F	111
0	30	48	P	50	80	p	70	112
1	31	49	Q	51	81	q	71	113
2	32	50	R	52	82	r	72	114
3	33	51	S	53	83	s	73	115
4	34	52	T	54	84	t	74	116
5	35	53	U	55	85	u	75	117
6	36	54	V	56	86	v	76	118
7	37	55	W	57	87	w	77	119
8	38	56	X	58	88	x	78	120
9	39	57	Y	59	89	y	79	121
:	3A	58	Z	5A	90	z	7A	122
;	3B	59	[	5B	91	{	7B	123
<	3C	60	\	5C	92	|	7C	124
=	3D	61	]	5D	93	}	7D	125
>	3E	62	^	5E	94	~	7E	126
?	3F	63	_	5F	95	Del	7F	127

The Q&A Clinic:

Questions by subscribers, answers by APRICOT FILE

Between us, the staff of APRICOT FILE and our contributors offer an unusually wide range of expertise and experience. As many subscribers will know, we handle a lot of queries each month by mail or by phone; and every so often we run the Q&A page to present what we feel are the more interesting and/or useful of these. If you want to make use of the service, just drop us a line ...

Q. Speed-up utilities

How do FAST, SPRINT, HARE, and QUIK work? Do they all do the same thing? What's the difference between them — and which is best? Is one more suitable than the others for a particular machine?

A. They all seem to do much the same kind of thing — they sit around in memory to short-circuit MS-DOS checks for redirection, pipes, filters and CTRL-P echoing. So they speed up screen handling at the cost of disabling those facilities. There's no particular BIOS or machine preferences, and the differences between them are essentially cosmetic ... except that HARE and SPRINT can both be turned off instantly with full recovery of the memory they occupied.

Best buy? Probably HARE, but only because you get so much else on the ACOLYTE disk (DEALS FOR READERS no AS035).

Q. WordStar miscellany

I find WordStar a magnificent program, but I'm a little confused by what we actually have. It claims to be version 3.4 on the disk label and in the demo, and 3.4 should include the CorrectStar spelling checker; but it boots up as WordStar 3.3 with SpellStar, which is useless. Can anyone help? And is it possible to run MicroPro's new WordStar 4 on the Apricot? If not, why not?

Why does WordStar Professional print at half the speed of WordStar 3.4 on my Apricot with Epson printer? My dealer said it was due to "technical reasons", but can it be fixed by DEBUG?

Does anyone know how to enter accented characters at the keyboard in WordStar 3.4? My daisywheel can print them, I can produce them at MS-DOS level, and I can convert non-WordStar files so that accented characters appear on the screen when read by WordStar ... but WordStar won't let me type them. It ignores all keyboard files other than its own. What is the secret?

A. IBM versions of WordStar 3.4 have CorrectStar, Apricot versions have the older (and worse) American SpellStar. CorrectStar was never implemented on the Apricot, as far as we know.

WordStar 4 is IBM-only, meaning it is customised for that hardware; and given MicroPro's attitude to Apricots it will probably stay that way. You can like it, lump it or solve *all* the WordStar problems mentioned above and plenty more by buying WordPerfect ...

As for the speed problem — no, you can't patch WordStar to speed it up. At least, not without a great deal of preliminary investigation and then much recoding. Early versions of WordStar 3.4 printed fairly quickly (but only locally — it would not accept redirection of PRN on a local network). Later versions, those shipped with the WordStar Professional pack, do print very slowly (but then they work ok with the LAN). We assume less patching of MS-DOS is used, hence the lethargy. Forget WordStar, buy WordPerfect or stick with SuperWriter.

Nor is there a solution for accented characters. Short of patching the keyboard attributes held in WordStar, which will foul up normal use, the only option we can think of is to use an IBM-compatible machine for printing (on IBMs and clones, special characters at the far end of the ASCII character set can be produced by holding down ALT and typing the decimal value of the ASCII code on the number pad).

ACOLYTE's utilities for sending special codes to the printer would be useful with just about any other WP package, but ACOLYTE and WordStar don't get on too well together.

Q. Communiqué

I am having problems downloading from Communiqué. Having used the MicroScreen keys to send a pre-prepared file, as soon as the message finishes weird vestigial words from my edit files appear and the system disconnects. The prompt >\N then comes on to the screen with a disconnect message. Any idea why?

A. Well, there's not enough information here to be sure; but it could be down to illegal characters in the file being sent, especially if PSS is being used. VT52 and VT100 emulations may be susceptible to similar problems.

Q. Conversion

Do you know a method of getting an Apricot PC to read IBM-compatible 3.5in disks from a Toshiba portable?

A. It depends how frequently you need to do this. Options are:

- If you've got double-sided drives, convert to BIOS 3.1 on the Apricot — it has an inbuilt option for reading and writing IBM-format 3.5in disks (but double-sided 720KB disks only).
- Use an electronic mail service as a file transfer medium. This means you will need a subscription to one of the e-mail services, a modem on one or both machines, and compatible comms software on both (such as the TRANSEND program, which we can supply in both formats from our DEALS FOR READERS lists); you then send from one, read on the other. This approach works safely only for text files and other straight ASCII material.
- Or you can plug the two machines together with an RS-232 cable and transfer files using an appropriate package in each — such as TRANSEND again, or the public-domain program KERMIT, both available from DEALS FOR READERS in Apricot and IBM 3.5in formats
- For occasional conversions, you might mail us the disks and we'll convert them for a modest fee.

Q. BSAVE/BLOAD

I have found that BSAVE and BLOAD provide a fast and efficient way of saving and recalling the screen for MS-BASIC programs — useful in referring to 'help' information or an index so that the saved screen can quickly be recalled after reference. To save the screen I use ...

```
DEF SEG=&HF000
BSAVE "SCR.SAV",0,&HFA0
```

And to recall it:

```
BLOAD "SCR.SAV"
```

(This works particularly well with RAMdisk.) Unfortunately I want to compile most of my programs now, and the Basic Compiler doesn't support BSAVE and BLOAD; is there an equivalent construction, perhaps using PEEK and POKE?

A. Yes, this is possible. PEEK the contents of the screen RAM into an array and POKE them back one byte at a time. If you're not using graphics, you could also move the screen RAM from &HF000 down into part of the graphics map — that 40K could hold ten screens, to restore in any order!

Q. Technical WP

I do a lot of word processing that involves formulae, Greek characters and other special symbols — and I have still not found a word processor that meets my needs. I have a Mannesmann Tally MT180L printer and an Apricot PC with 256KB and twin single-sided disks.

A. There isn't any obvious add-on to SuperWriter that could be used. But there is a purpose-designed 'scientific' word processor available for the Apricot — VUWRITER from **Vuman Computer Systems** in Manchester, tel **061-226 8746**. That package will almost certainly demand 512K, though. You might also enquire about BRIT SCIENTEX; it was written by VUWRITER's original author and reportedly improves on his first try. A Sirius version was sold outside the UK, so it may be available in Apricot format ... **Scientex** is on **0438 351999**.

Q. SuperWriter date prompt

I have tried to implement the date prompt suggested by Steve Rainey in the *Readers' Writes* pages of APRICOT FILE Release 2.6. It doesn't work in SuperWriter, but it does work in MS-DOS when I come out of the word processor. Any ideas why?

A. Basically, SuperWriter takes over the screen. What works under MS-DOS will be lost when you load SuperWriter.

Q. Lotus Dates

I would like to prepare Lotus 1-2-3 reports automatically for specific dates by using a macro; briefly, I am trying to copy a date and add 30 to it in order to force the month-name to change. The @DATE function seems to be the obvious candidate, but I can't persuade it to work.

A. Adding 30 isn't the best way if you just want to change the month. That should be simple enough with a calculated reference to the cell containing the starter date, though the actual formula may be a bit verbose. Assuming your start date is in cell A1, try this:

```
@DATE,@YEAR(A1),@MONTH(A1)+1,@DAY(A1)
```

(Don't forget — there should be no spaces in the @ functions.) @DATE uses year, month and day in that order; enter this formula, and you should be able to construct a new date with the same year and day references as in A1 but with the next month showing.

Moving beyond December into the next *year* might be trickier, but that too is possible to automate. For the @YEAR part of the formula, use an IF which changes the year when the start-date month is December:

```
@IF@MONTH(A1)=12,@YEAR(A1)+1,@YEAR(A1)
```

And replace the @MONTH bit with another IF construction that switches from December to January:

```
@IF@MONTH(A1)=12,1,@MONTH(A1)
```

Q. F1 and BIOS 3.1

I have recently had my 256KB Apricot F1 upgraded to BIOS 3.1. But now the display appears to start at about column 25, with the righthand end of the display wrapping around to appear on the left on the screen. Any solutions?

A. You probably used to get 256-line mode and you're getting 200-line mode from BIOS 3.1 (or *vice versa*).

Q. Segment-switching in Basic

I have a 640KB Apricot with 10MB disk; I want to run a 41,000-byte GW-BASIC program with Graphics and 800400EG loaded. But Basic can use only one segment, so I don't have enough room in memory. It has been suggested to me that I write a machine-code routine to switch in another segment and make 60K available for programs; how do I do this?

A. Well, the simplest solution requires no assembler — why not keep all the data on disk and just fetch whatever's required as and when you need it? With all that memory, you could run a huge RAMdisk for this.

Q. RINSTALL and defaults

I have patched SuperCalc 3 to load my own keyboard: no problems, the program works perfectly with and without graphics. But to save loading the keyboard at the start of each session and reloading the default afterwards, I used the 'Black Hole' utility RINSTALL to prepare a small COM file that does the job. This works — except when viewing a graph: when the Return key is pressed to return to the program, the screen fills with garbage and the program aborts. Any thoughts?

A. We don't know why it crashes; but CTRL-CLEAR usually re-establishes text mode and Return should bring back the MS-DOS prompt. Overall, though, RINSTALL is a bit of a pig ...

Q. Ansible extras

I would appreciate your advice on which of ASD1 and ASD2 would be of most use to a new user of SuperWriter version 1.02. And I particularly like the WYSIWYG approach of GEMWrite; is anything like that possible with later version of SuperWriter?

A. Our view is that *both* are indispensable if you're using SuperWriter 1.02, though ASD1 is was produced first and so probably reflects the authors' initial concerns (which may also be yours). As for WYSIWYG, you'll have to dump SuperWriter in favour of something else if you want on-screen justification; but you can at least produce all (or nearly all) of the characters on your printer if you set up with an appropriately-configured keyboard and font. Ansible's APP utility on the ACOLYTE disk probably provides the simplest method of doing this ...

Q. Super XQTs

Do you know of any tutorial/reference book that explains SuperCalc and SuperWriter XQT files *in full*? I am using an Apricot Portable.

A. Diligent perusal of back issues of APRICOT FILE should deliver the goods — try Releases 2.1, 2.2 and 2.5. But in terms of everything-you-need-in-one-place books, Ansible Information has the only recommendable such publication; it costs £12.50 (inc p&p) and is comprehensive on SuperWriter XQTs with an add-on piece on SuperCalc's.

Q. GEM drivers

I am using an F2 with 768K, and the local agent here (New Zealand) isn't able to help with this problem: my daisywheel printer is a Japanese Nakajima, and while it will print perfectly in draft mode I am unable to get it to work from the GEM output file. Are there any new printer drivers available?

A. GEM will work ok on a daisywheel in draft mode, and for text only; but you need a dot matrix printer to get GEM output to print all the facilities available — including graphics.

Q. Sounds like ...

I am looking for a software package to run on an Apricot Xen-i which will search a database file for similar-sounding names — such as 'Cluff' and 'Clough'. Do you know of any?

A. The only such facility we know of for general-purpose use happens to be (a) cheap and (b) available from us. It's a feature called **Soundex** on the shareware program PC-FILE III, it costs just £15 in Apricot format and £10 in IBM format (all our prices include VAT and despatch), and you can order it from our DEALS FOR READERS service. ◆

Doodles by David

Tips and Tricks

The occasionally essential column

By David St.John Wallis

Is there anything which ungladdens the heart so much as a flat frog? Yea, verily, 'tis the sound of a hard disk grinding to a halt. This happened to me just after my move — and not, as I thought at the time, in consequence of thumps amid thunderstorms.

It turned out to be a loose earth wire to the Winchester. So if you get (A) screen flicker and (B) grinding Winchesters, pluck your courage and your screwdrivers in both hand and burrow into the bowels of the cpu. There alurking in the cat's cradle you will find a green wire appendixed to the hard disk. Tug it. If it comes orf in your hand, you may have enjoyed reading this article and saved yourself an huge bill. Occasionally essential? Quite.

Sparkle

Here's a thing to clear up some of the garbage on your hard disk. If you haven't got DOTOALL, apply now for DEALS FOR READERS disk AS010 (price £11.50).

CLOSE.BAT

```
CD\
DOTOALL DEL *.BAK
DOTOALL DEL *.BK!
DOTOALL DEL *.SAV
DOTOALL DEL *.PRN
DOTOALL DEL *.PIC
DOTOALL DEL TEST*.*
DOTOALL DEL TEMP*.*
DOTOALL DEL *.$*
A:
CD\
SAVEDATE
```

Scrub SAVEDATE if you want. It's a funny program anyway — mucks up dates.

By the way, a punter who phoned me said DOTOALL and KOPY don't work with BIOS 3.1. Well, they do with 2.11. Another punter wanted MS-DOS 3.2. Another, shredded beef in oyster sauce. Sigh ...

Spring fever

Brian who runs Coxhead & Welch in Grayshott sells a fine line in faery settee upholstery springs. Huge was his delight when I told him such springs fit a treat beneath the CONTROL key.

So you keep getting ^G or ^H or ^1? Lever off the key and install said Brian's spring — and save yourself grief with Apricot Computer Maintenance. Demuth Way repairs = Big Bucks; Brian's solution = 3p.

And avoid squirting everything in sight with WD40. For once, this miracle gunge is not the answer. Erratic clocks, snakes in your lakes, dead carburettor, sex problems ... WD40 is the gunge of choice. But not for Apricots.

Slowdown

Chaps keep telling me their disks are slowing down. And yes, they will after a bit. So CHKDSK shows lots of non-contiguous blocks and cross-linked files? You've been using your computer, you naughty thing. And the disks have got a bit full. And you've wiped some bits and written other bits back.

The clever FAT has been trying to keep track of all this garbage. And it has, too. But it just takes longer for the computer to find the bits of file it wants because you've scattered them about so much.

Solution: if you've got OPTIMISER, run it. If you haven't, back up everything and re-format the disk. If you can't be bothered, try copying the horrid discontiguous files on to a newly formatted floppy and then after a bit (like bits of days) copy them back to the hard disk. Don't use wild cards.

Elegance

In theory, anything entered into one computer should be accessible by another. The problem I had was to analyse, on Lotus among other things, Hungarian honey exports to Holland which had been prepared on an IBM with a Framework spreadsheet (which I didn't know) in Hungarian, in Budapest. How to do the job?

First, get the people in Hungary to translate the Hungarian spreadsheet into an English TXT file. Second, get the floppies. Third, get a very good friend with an IBM word processor to insert commas between the columns. You wind up with a file which looks like this:

```
Industry/Product HONEY Countries, Exports,
Imports, Code
Holland, 248934, 345293, 0
Canada, 4342480, 1926786, 0
U.S.A, 2069200, 7489290, 0
Austria, 157275, 212060, 0
Benelux, 458646, 270467, 0
Denmark, 26482, 288837, 1
```

... and so on. This then was e-mailed *chez moi* (that's another easy way of getting stuff from an IBM to an Apricot, by the way). Then, for some reason, my Lotus TRANSLATE program is always running out of memory although I have 960KB installed. So Sorcim's SDI was tried. I went for option B — that's comma-separated file into SuperCalc's CAL format. And lo! It worked.

Next, not bothering with SuperCalc itself, SDI was told to translate the file from CAL to DIF. Finally Lotus's TRANSLATE utility was used, and the DIF-to-WKS option was chosen. And it worked too.

Cheap fix

My Juki went peculiar too. Phone calls to Micro Peripherals were wont to produce replies like "Well, send it up to us mate". Not a lotta help if you've started a bonfire with the box and don't want to be without the thing for months.

So, there was this chap in the *Yellow Pages*. "Let me come and fix it," quoth he. "Let's meet up the boozer," quoth I, leery about mileage charges. Two days later and I had it back, happy and purring and working. So, if you're anywhere near Camberley with a duff printer — call **Peter Bradley**, a great and good man. His number is **0252 548254**. And no, I'm not in for a cut.

Caveat

I had to get that new PSU after all. The chap at Apricot Computer Maintenance said "Send us your old one and get the new one at half price". I got the new one and paid the full price — £115 something. They got the old one. Umpteen phone calls later they are still holding on to the other half of the price ...

The Prestel man

He gave me a call. "Woe!" He opined. "We've just spotted that you put your Prestel ID on your letterhead."

"And on compliment slips and business cards and anything else I can think of including hotel questionnaires asking me how I liked the room waiter."

"Why?" He queried.

"I have never ever had a message from anyone on Prestel and this is my tiny way of getting one."

"Has it ever worked?"

"It has not, kind sir. Please send one and that right speedily."

"Don't you know that we always advise all punters not to release their ID numbers. You should guard this as you would your daughter's virginity. Or your own. This is jolly secret. Why just the other day ..." And on and on he droned. An hour or two later (well, he was paying for the call) a small but insistent thread of triviality overcame me.

"Those of us hallowed few who actually access Prestel from time to time do have passwords, don't we?"

"Tush," quoth he, a nice man called Eddie Roach. "Any fool can break those."

Hmm. Seems to me the number of permutations is around 1.5 million. Any danged cretinous villain who wants to raid my box is welcome. Prince Philip, I'm not.

Anyway, the uproar of all this is that said Roach feels that Communiqué version 2.20 wasn't ever written properly, and that in order to access Prestel you have to tell the software something critical about upper and lower case. Funny. You can spend your whole life not worrying about things like this and you never had to do it before anyway. Quoth Eddie exsanguinously: "A phone call to Apricot's Hot Line at 021-501 2284 will put you right." Poor fool. Well, he wasn't to know what we know, was he? ◆

Contributors in this issue

Dennis Jarrett is Managing Editor of the TP Group newsletters and a frequent contributor.

Lindsay Doyle currently writes for APRICOT FILE on a variety of esoteric subjects. Over the years he has contributed over 75 articles to various publications. (Most of these are now defunct — is there a message here somewhere?) In addition to pounding away on an Apricot Xi of long standing, on which all the real work gets done, he is now playing with a new Amiga and HP plotter.

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.

David Gaunt is a research fellow in the microbiology department at University College, Cardiff. His computer interests include interfacing BBC Model Bs to lab equipment (and you though *Apricots* were infuriating).

Garret Keogh is the staff writer for TP Group, a computer enthusiast and general dogsbody; since coming to TP he has acquired more knowledge about desktop micros and envelope-stuffing than he ever thought possible — at the grand old age of nineteen he is adding to his computer education through a long process of trial and error(s).

David Langford is a full-time author and computer addict, whose latest book *EARTHDOOM!* — a spoof of disaster novels, written with John Grant and scorned by Kingsley Amis — is "by no means" about Apricot BIOS variants. (The inclusion of a permanently drunk character called Apricot should not be misinterpreted.)

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.

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 also has a sort of computer software company which doesn't do very well. He says that all contributions will be gracefully acknowledged.

Damien Thorne is a pseudonym for one of the first employees of ACT Advanced Technology.

Tudor Wynn-Williams spent seven years in Helsinki rewriting reports and publicity material written by Finns in English. Now back in Britain, he does similar work freelance, sending texts to Finland by electronic mail for direct setting.

Sifting through SuperWriter

Bugs and bodges

By Tudor Wynn-Williams

In two years' intensive use of SuperWriter I kept getting nasty surprises. It took a lot of effort to find ways of making SuperWriter print certain lines correctly, and I discovered so many discrepancies between the manual and the program that eventually I decided to go right through the manual checking all the features mentioned.

A file listing the errors I found runs to a couple of thousand words. You can have a copy of it, plus other goodies, for a nominal copying fee (see the box at the end of this *Workshop* for order details). Meanwhile, and for the impecunious, here are some of the 'features' you may not notice until it's too late.

Document History

Any author name entered for a document becomes the default author name for all subsequently-created documents and will be entered into their document histories. From then on you must have some kind of 'author' listed, even if it's only a space, since it is impossible to change an author name to less than a space.

Tab settings

If you set new tab positions in the Global Settings Menu, a tab is set automatically at every column to the right of the last tab *you* actually set. None of these extra unwanted tabs are shown on the Global Settings Menu screen.

The last position at which you can actually set a tab is with the cursor at the penultimate column, and this results in a line ending with three tab stops. On tabbing to these, you'll find that the next characters are typed at the penultimate character position, the last position, and the one just beyond the last column — producing an immediate word-wrap.

So there is no way tabs can be set with less than three tabs at the right margin. If you try to eliminate all tab settings by spacing all the way to the right margin, then leaving with Return, a tab is set instead at *every* column — though none are shown on the Global Settings Menu screen.

Tab positions (but not line width and visibilities set in the Global Settings Menu) become default values for a document created immediately after creating or editing one with altered tab settings. This means that if a document of line width 40 with specially set tabs is processed before one of line width 60, the second will automatically have a line ending with 22 tab stops in addition to those shown on the Global Settings Menu screen.

Finding text

Where two strings immediately follow one another, the second occurrence is ignored by the Veto choice of the Find option. The numbers of times 'M' is found in the following sequences in this case are ...

string	*found*
M	1
MM	1
MMM	2
MMMM	2
MMMMM	3
MMMMMM	3

Pressing 'Y' or Return stops the search in the Find, Veto option.

Indents

Negative paragraph indents, such as PI-5 and the like, are effective commands. But negative indents are not accepted by the program; they are treated as equivalent to IN 0. Also, positive indents are not summed: IN 20 following IN 10 produces a temporary left margin at character position 20 rather than 30.

External format

The External format for output retains boldfacing, underlining, italicising, superscript and subscript print directives. But, as with the System format, it converts print directives for centering, right-justification and hanging indentation to blank equivalents; page breaks become form feeds; and hyphenation remains as CTRL-^.

Format settings

Embedded formatting commands override any setting entered in the Print menu. They can be entered at any part of a document, but a Top Margin command will not apply to a page already being printed if it is less than the current Top Margin setting and a new line has already been started before the command is encountered.

The Left Margin, Line Width and Line Spacing may be changed anywhere, and will apply to all text following the previous new line set in formatting. Page Length and Bottom Margin may also be set anywhere, and they will apply to that page if they are consistent with what has already been printed on that page; if not, a line containing these commands will be printed normally, with no error messages, and the page will be ended at the end of the line with subsequent pages being printed according to these new embedded commands.

But even at the start of a document, the Top and Bottom Margins and Page Length must be set in an

order such that any command is not inconsistent with any previously read commands and default settings not yet overridden: TM0 must precede PL9.

Negative vertical increments do not move the paper upwards. The only negative increments permitted are those no greater than the current vertical increment, which may have been set through LPI: V-3 following LPI6 gives the same result as V5. Negative Vertical Increments are cumulative on the previous increment; positive are not, they are the increment actually applied.

You may have problems with an embedded command sequence such as OUT27,90 if you try to print it bidirectionally.

CNP

The rule for CNP is that a new page is started if there would otherwise be *n* or fewer lines in the last paragraph of the page, not fewer than *n* lines.

Ragged left margin

In proportional printing SuperWriter produces a flush right margin but the left margin is ragged: the letters at the left margin have their centres vertically aligned, so 'm' at the left margin extends further to the left than 'l'.

This problem can be solved by beginning every printed line with a hard space. You could insert these hard spaces one by one at the correct places, but it is much easier to replace all spaces with ...

```
\K-5\ \K-5\<HARD SPACE>
```

This does not affect the spacing between words in a line (substitute K-6 for K-5 if printing at CPI 10). SuperWriter ignores the final space that would otherwise end a printed line; so the following printed line begins with half a hard space, which produces a flush-left margin.

The Replace command does not recognise the word-wrapped space at the end of a screen line, however. So first replace all spaces with ...

```
QQ<RETURN>
```

This *does* find every space, as the next space to be replaced is always near the beginning of a line. Then you can replace all occurrences of QQ <RETURN> with the \K-5\ \K-5\<HARD SPACE> sequence.

A \K-5\<HARD SPACE> must also be added after every Return and hanging indent command to ensure that paragraphs start flush. You may also need to remove \K-5\ and <HARD SPACE> from lines with your original embedded formatting commands.

An execute file can do this whole procedure, which doesn't seem to slow printing; it does however increase file length roughly threefold — so files to be treated must be much smaller than the workspace capacity.

The Dictionary

There are far too many errors in the SuperWriter dictionary for me to list, as you can see by using the dictionary maintenance option supplied with Version 1.03 of SuperWriter that lets you list the words in a dictionary on screen or into a file. This shows that the SuperWriter dictionary gets off to a grand start with 'abada' as its second word. Considerable research has revealed that according to the full Oxford English Dictionary (I have found it listed in no other) 'abada' is an old and long-since obsolete name for the rhinoceros, so I would not myself have included it amongst the 20,000 most useful words in the English language; alternatively, since 'Addis' is listed later, perhaps they thought the capital of Ethiopia is 'Addis Abada'. That would not be too surprising as a little further on the plural of 'abbey' is given as 'abbies'.

Versions

My Apricot was sold with SuperWriter version D1.09. I have since tried out version 1.03, which was supplied with later Apricots and is said to be the newer and better version. Surprisingly, I found a number of additional bugs in 1.03 that are not in D1.09, which leaves me sceptical about the true relationship between these two versions. Needless to say, neither Apricot nor Computer Associates seem interested in clarifying things or correcting any of the bugs ...

In some cases the two versions will print the same file differently; and you may sometimes want to reprint old SuperWriter files. My general advice would be to stick to the version of SuperWriter you are already using. But for anyone considering a change, here are the main differences between the versions that I had already noticed before I decided to give up the idea of 'upgrading' to version 1.03 and returned to the devil I knew. Some of them might apply only to the Juki 6100 printer I use.

- **Paragraph indent:** in version 1.03 the command PI has no effect on a paragraph beginning immediately after it; at least one Return (in addition to the one ending the PI command) *must* follow PI before a paragraph is indented. And version 1.03 does not implement a PI command in a paragraph immediately following a page break set by the CONTROL-P print directive, although it does so if the page break is automatic.
- **Top margin:** a new Top Margin setting in version 1.03 does not always apply to the page it is printed on, even in cases where it *would* take effect in version D1.09. In version 1.03, TM0 heading the first page will apply to that page; TM17 heading the second page will apply to the second and third page; TM23 heading the third page will apply to the fourth page.
- **Printing:** in several respects, version 1.03 will not print some files correctly — nor will it

show their correct appearance in Format On-Screen. In particular:

- Embedded commands CPI and PROP ON to change control settings are ignored in Format On-screen but not in printing. This means that even page breaks may be shown in the wrong place in Format On-screen if the CPI and proportional control settings are changed with embedded formatting commands.
- If the first character of an indented paragraph is underlined, the underline will be shown as beginning at the left margin — *not* at the start of the paragraph.
- A Kerning command is correctly printed, but generates an error message in Format On-screen.
- A line following an SP0 command is not shown on screen; in version D1.09 it is shown, with the same line number as the previous line (both these 'lines' are in fact printed on the same line). The SP0 command operates identically in the two versions when it comes to printing.
- If LINE-13 follows printing of line 40, version 1.03 will move the paper up and print the next line on line 13, above the previously printed line, just as version D1.09 does; but in Format On-screen version 1.03 will incorrectly show this line as being printed at line 41. The next line is, however, correctly shown as line 14.

In all these cases version D1.09 correctly shows how the document will be printed.

- **Proportional printing:** version 1.03 is less convenient in that the control setting for proportional printing cannot be changed from N to Y between printings of documents without first returning to the main menu — trying to do this gives an incorrect 'Proportional Capability NOT Installed' error message. Also, when printing is complete, version 1.03 does not display a message to indicate this.
- **Default pitch:** if you do change versions, you should note that the default pitch is CPI10 in version 1.03 but CPI12 in D1.09. Changing the control setting for pitch in version 1.03 (but not D1.09) automatically changes the format settings for left margin and line width to maintain the same margins in inches.
- **Justification:** version 1.03 sometimes inserts extra spaces between words to indicate justification in Format On-screen; but if versions 1.03 and D1.09 are set identically, most lines are printed identically. Occasionally the two versions include different numbers of words in a line, however, and word-wrapping is done after different numbers of words; that applies at least with justified proportional printing. This means you cannot rely on an old document being reprinted identically by the other version of SuperWriter, which for me is sufficient reason for rejecting a change.
- **Bidirectional print:** at least with the Juki 6100 printer, version 1.03 supports only monodirectional print in proportional printing: the BI ON and BI OFF commands are ignored as errors. This makes printing much slower than with D1.09.
- **Dictionary:** version 1.03 is able to output the contents of a dictionary (even a D1.09 dictionary) as a file. This shows there are three fewer words in the version 1.03 dictionary: 'all-singing', 'SuperWriter' and 'unfamilar' (incorrectly spelled) are omitted. SW.DEF is supplied with only version 1.03, but SW.DEF files work with both versions.

Finale

SuperWriter has some very useful features for writers, such as the CONTROL-O P option for opening up space in a text for additions. It was also free with the Apricot and does not need extra memory.

As you may by now have gathered, however, I've found printing SuperWriter files can be very frustrating. I depend on a word processor for my living, and I decided I needed a better WP program. After thoroughly comparing the programs available, I bought WordPerfect and have been using intensively it for about a year. In all that time I have found no significant bugs, and the manual is very good. Contrast SuperWriter! I can heartily recommend WordPerfect to anyone thinking of upgrading. ◆

1987 TUDOR WYNN-WILLIAMS

The Bug File

For my full list of SuperWriter errors, send me a disk and £5 plus VAT copying fee (total £5.75):

Tudor Wynn-Williams
Bryn Elmen
Dolybont
Borth
Dyfed SY24 5LY
tel 097 081 375

For charge fee you will also get an Execute file to print a person's name and address on an envelope and in a letter, files with examples of errors SuperWriter makes in printing some lines and ways of getting these lines correctly printed, and printouts of the examples.

Winners and losers

Every year the US computer magazine *Datamation* publishes an analysis of the financial performance of the **data processing business**, based on the previous year's returns. For 1986 the DP industry as a whole continued its strong growth rate — total **turnover** up 17 per cent to nearly **$200 billion** — though **profitability** was hit by IBM's poor performance and the sticky progress of the dollar (overall, profits dropped by four per cent to just under $26 billion).

IBM remains the largest single company, of course, with 1986 sales of $49.6 billion. The Sperry/Burroughs merger puts **Unisys** into second place, a long way behind at $9.3 billion; then comes **DEC**, **Fujitsu**, **NEC** and **Hitachi** (the Japanese owe their prominence partly to the fact that *Datamation* uses dollar-conversion figures, and the yen has been particularly strong against the dollar).

The **microcomputer** subsection makes for interesting reading:

company	turnover	change
IBM (1)	$3,000m	+2.7%
Apple (2)	$1,781m	+11.1%
Olivetti (3)	$1,268m	+43.3%
Tandy (4)	$997m	+25.1%
Unisys	$800m	
NEC	$697m	+105.9%
Compaq (7)	$625m	+24.1%
AT&T	$600m	+20.0%
Toshiba	$582m	+100.7%
Zenith (10)	$548m	+55.7%

Out from the 1986 Top Ten go Commodore and Convergent Technologies (both had a very bad year, though Commodore appears to have turned round now) along with HP and Sperry (but Unisys now incorporates the Sperry figures and looks strong). The best performers in terms of growth rate are the Japanese again, with the yen-dollar exchange rate a major factor; Olivetti looks good, too, not only because the dollar has also been weak against European currencies.

The table obviously illustrates the strength of AT-type sales, particularly into corporate accounts — all the Top Ten companies sell 286- and 386-based machines as their main product lines.

Interestingly, profits on micro sales aren't good enough to get any of these hardware suppliers into the Top Ten **profit-makers** — though that list is headed by **Microsoft** (a net return on sales of 22.1 per cent) with **Lotus** at no 4 (17.1 per cent) and drive supplier **Seagate** at no 5 (12.7 per cent). **Computer Associates**, which is beefing up its mainframe software side as well as capitalising on SuperCalc, rates no 8 with an 11.5 per cent return. **Norsk Data**, which recently beat Apricot to the acquisition of Wordplex, happened to return 18.1 per cent on sales to take third position.

The SoapBox — Release 2

Tell us what you liked about the last year's APRICOT FILE (and what you didn't). 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:						
Disinformation	☐	☐	☐	☐	☐	
Doodles by David	☐	☐	☐	☐	☐	
Workshops and Technotes:						
Bandage & Plaster	☐	☐	☐	☐	☐	
SuperWriter	☐	☐	☐	☐	☐	
SuperCalc	☐	☐	☐	☐	☐	
MS-DOS	☐	☐	☐	☐	☐	
comms	☐	☐	☐	☐	☐	
other	☐	☐	☐	☐	☐	
Regulars						
News	☐	☐	☐	☐	☐	
Deals for Readers	☐	☐	☐	☐	☐	

Comms for business beginners

The **Advisory Committee on Telecommunications** has published another in its series of booklets about comms services. The title is *A Basic Guide to Telex, Electronic Mail and Fax Services for Small Businesses*, it's free, and the 21 pages provide a quick 'n' easy intro to the methods available for transmitting written information quickly over Telex, e-mail and fax networks. There's some discussion of their relative merits and a few words of advice about finding the service best suited to a particular business's needs. Call **01-353 4020.**

Tax forms by computer

The **Inland Revenue** has announced that it's prepared to accept computer-produced tax forms from tax year 1988/89 onwards. The forms have to provide the same information as the official returns, of course, and you'd be well advised to check with your local tax office (maybe the supplier of your nominal ledger package too) before throwing away all the Revenue's bumf.

Apricot gasman cometh

Apricot dealer **Megabyte** has clocked up a £300,000 LAN contract from the **Gas Consumers Council**. A dozen regional offices will get a 286-based file server with two or three workstations and a laser printer; head office will have a 300MB 386-based file server, eleven workstations, and a pair of laser printers. The Council said they liked the price, speed, "overall capability" and easy upgrading; Apricot couldn't have put it better. Megabyte is providing a lot of bespoke software for the Council's complaints-handling activities.

Online multilingual dictionary

Microlink, the information facility which operates inside Telecom Gold (and which has taken to describing itself as the UK's fastest growing electronic mail service), is now offering its subscribers access to the Great European Dictionary — which claims to be the world's biggest (and most up-to-date) multilingual dictionary. It runs on an EEC-sponsored mainframe in Luxembourg, represents a massive collaboration of language experts in the EEC, and consists of more than 380,000 words and phrases in English, German, Danish, Dutch, French, Italian, Portuguese and Spanish. It offers translations from any one of these languages into any other — or into more than one at the same time; and it provides explanations of the word with notes on its use. Details from Microlink on **061-456 8383.**

Multisoft for Compsoft

The proposed reverse takeover whereby **Sagesoft** would acquire **Compsoft** and a back-door Stock Exchange listing has collapsed. The two companies apparently failed to agree on terms, and it looks as though accounting package specialist **Multisoft** is Compsoft's new suitor — Compsoft's MS-DOS BookMaker package at the low end of the market would fit well with Multisoft's Unix products. Compsoft made a slight loss on last year's £2.2 million turnover, Multisoft claims profits of around £120 million on sales of £3 million.

Sagesoft, by contrast, is anticipating profits of £1.2 million on sales of £3.3 million for its year ending 30 September, and MD David Goldman has said he expects a USM flotation "probably within the next six months". Recent market research figures from **Context** give Sagesoft a 27.7 per cent by volume of the market for accounting software, with Sage Accountant also figuring as the top-selling software package for micros in the first quarter (7.4 per cent of all sales, ahead of Lotus 1-2-3 at 7.2 per cent.)

Quest profits

At the **Quest International** group, turnover for the year to February 1987 was up from £11.4 million to **£28.8 million.** Profits also went through the roof, rising from £225,000 to £2.9 million.

New MD for Apricot Xenix

Over the last 18 months, Apricot has put a lot of time and money into Xenix. It has now set up a Xenix division under what looks like a well-qualified managing director — **Peter Thomas** formerly ran sales at Rair, and was previously UK general manager for Altos. His Official Comment includes the view that 'Apricot has the products and resources to lead the 3- to 24-user Xenix marketplace".

Take AIM

Hull-based legal systems supplier **AIM** has been bought by its management from its parent, a food company called William Jackson. AIM's founder Clive Telfer and four other existing directors now own the company, promising a USM quotation and complementary acquisitions in the near future.

Facts on fax

Did you know that as of March 1987 there were **100,000 facsimile terminals** in the UK — and within the next 12 months that is expected to grow to 170,000? So says British Telecom, which will be publishing the new edition of its *Fax Book* this autumn. Compared with the 22,000 entries in the 1987 version, this one is going to be a biggie — 40,000 or more entries — and if you want to be included, call **Freefone Fax Directory.**

Computers for PR

Independent **PR consultancies** aren't as *au fait* with computing as the in-house public relations departments, says a report from the PR services company **PNA** which surveyed just over a hundred respondents. Only 61 per cent of the independents have one or more micros, against 81 per cent of the departments. And the usage of electronic mail is similarly marked — 38 per cent against 16. ◆

Showtime

We've emptied out the piggy-bank to take a tiny last-minute stand at the *Personal Computer World Show*, to be held at London's Olympia on 23-27 September. We look forward to welcoming old and new friends at Stand no. 5208 on Level Two in the Business Hall.