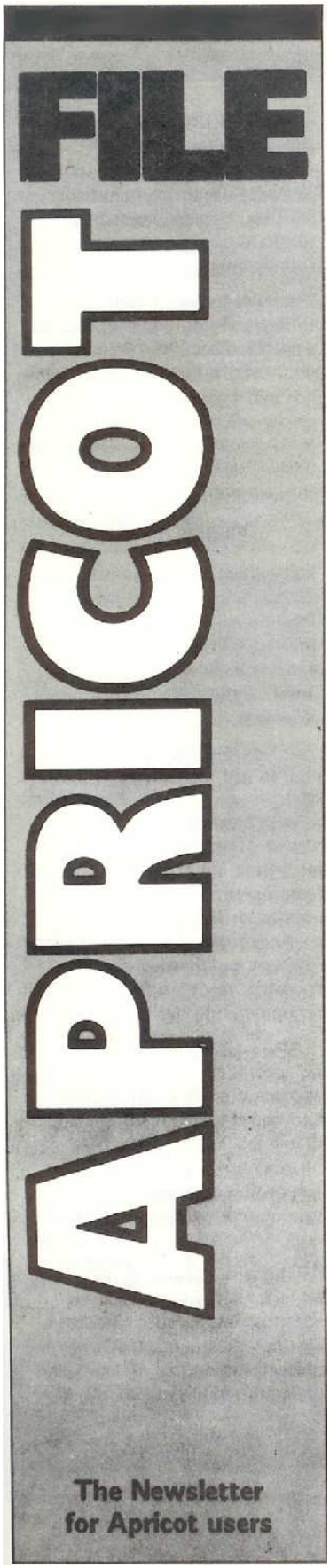
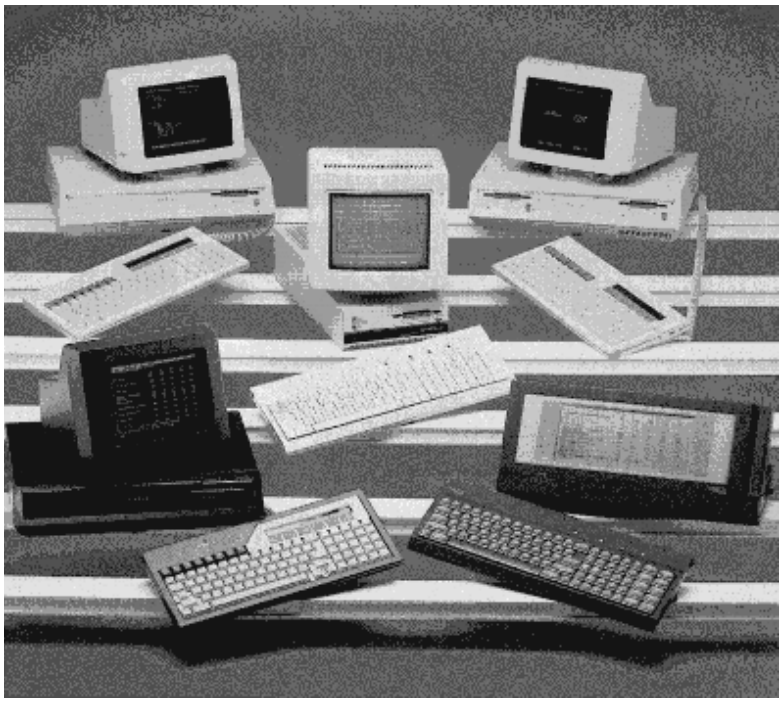


Apricot File 1_2



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 1.2

November 1985

FILE APRICOT

The Newsletter for Apricot users

Apricot's market share

by Henry Deckhand

A couple of weeks ago the SUNDAY TIMES published a piece by its computer correspondent **Jane Bird** which reported the conclusions of a market research survey by **Wharton Consultants**. This suggested that Apricot's share of the UK market was dropping like a stone — indeed, that August sales gave Apricot a mere four per cent of the UK's total shipments of personal computers.

This had all been reported the previous week in the computer press. But it's a measure of the relative importance of Britain's publications that the SUNDAY TIMES article caused all the furore.

Apricot's share price was already wobbling from two weeks' of bad news, sparked off by the group's unnecessary but fairly open-handed announcement that its half-year figures would be distinctly iffy thanks primarily to inventory write-offs on the Portable and a slow (not to say undetectable) start-up in the States and mainland Europe.

But shareholders and brokers apparently believe the SUNDAY TIMES, for the following Monday saw a big slide in the share price (it touched 56p at one point, having been in the low 70s the previous week ... and 280p earlier in the year). And on Monday Apricot was bombarded with phone calls — 250 at least, we're told — from worried stockbrokers, investors, dealers, big corporate buyers, and Uncle Tom Cobbley's mates.

Now, it's clear that Wharton's figures are just plain wrong. Apricot itself claims to have shifted 1,700 units in August, out of a guesstimated total of 12,000 PC shipments that month.

You don't just have to listen to Apricot's own people ("completely at odds with the findings of recognised research organisations ... based on totally invalid research ... ridiculous ... if it had been anywhere but the SUNDAY TIMES the article would have been treated with the contempt it deserves"). The figures from two other high-profile market researchers in the dealer game, **Romtec** and **Context**, both give what feels like a more realistic picture — Romtec has Apricot at 17 per cent of the total sales for August, Context offers 20.4 per cent. The **National Computing Centre** variously quotes 20 per cent (its director) and 30 per cent (a bloke from its Microsystems Centre operation).

So where did Wharton's four per cent come from? Well, Wharton doesn't poll many dealers to get its monthly sales figures: it has a panel of 250 and calls a hundred of them each month. Romtec and Context both claim to poll at least 200 a month.

▶▶▶ CONTINUED ON PAGE 12

In this issue

Six Fortran compilers — group test 3
The Disinformation Column 10
Writers in this issue 13
Polyprint reviewed 14
Deals for Readers 18
GSX Workshop: part two 21
Readers' writes 23
Software round-up 24
Q&A Service 26
Free Classified Ads 27

About the newsletter

Our aim is to provide a complete editorial and information service to users of the Apricot business 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 or any of its dealers. Nor for that matter are we sponsored by or directly associated with any user group or any other such organization, though in principle our goals and guiding intentions are very similar to the declared aims of many such bodies.

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.

Essential disclaimer: The Publishers do not necessarily agree with the views expressed by contributors, and assume no responsibility for errors of reproduction or interpretation in the subject matter of this magazine or from any results arising therefrom.

Copyright, permissions and reprints

Unless otherwise indicated, the copyright on all contents of APRICOT FILE is held by the Publishers or by the Publishers and Authors jointly.

Material published in APRICOT FILE may not be reproduced in any form without the express permission of the Publisher, except for short excerpts quoted for the purposes of review or reference and duly credited.

Ordinarily though we will accede willingly to any requests for occasional photocopying, particularly as there isn't much we can do to enforce copyright in those cases: and we will generally allow bulk reproduction of articles for educational purposes, asking only that the copy includes a credit to APRICOT FILE.

But in other cases we will enforce our copyright protection against unauthorised reproduction. If you need a number of copies of a particular article, why not ask us to quote for a reprint?

Contact points

You can reach us for queries on **Editorial, Subscriptions,** and **Deals for Readers** at:

APRICOT FILE
TP GROUP
PO Box 509
London N1 1YL
England

Telephone **01-609 3800**

Electronic mail
Telecom Gold 81:JET767
Easylink 19011170
The Source BDE566

Telex **265871 MONREF G**
quoting reference 81:JET767

Subscribe!

You may have received this copy of APRICOT FILE in one of our promotional mailings; or you may be reading someone else's copy. Either way we hope you enjoy the newsletter and find it useful.

But if that is the case, to receive any further copies you'll have to subscribe: this publication is not distributed free, and it is not available on the newsstands.

The subscription rates: The annual subscription to APRICOT FILE buys 12 monthly issues at the following rates:

delivery in ...	
Britain	**£35.00**
Rest of Europe	**£38.00**
Elsewhere (air mail)	**£50.00**

Payment: Your subscription order should be accompanied by payment in Sterling, with cheques and money orders to be payable to 'TP GROUP'. We can only accept prepayments, because we're small and struggling: and unfortunately we can only accept payments in Sterling, because our bank surcharges us for processing foreign money.

The Guarantee: 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.

Write for us!

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

If you have something you'd like to say, contact the Editor first to talk about it. Thereafter articles can be submitted in two ways — transmitted via electronic mail (Micromail/Telecom Gold, The Source, or Easylink): or in word processor format on disk (we can handle most WP formats, but it's safest to send an accompanying hardcopy too).

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

Unless otherwise agreed, editorial submissions will be accepted on an 'all rights' basis. On the other hand, that's just to protect you and us: if you want to retain copyright, say so. ◆

ISSN 0268-6066

Group Test

Fortran on the Apricot

by Edward N Bromhead

Read your *Apricot Owners Manual* and you will find Fortran described as a programming language designed principally for scientists and engineers. Elsewhere the booklet points out the mathematical and number-handling facilities of Fortran, and says that its main use is to transfer existing programs from big computers to small ones.

This isn't strictly true — rather it's the case that most large scientific and engineering programs for mainframe computers are written in Fortran as a result of a historical accident. There is therefore a market for software that allows the simple transfer of these programs to computers like the Apricot, computers which in fact are larger in capacity and faster than most of the mainframes of a few years ago.

So if like me you're a scientist or engineer with a major investment of time and effort in a number of programs for your own work, but you now want the convenience of a desktop personal computer like the Apricot, how do you go about getting your Fortran programs to work on it? What are the costs? The pitfalls?

My own first contact with an Apricot was in the window of Debenhams in Staines just after the launch. A quick calculation of what 256K bytes costs on the computer I was using at that time, a peep at the price list, a check that the software list which included a Fortran compiler ... and I had convinced myself I needed one.

Within a week I was working on it, and now I have two: one in the office and one at home!

This article is the result of a year-and-a-half long grappling with the machines and programming them. During that time I discovered a number of fundamental errors in my original approach — not least of which was a major misapprehension about those 256K bytes and how much of that space was either free to use, or readily accessible to my programs.

Working with the Apricot

For the mainframe user there's likely to be only one compiler for a particular language: on a micro you're embarrassed for choice. This article investigates Fortran on the Apricot with a comparison of several different compilers. They all have strengths and weaknesses, and all cost much the same — too much to lash out on spec, but not so much that if you need/want one it is out of reach!

The first step is to transfer any source code from the mainframe to the micro. That's easy if the terminals on the mainframe use RS232 connectors: with the freebie ASYNC on the Apricot, and the appropriate lead, you're in business. Where some other comms system is in use, don't despair. But you may need expert help (and possibly wads of money too).

Don't worry about space on a disk for your program: even a single-sided microfloppy can easily cope with the source code for a huge program

Compiler versus interpreter

The great advantage of using a *compiler* is that the conversion from your high-level programming language source code to the instructions which a computer can obey is a once-and-for-all process. You then have an already-converted program which can be used time and time again with no overhead for conversion.

Another advantage of compilation is that the original program is secure from tampering to a much greater extent than is the source code.

But a compiler is inefficient for development purposes. Before even the tiniest modification can be tested, most or all of the compilation process must be redone.

Basic can be *interpreted* — that is, it can be obeyed step by step without the translation of the whole code at the beginning. It can therefore be run almost instantly after a modification for test purposes; and when satisfactory, that code can be compiled. This probably accounts for the popularity of Basic.

But run-time speeds for interpreted programs are reduced if the code contains many comments; so many programmers leave them out. And the language structure is condusive to messy programming with many patches.

What's more, much engineering software already exists: and it tends to be in Fortran. As far as I know, Fortran cannot be interpreted; and even if you're attracted by virtues of interpreted languages, the conversion of a working Fortran program of any length into Basic will prove an error-prone and unrewarding task.

Some Fortran compilers do have as an option the ability to use a *symbolic debugger*. This enables the processor to track or step through the compiler code, and to relate back to the symbology (the variable names and program structures) of the original source code — useful to the developer, but it makes the compiled code larger and slower in execution.

The choice

Microsoft's **MS-Fortran** is the compiler you'll probably get from your friendly ACT/Apricot Ltd dealer if you just say you want a Fortran compiler.

There's also the **UCSD p-System Fortran**. That can be obtained at a slightly reduced price from TDI using the voucher which came with early Apricot shipments, although these days you'll have to pay the full price for the operating system — which makes this route to Fortran a dear one.

Then there are compilers from Supersoft (**SSS Fortran**) and Prospero Software (**Pro Fortran**); Digital Research has **DR Fortran**; and Ryan-Macfarland's **RM-Fortran** is just about ready for the Apricot.

Four Fortrans

Fortran has two basic standards: laid down by the American standards organisation ANSI in 1966 and 1977, they are known as Fortran 66 (or Fortran IV) and Fortran 77 respectively.

To make it more complicated, each standard has both a full language specification and a subset, with the latter missing out a number of the more advanced features of the full language.

Both of the Fortran 66 compilers — SSS Fortran and Pro Fortran — are full implementations: two of the Fortran 77 compilers implement the subset only (MS-Fortran and p-System Fortran) and two have the full standard (DR and RM-Fortran).

There are really only two differences between a full Fortran 66 implementation and the subset of Fortran 77 ... but they are enough to upset many a well-designed and proven program. These are the areas in which Fortran 77's facilities are a *replacement* of those in Fortran 66 and not just an enhancement: handling of character data, and the semantics of DO loops.

The character-handling is the more difficult area. Programs which do a great deal of it using Fortran 66 conventions are very difficult to convert, and it is more cost-effective to obtain a Fortran 66 compiler than to spend a lot of time in programming a conversion.

Fortran 77 (even the subset) does have some *extensions* to the facilities of the earlier standard, notably in the area of structured control of the logic of a program — useful when programming from scratch, but not necessarily so useful when considering putting existing software on a micro.

Supersoft's SSS Fortran is based on concepts developed at the University of Waterloo in Ontario for WATFOR and WATFIV — compilers used widely on IBM mainframes in the 1960s and 70s. If you're used to those, you would find this compiler just up your street; if not, some of the things it does might not be entirely to your liking ...

What you get

When you buy a compiler, that isn't the end of it. You need some other facilities as well. These include a **text editor** for creating and amending program and data files: a **linker** to join up the compiled code with a library of standard routines: the **routine library** itself: and ideally a **librarian utility** to add routines to the library, or even create new libraries.

So what you get with a compiler is one key factor in deciding whether it's a good buy — or rather, what you *don't* get and will have to buy separately.

The p-System really scores here. With it you get all of those facilities. And they're properly integrated into the system: so after a test compilation in which the compiler has detected an error, it's simple to summon the editor and jump direct to the offending line in the program. That is not a facility offered by any of the others.

With them you could use EDLIN for editing: or any word processor will do. There's SuperWriter (if you got it for free); I use WordStar as my text editor, but that's because I've got used to it rather than any intrinsic merits.

With MS-Fortran a linker called MS-LINK is incorporated with the package. But there isn't a librarian — for that you'll have to buy the Assembler package at a further £95. The same goes RM-Fortran, but for SSS Fortran you need the Assembler package for MS LINK as well. Pro Fortran has its own versions of these utilities. The Digital Research compiler is complete with DR's LINK86 and LIB86. RM-Fortran includes a recent version of MS-LINK (version 2.50) but no librarian.

Performance

When a program has been compiled it should execute very rapidly. Any difference in speed should be attributable to the minor differences between the compilers and their libraries of standard routines.

The p-System compiler falls down here. The result of a p-System compilation is a set of p-Codes: these must be interpreted into machine code at run time. Execution of complicated mathematics is therefore appallingly slow.

To overcome this there's a 'native code generator' (at extra cost) to convert compute-bound sections of the program directly to 8086 instructions. Or you can use the recently-released facility which enables p-System programs to run under MS DOS. The former enlarges the code significantly, which the p-System does not like very much; and the latter means you have to cope with two operating systems.

The other compilers perform various degrees of optimisation in the translation process; and since 8086 code is generated, all run much faster.

The 8087 arithmetic processor option can make an enormous difference to the computation speed, but the chip won't actually be used by the compiler

at all unless the appropriate library is included in the link stage — except in MS-Fortran, where software emulation routines first test for the presence of an 8087 and use it if one is found. Linking with the non-8087 library carries a penalty of 4 to 8K bytes of extra code, and so for obvious reasons the reverse procedure won't work.

Libraries

Each of the compilers has at least one and possibly as many as four separate libraries of standard routines. These normally include all the standard or intrinsic functions defined in the relevant Fortran specification, plus one or two extras. The compilers reviewed have at least two libraries, one including the code to drive the 8087 maths co-processor and the other to drives a software emulation on the 8086. You may find yourself paying extra for this 8087 support.

The SSS Fortran compiler has a large number of additional routines for useful functions like accessing the clock and handling strings; there are also Basic-like PEEK and POKE functions which most of the others don't have.

Pro Fortran shares the more useful of these functions, but in a smaller set.

The additional libraries are used where a compact program is to be produced through the use of the so-called 'small model'. Only DR Fortran and Pro Fortran can do this, and they can therefore produce short code for simple programs.

Precision

So Fortran is strong on precision, is it? Well, on the Apricot the p-System compiler is unique in that it can set the basic precision of real calculation to eight bytes, four, or a lesser precision. All real operations are then done in accordance with this preset precision choice. The p-System operating system has to be reconfigured to change it, however, which can be tedious.

All of the other compilers reviewed allow you to select the precision with which real data is stored. This is normally at the four-byte or eight-byte level, the precision of which are respectively about 7 and 16 decimal digits.

In principle there is a trade-off between the use of individual bits in the 4 or 8 bytes for extra precision (ie for more decimal places) and the capability to deal with larger overall numbers. A happy compromise on precision, maximum number size, and economy of use of RAM would be feasible if there was a six-byte real number (48 bits): but microcomputer compiler writers seem to have neglected that possibility ...

The DR Fortran is unique in offering 10-byte real arithmetic. MS-Fortran includes facilities to set round-off and precision on numerical operations 'on the fly', though this does affect four-byte operations.

Integer quantities can be stored with absolute precision, but with a size which depends on the number of bytes allocated to them. The compilers all differ in their provision for this; but broadly speaking all allow two-byte integers (in the range -32767 to +32767) and some permit four-byte integers (up to about 21 million).

As befits a serious (and relatively expensive) compiler, RM Fortran only provides for four-byte integer arithmetic: memory savings with smaller integer sizes are usually small anyway.

As with real data types, the DR product supports a wider range of precisions in integer precision — one-, two-, four- and eight-byte integer data types.

A major disappointment with most of the compilers comes from the way they address the available data space. Memory on 8086-based computers is subdivided into 64K byte blocks, the address of any of which can be accessed relative to its origin with a two-byte ('short') integer. To find the start address of such a block, two such short integers ('i' and 'j') are needed; and the address is then computed from 16*i + j. (Actually, using the 32 bits rationally would allow the addressing of four *gigabytes* rather than the 896KB accessible using this method.)

A compiled program might need to address a memory location more than 64K bytes away from the currently held address — to pick up some data, or to jump to if the logical flow of the program demands it. Obviously there are advantages if there are no jumps of more than 64KB, if both code and data each are contained within a 64K byte 'segment', because 'short' addresses can be used and the program becomes both smaller and quicker.

Such a program is said to use the 'small model'. One which uses what is effectively 20-bit addressing uses the 'large model'.

The compilers all differ in the way they use these two models. The p-System disregards the memory on the Apricot in excess of 128K (though that can be used as RAM disk).

Microsoft's compiler puts all working data in one 64K block — including blank COMMON — but it allows the use of extra memory provided that it is accessed by means of named COMMON blocks of up to 64KB each. Presumably, the COMMON block name reference holds a long integer address for the start of the COMMON block. A large-model approach is used for code.

Supersoft's compiler works in a different way, one that I haven't quite got to the bottom of yet. It looks as though the start address of all data items must lie within a 64KB segment: but since a short integer can be used to address any element of it (only two-byte integers are available for the programmer's use) arrays of from 1 to 32767 can

be utilised, effectively extending the top end of the data space. Needless to say, there's absolutely no documentation on such matters. The whole of memory is available via the PEEK/POKE facilities, however; and a little code in the form of 'put' and 'get' subroutines opens up a range of facilities.

A compiler option within both the Pro Fortran and DR Fortran compilers allows you to select the small or large model. For quick-and-dirty programs, this is something of an advantage — though most of the time the large model will be used just to access more of the data space. (Since nearly half of the standard 256K memory is occupied by operating system, there isn't room for much more than a small model program. But a program could be long on code and short on data space — or, more usually, short on code and long on data. A large model program, with perhaps as much as 30KB extra overhead, would then be required anyway).

The interpretation of the large model also varies. Pro Fortran sticks to 64KB COMMON blocks (except for blank COMMON, which can fill the remaining free memory); the DR and RM Fortrans allow greater flexibility with arrays of up to 64K *elements*.

Linking

Linking is the procedure whereby different compiled parts of a program are brought together. These modules comprise separate subroutines and functions; they may be compiled individually or all together (the Microsoft compiler is the weakest here — it has a small capacity) and they then need to be joined with system code.

The 'linker' or 'link-editor' that does this is supplied with the Microsoft, DR and Prospero packages — respectively MS-LINK, LINK-86, and PROLINK. Overlaid programs need special attention at link time, and of these, only MS-LINK cannot create overlaid programs. (*Real* programmers would probably dump these, and invest in Phoenix's PLINK to complement their PMATE text editors ...)

As an aside, it comes as a real blow to discover that the documentation for MS-LINK is better in the Supersoft manual (for a package which doesn't include it) than in ACT's Microsoft Fortran manuals.

Bugs and problems

You might think that a compiler would be relatively bug-free. Not so! My Microsoft compiler hangs up sometimes when parsing strings such as 14E15 whether this is part of a format or is an exponential-type constant — but only when run on a machine equippped with an 8087 chip. At times I have had similar troubles with the others, too.

The Microsoft product also is not too keen on long source code files. Once it has filled the available cross-reference space internally, it just hangs up. These programs have to be broken down into shorter units to be compiled separately.

Supersoft's compiler does this for you, making a separate .OBJ file for each subprogram, and conveniently writing a response file listing the names of these for use with the linker. It seems happier with the longer source files.

Perhaps a worse problem with the Microsoft Fortran is the a that probably relates in some way to the Apricot's BIOS: under certain conditions it causes programs using the 8087 chip to hang mysteriously. This seems to be the incorrect setting or cancelling of an 8087 interrupt mask: running the program from WordStar (or with certain early versions of the BIOS) turned out to be the only ways to get results. Who said WordStar was unfriendly? Using CCPM's PC-MODE avoids the problem too, but you need lots of extra RAM.

(All this might seem like a dig at MS-Fortran. It isn't really, it's just that I have more experience with that compiler.)

All of the compilers have undocumented features which can be found — sometimes by accident, sometimes by diligent trial. In particular, compile and run-time error messages can be inadequate: documentation may be perfunctory or misleading: the compiler's behaviour can be unpredictable. Too often manual writers assume a detailed knowledge of Assembler programming or operating system intrinsics. In my view this is neither necessary nor desirable for the user of a high-level language ...

Getting hold of bug fixes and enhancements isn't easy either. My original dealer gives no support: no-one there will answer correspondence, return phone calls or do any of the things for which a dealer is supposed to need the full RRP profit margins. ACT as was (Apricot Ltd now) doesn't support end users, although it did upgrade my MS-Fortran to version 3.13 (the current release number is a lot higher than that) when I couldn't get anything from my (then) dealer ...

Disk space

Some of the compilers are very large programs in their own right, and occupy a great deal of disk space. It might be necessary therefore to swap disks during a compile and link operation. On the whole, though, a set of two single-sided working disks is enough to contain the operating system and all the necessary files — which is by no means all those supplied with the software.

When using MS-LINK on a 256K Apricot for a largish program, extra disk space is needed by the linker for writing temporary files. It will crash if there isn't enough memory.

Speed of compilation and execution

Some compilers claim high speed. That's rubbish. All of these compilers are 'multi-pass'; they run through the code at least twice *before* they go through the linking process. What one does quickly, another does slowly.

Anyway, what counts isn't the *compile* time: it's the *run* time for an already-developed program. So which executes fastest?

Well, except for p-System they're all much the same. In a number of benchmarks MS-Fortran came out as slightly faster than Prospero's Pro-Fortran which in turn bettered the Supersoft SSS Fortran — but scarcely noticeably so. In the same tests, the Prospero code was shorter than Microsoft's and substantially smaller than Supersoft's.

RM Fortran was able to give a code which ran fastest of all, but to do this needed a whopping 70KB more than the biggest of the others!

On the other hand, the test source program was running in ten minutes with the Supersoft compiler: it needed several hour's work on MS-Fortran: and it took all day with Prospero's package! That was my fault, and it just goes to show how misleading some comparisons could be. All of the compilers needed the large model, and a bit of juggling with the code could have got it down to small model size for Pro Fortran. Then the Prospero compiler would have won hands down.

With any compiler, run times can be reduced in several ways:

- use small-model code for an improvement of 50 to 100 per cent
- use the 8087 to run 200-300 per cent faster
- write Fortran source efficiently. Only small changes may be required, but these differ from compiler to compiler as side effects of the optimisation procedure used. That should produce a 50 per cent gain.

These compare to about a maximum run-time difference of 15 to 20 per cent between different compilers for programs without any fine tuning.

Restrictions and extensions

There is no such thing as a *standard* Fortran compiler. All of the compilers considered here have one or more extensions and restrictions.

The major restriction occurs with the Fortran-77 compilers where only the subset is supported: the major extensions are the provision of 'Include' statements, and — in the Fortran-66 compilers — the use of character variables, implicit typing, and the definition of numeric variable precision by specifying the number of bytes per variable (as in REAL*8). The restrictions are listed in the comparison table.

Part of the rationale for the change to Fortran 77 is that it contains facilities for writing more structured programs. Supersoft SSS Fortran has a preprocessor called Ratfor (*Rat*ional *For*tran) to translate a structured code into standard Supersoft code. This is distributed in source form, and is therefore available to be modified to translate Fortran 77 to Fortran 66. I cannot see why Supersoft hasn't done it already!

Best Buy?

If I lived in remote isolation from the real world of MS-DOS dominance: if I wanted to be a program developer with no connection to past work: if there was no need to worry about execution speed: and if I could be happy that no-one else would ever want to run my programs on their Apricots ... I would opt for the **UCSD p-System Fortran** because it is so complete. But in those circumstances I probably wouldn't be using Fortran anyway.

And in the real world I couldn't recommend that anyone except a p-System enthusiast go this way. I regard my own investment in p-System Fortran as a waste of money, best written off against experience ...

The Microsoft **MS-Fortran** compiler is highly compatible with MS-DOS, as might be expected, and it works straight out of the box. In addition there's the interface to GSX (which the DR compiler also provides).

Still, this compiler annoys in the things it leaves wanting. Not that most people want complex arithmetic: but without a Block Data facility you cannot initialise the data values in COMMON blocks, you can't use large arrays, and without the string function LEN passing character data is tedious. You don't want code that is iffy on whether it runs or not, either!

SSS Fortran by Supersoft allows easier access to long arrays than does MS-Fortran (provided there's only one or two of them!), and it has a more complete list of character-handling and general utility routines. It's irritating though that you need to purchase several packages.

I wouldn't bother with Supersoft's **Ratfor** unless I wanted to use Fortran 77 features, and then I would customise the package to translate Fortran 77 features into their equivalent in Fortran 66. This would inlude more conventional file manipulation statements, and a more conventional IF ... THEN ... ELSE structure than you get in the somewhat idiosyncratic Ratfor. On the other hand, if I had a lot of experience of WATFOR or WATFIV, this is probably the one I'd plump for.

Pro Fortran by Prospero has broad similarities to the Supersoft product, resulting largely from its espousal of the 66 standard. It is a complete package with linker and librarian. The package is compact, and runs as documented.

In fact Pro Fortran offers what is probably the best value for money, if only because its developers are UK-based; they can and do respond quickly to their users.

When I tried it on a friend's machine, however, I really couldn't get on with it — in retrospect, this was probably because I'd got too hooked on the use of Fortran 77 features to want to go back to the 66 standard. Still, Prospero is rumoured to have a Fortran 77 compiler on the stocks: this could be worth waiting for.

To get the best out of the Ryan-Macfarland **RM-Fortran** a 512K Apricot is essential (it's merely desirable in other cases). You'll also need additional utilities, probably in the form of a librarian. You wouldn't be disappointed in this package: but you would be a lot poorer than if you bought any other, particularly as you cannot produce a non-8087 program — so straightaway you have about £700 worth of extra hardware to buy!

DR Fortran from Digital Research compiler is a relatively new product. It has a lot of friends already, though, and the latest release (4.1) has considerable polish. Many of the MS-Fortran niggles have been absent from this one from its inception.

Personally, I like DR software: all my own programs compiled for an MS-DOS environment run quite happily and as fast under CCPM — and this one is my favourite.

Last words

Any engineer or scientist with a Fortran background could easily come to terms with only one of these compilers.

On the other hand, running a program through a different compiler can often give additional insight into the structure of his program; and with a judicious choice of software, particularly if the generated object modules are sufficiently compatible to enable them to be linked together, some extra facilities can be obtained. This will cause a larger overhead as there will be some duplication of library routines from the two compilers. ◆

Suppliers

□ **Digital Research DR Fortran**

Apricot UK Ltd
Shenstone House
Dudley Road
Halesowen
West Midlands B63 3NT
tel 021-501 2284

Most ACT/Apricot dealers can supply this compiler; it's starting to appear in Apricot format from independents too.

□ **Microsoft MS-Fortran**

Apricot UK Ltd
[address as above]

Again, most ACT/Apricot dealers will be able to supply this one: it's also available from other sources — notably independent distributors. This is one that has the Apricot format:

Telesystems Ltd
3 Wycombe Road
Prestwood
Bucks HP16 0ND
tel 024065 6365

□ **Prospero Pro Fortran**

Prospero Software
190 Castlenau
London SW13 9DH
tel 01-741 8531

□ **Ryan-Macfarland RM-Fortran**

Ryan-Macfarland Corporation
Crown House
Turners Hill
Cheshunt
Waltham Cross
Herts
tel 0992 24981

□ **Supersoft SSS Fortran**

UK availability of the (US-made) Supersoft products has been a bit erratic of late, partly because Supersoft doesn't have an exclusive distributor here. These two say they can supply Apricot format:

Raven Computers
28-32 Cheapside
Bradford
West Yorks BD1 4JA
tel. 0274 309386

Microcomputer Products International
Central House
Barking
Essex IG11 8NT
tel. 01-591 6511

□ **UCSD p-System Fortran**

TDI Ltd
29 Alma Vale Road
Bristol BS8 2HL
tel 0272 742796

TDI is the UK agent for all UCSD p-System products.

Facilities

Manufacturer	UCSD	Microsoft	Digital Research	Prospero Software	Supersoft	Ryan-Macfarland
Compiler	p-System Fortran	MS-Fortran	DR Fortran	Pro Fortran	SSS Fortran	RM-Fortran
Origin	USA	USA	USA	UK	USA	USA
Fortran standard	77 subset	77 subset	77 full	66 full	66 full	77 full
Linker	yes	yes	yes	yes	no	yes
Librarian	yes	no	no	yes	no	no
8087 support	extra cost	yes	yes	yes	extra cost	needs chip
Block data	yes	no *[note 1]*	yes	yes	yes	yes
Core-to-core i/o [CTC] or [ED]	CTC	CTC	CTC	ED	CTC	ED
Character type	yes	yes	yes	in string literals	yes	yes
Hollerith type	in format	in format	in format	yes	yes	yes
Real types:						
— 4 byte	yes	yes	yes	yes	yes	yes
— 8 byte	yes	yes	yes	yes	yes	yes
— 10 byte	no	no	yes	no	no	no
Integer types:						
— 1 byte	no	no	yes	yes	yes	no
— 2 byte	yes	yes	yes	yes	yes	no
— 4 byte	no	yes	no	no	no	yes
— 8 byte	no	no	yes	no	no	no
Complex	no	no *[note 1]*	yes	yes	yes	yes
Logical types:						
— 1 byte	yes	yes	yes	yes	yes	no
— 2 byte	yes	yes	yes	yes	yes	no
— 4 byte	yes	yes	yes	yes	no	yes
IF-THEN-ELSE	yes	yes	yes	no	in Ratfor	yes
WHILE-UNTIL	no	no	no	no	in Ratfor	no
Price	£200	£350	£375	£320	£295	£595
8087 support	£50	incl	incl	incl	£35	incl
Other extras:						
— librarian	no	£95	no	no	£95	£95
— linker	no	no	no	no	[in librarian]	no
— optimiser	no	no	no	no	£65 [Ratfor]	no
Operating systems	p-System	MS-DOS	MS-DOS CP/M-86	MS-DOS CP/M-86	MS-DOS	MS-DOS *[note 2]*

Notes:

1. supposed to be present in latest version, 3.20
2. needs more than standard RAM and 8087 to work on Apricot

The Disinformation Column

Tricks with SuperWriter

by David Langford

The subject of this column is Revealed Truth about the Joys of Word Processing and Programming, varied with Amazing Cock-Ups and Random Abuse.

To introduce myself: I'm a professional writer who in 1984 picked the Apricot PC as an ideal word processing machine — because the keyboard and screen knocked spots off the opposition, because the opposition wouldn't fit on my desk, and because my oldest drinking pal is a secret Apricot dealer. He even slipped me a bootleg copy of DISK which, unlike the version supplied by ACT with my double-sided drives, would actually format double-sided disks ...

What struck me early on (apart from a challenging lack of user-friendly manuals) was the way everyone rubbished the word processor then bundled with Apricots, Sorcim's SuperWriter.

"WordStar is the industry standard", one stern friend insisted. So is the QWERTY keyboard, originally devised to slow typists to a pace that the first clumsy machines could take: I found that no matter how slowly I typed, WordStar could be slower.

"Vizawrite is state-of-the-art", urged a salesman while pointing out essential frills like windows and pull-down menus. These had apparently been introduced at the expense of less trendy effects like debugging: I wrestled with Vizawrite all day and it kept hanging up for the oddest reasons. It also refused to support spelling dictionary editing or examination, one-and-a-half-spaced printout, microjustification, proportional spacing, or documents longer than about 2,250 words (the rest of the workspace being reserved for an unwanted Document 2).

Slowly I also realised that the cluttered screen, with its pretty borders, was depriving me of four lines and several columns — which in SuperWriter contained actual text. *This* was the state of the art?

So I plumped for the despised SuperWriter ... and discovered that even ACT hadn't given it a fair deal: the manual was several releases behind the program. Sorcim wasn't interested in giving support ("We're concentrating on EasyWriter II these days."), but a kindly ACT mole eventually passed on some tatty xeroxed sheets which supplemented the manual.

Curiosity aroused, I used DEBUG to inspect the vitals of SuperWriter and learn still more.

The upshot was that a pal and I became Ansible Information, preparing and flogging a disk of utilities to help SuperWriter users. (Note for the curious: in science fiction, an 'ansible' is a faster-than-light communications device. I have been known to write SF.) So here are some snippets from the workbench.

Which SuperWriter?

Look at the title screen. Around Spring 1984, ACT stopped providing SuperWriter version D1.09 and introduced what only sounds like a retrograde step in the shape of SuperWriter 1.03. Your dealer should really offer you the chance to upgrade. (1.03 is also the version in the £95 'Super Pack' — which is good value for SuperWriter alone; you also get SuperCalc to toy with and SuperPlanner to throw away.)

Version 1.03 has new options: 'Create Backup Yes/No' on the Save Menu, a numbered tab ruler on the Global Menu, the option in Dictionary Maintenance to dump your entire spell-check dictionary as a text file, and a wider range of installable printers.

In fact it's generally superior, except on two points: Print Menu formatting seems somewhat slower, and it does not support the 'dual hyphen'.

The 'dual hyphen'

No, don't rush to the manual; this, like so many other things about SuperWriter, is not documented. Sorcim does admit that Control-P followed by a hyphen will insert a 'soft' hyphen into a word: this appears as a dim hyphen sign at which the word can be 'broken' by word-wrap, and it's only printed when broken. (Its only real use is to even out text, especially when printing with the JUST or JUSTC commands to give a straight right-hand margin. Using soft hyphens and justification in submissions to professional editors is one of many useful ways to increase your haul of rejection slips.)

In SuperWriter D1.09, a hyphen followed by a soft hyphen is printed as a single hyphen — whether or not the word is broken at the soft hyphen. This has obvious advantages, since SuperWriter doesn't break words at normal hyphens. But this useful feature mysteriously vanished in version 1.03.

Check the dictionary

There are some dodgey items in the 23,000-word lexicon supplied as SW.DIC: Americanisms, for example, inconsistently mixed with English usage. Among the weirder inclusions I particularly like 'emperoress', 'strmummy' and 'undp'; the most charming is 'londonberry', presumably a kind of Irish fruit.

Cleaning this up was a lot of work and a lot of fun. My colleague Chris Priest took a different line: he started with a blank dictionary and monitored everything added to it, in hope of discovering his working vocabulary ...

Funny characters

Users of Apricots with MicroScreens will have discovered that INS MARK ^ and a letter key inserts a weird double-character into the file — a control character, in fact, comprising the up-arrow sign followed by the chosen letter, both displayed in half intensity. (Those without MicroScreens can get the same effect by Control-Q 2 ^ and a letter.)

Most of these special characters aren't much use save in Execute files, yet another undocumented joy of SuperWriter (more of this hefty subject some other day). But one exception is ^H or backspace.

Say you want to fake an umlaut by overprinting a double-quote on an 'o'. It's often handier to insert ^H than to use an embedded OUT format command. But note that if underlined text follows on the same line, the underscore will be one space out of sync.

Blinking cursor

That placid reverse-video cursor is fine for most purposes, but it can get lost — as when editing a file with big blocks of capital letters. No problem: you can make it blink on the PC or Xi. (The facility has apparently been disabled on the F series.)

You probably start SuperWriting by loading it via a batch file which also loads a keyboard configuration. If so, you can actually use SuperWriter to insert an Escape code into such a file to produce a blinking cursor throughout your word processing. Include these lines in your batch file:

```
ECHO ^[2
SW
ECHO ^[3
```

INS MARK followed by either [or ESC will generate the Escape code ^[. Escape 2 — ^[2 — switches the blinking on, Escape 3 (that ^[3 sequence in the third line) turns the blinking off again.

Add ECHO OFF at the beginning to tidy the result, and this could be the entire text of a file called, say, W.BAT which loads blinking SuperWriter when you simply type 'W' and Return.

Want to load a text file at the same time by typing 'W filename'? Then have 'SW %1' in your batch file instead of plain 'SW'. This works — trust me!

Variables

My favourite SuperWriter bug is the way that Control-C — 'page down' — also acts as a Control-C interrupt to halt SuperWriter's internal background printing. But most of the real SuperWriter bugs lie in the area of variable text.

Avoid 'dollar' variables in particular: this way madness lies. Here, though, are a few system variables you won't find in the manual. Insert them between backslashes in your text to substitute the appropriate result when your file is printed:

%CDATE	gives file creation date in US format
%DATE	today's date in US format
%FILENAME	the name of the file being printed
%FNAME	same as %FILENAME
%MDATE	date of the last file modification (US format)
%TIME	the time in 24-hour clock format

More disinformation about SuperWriter and my other true love, Turbo Pascal 3.0, will follow in later issues. ◆

Coming soon

We don't have a rigid 'forthcoming features list'. But we do have on-going series, commissioned articles ready for the next issue, and a lot of plans. Among them:

□ **Workshops** to provide some insight into popular packages. Subjects we want to cover include dBASE, SuperWriter, SuperCalc, Lotus 1-2-3, and Cardbox; let us know if there are other packages we should write about.

□ **Technotes** are shorter pieces in the same vein, usually of a more technical nature.

□ **Major reviews** are scheduled on several subjects — among them CAD packages, Fortran tools, MS-DOS utilities. Coming up later: low-cost modems, cheap plotters, Telex and electronic mail, and more

□ **More DEALS FOR READERS disks** are already in preparation. We're looking for a source of inexpensive top-quality memory boards, too

□ **The Q&A Query-Answering Service** will appear regularly from now onwards. If you have any questions on any Apricot-related subject, try us — phone 01-609 3800 or write in. We'll try to respond personally within 48 hours, and we'll print the more interesting queries.

But it's *your* newsletter: tell us what you what to read — and don't hesitate about offering us your articles!

Apricot's market share

CONTINUED FROM PAGE 1

And Wharton's dealer panel probably isn't too representative. It's easy enough to get a complete list of dealers for IBM and Compaq; it's not too hard to get hold of Olivetti's list, too. With those names you're covering the largest manufacturers of 16-bit machines in the UK ... apart from Apricot. The company doesn't allow easy access to its dealer list, and it's damned difficult to find a complete list of Apricot dealers — we know, we've been trying to get one for the purposes of promoting APRICOT FILE to those people.

A market research outfit will have to do what we did, build a list of dealers by other means. Actually there are all kinds of ways of doing that: in particular, many dealers advertise Apricots or put the 'Apricot-approved' sticker on their ads. A judicious trawling of the computer and business press plus diligent perusal of the country's Yellow Pages will quite easily provide a list of 300 or so known Apricot dealers.

It's clear that Apricot sales have been slipping as a proportion of total sales: street-level prices on the PC and especially on compatibles have been getting lower and lower this year, removing much of the Apricot's appeal on straight cost. By contrast, Apricot Ltd appears to be holding the line much more firmly on discounting: most Apricots are still sold at or near the RRP. The bad press obviously hasn't helped the company's image, either.

Turning stones

So gone are the days (they happened a year ago) when Roger Foster could easily claim that ACT was running IBM a very close second, say 30 per cent of Britain's PC purchases to IBM's 35 per cent.

But the Apricot is still selling well — very well, in some areas. We'd reckon that the 1-2-3 promotion in September was worth a couple of hundred more shipments overall. Still, our gut feeling is that Apricot *isn't* doing as well as of late: that it's market share *is* declining: that the two years' euphoria and hype *had* to be put down somehow sooner or later: but that 15 to 20 per cent of the UK market is not unreasonable both as a target for the company and as a fact.

One well-known independent London dealer we spoke to was wary of sticking his neck out with so much flak flying around (a common reaction among Apricot dealers these days) but he was prepared to offer his view of the matter. He agreed with parts of the Bird article — the bits criticising Apricot's often, er, "optimistic" advertising claims; but he most definitely disagreed with the market share quoted — "damn lies and statistics" were the words he used.

He did however confirm that his company had experienced a recent "small" downturn in Apricot sales. But he pinned this fairly and squarely on the 'PC2' effect whereby sales of existing products plummet as soon as the end-users get wind of its rumoured replacement.

In any event, *his* down-turn had occurred *after* the period covered by the Wharton survey. "The Apricot PC range was a little long in the tooth, the F series had yet to make an impact, and and it was quite predictable what would happen to sales once word got out about an impending 286-based machine." The report had "certainly caused jitters" both among dealers and end-users; "you only have to look at what happened to Ericsson sales when it was reported that production of the Ericsson PC had been halted".

His conclusion: "the report had probably been damaging to sales". His main fear: that the computer buying public would treat Apricot's swift and expensive rebuttals the same way that they treat the company's other ads — "with a pinch of salt ..."

Reactions

Apricot too clearly found the article damaging enough to merit a reply "to put the record straight". An Apricotperson told us: "It was a pity the article didn't appear in a trade journal where it would have been dismissed and would not have attracted any attention — the SUNDAY TIMES gave it an air of authority and quality it did not deserve". (That's actually a bit harsh, since most of the computer-press reports did cast doubt on Wharton's claims and did quote some of the alternative figures.)

Anyhow, Apricot complained to the SUNDAY TIMES, and the SUNDAY TIMES retorted that the article merely commented on a report that was available to the industry — so no right of reply was warranted.

That's the official (and possibly diplomatic) line from both sides. We heard a slightly different story: that initially the SUNDAY TIMES agreed to carry an article about the company and/or an interview with Roger Foster, but 24 hours later withdrew that offer and instead said it would print a letter from Apricot. And a couple of days later that offer too was withdrawn ...

Probably all rumour, of course. Apricot's concern, as it was put to us, was "the fact that the survey was based on invalid research ... The figures are quite wrong: the dealers polled were unrepresentative". And the sudden rash of ads in the quality Sundays a fortnight later represented Apricot "now trying to put the record straight".

Those full-page ads kicked off with "Apricot shows its true colours" and listed what the company obviously feels are its strong points. Specifically:

- Apricot sales were £22 million, £50 million and £92 million respectively in the last three years
- "We've sold more computers in the first six months of this year than in the whole of 1983/84"
- Apricot holds the number two position in UK sales over the last four months — Apricot 19.2 per cent, IBM 43.4 per cent (Romtec's figures)
- international sales in the past year increased tenfold to £22 million
- shipments of Apricots during August were up by 70 per cent over the same month a year ago
- lots of blue-chip companies buy Apricot
- Apricot's three mainline machines were among the UK's top seven best-selling micros in September (Romtec again)

Yes, there's a small *mea culpa* note mentioning "problems which will seriously affect our first half-year results". But this is a straight-from-the-shoulder ad telling it like it is: Apricot clearly wanted to be seen as an open, honest company interested in the truth ...

Romtec's September best-sellers

1.	IBM PC/XT
2.	IBM PC
3.	IBM PC AT
4.	Apricot Xi
5.	Apricot PC
6.	Olivetti M24
7.	Apricot F1
8.	Apple Macintosh
9.	Apple IIe
10.	Apple IIc

We reckon that particular little campaign cost Apricot something around £100,000. A high price to counter one small article? Well, yes and no. Apparently the ads were booked in advance to launch Xen: but they were amended to include more the company's corporate stature generally and market share in particular.

So where does this leave us? Lord knows we have no particular axes to grind: we aren't apologists for Apricot, we get no favours from them (we've yet to receive a single subscription from anyone working for the company!), and we've our own critical views on Apricot Ltd.

But we do have a degree of worry about the ethics involved. We wouldn't accuse Jane Bird of irresponsible journalism: after all, she *was* just reporting Wharton's conclusions. But the words of the SUNDAY TIMES very obviously do carry weight in the City and among putative purchasers of Apricots: the toothless Press Council provides no effective redress once the damage has been done, and it seems no defence to claim that the *facts* are right (yes, Wharton *did* publish those figures) if the *implications* are not. ◆

Writers in this issue

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

Henry Deckhand is a pseudonymous cynic whose role at the newsletter is to generate jaundiced opinions about the computer industry and the way it conducts itself in public.

Lindsay Doyle escaped from the dungeons of MICROCOMPUTER PRINTOUT in the confusion following that magazine's demise, and had contributed just one piece to 16 BIT COMPUTING before that magazine also crashed. But we don't hold that against him: he now resides in a specially-furnished cell deep beneath the editorial offices of APRICOT FILE, and there he is motivated by the promise of occasional bread and water to generate a series of articles about using the Apricot PC and Xi.

David Langford is a full-time writer and computer addict, whose latest novel LEAKY ESTABLISHMENT is "by no means" about his years at AWRE Aldermaston. He is inordinately proud of winning one of the 1985 Hugo Awards, Science Fiction's 'Oscars'.

John Bunney describes himself as "an ageing architect, now pretty well past it". That combined with many years messing around with large and small computers has produced a one-man-band design researcher and consultant in microcomputer applications, programming and data management.

Edward Bromhead is Reader in Geotechnical Engineering at Kingston Polytechnic's School of Civil Engineering. He says he doesn't believe that integrated packages, icons and mice are the answer to Life, the Universe and Everything ...

E Hinchliffe is an ex-engineer enjoying early retirement and generally hindering his wife, who's a freelance processor of other people's words (mostly dissertations) and uses an Apricot.

Review

Pretty printing, pretty good price

Polyprint reviewed

by John Bunney

PolyPrint is an appealing little MS-DOS utility from a company called Arcom **Software. It turns a bog-standard dot matrix printer into something that can print formatted documents in 12 stylish proportional and letter-spaced NLQ typefaces.**

It will work on most dot matrix printers capable of dot graphics (including the Epson JX-80 colour printer). It is inexpensive. And the printed results are sufficiently good to substitute for, if not pass for, typeset material for many purposes. At less than £30 all-in, how can you go wrong?

What you get

PolyPrint is a post-processor — you write something with a word processor and run it through PolyPrint instead of printing it immediately. Depending on how you've set up PolyPrint, it will produce nicely-formatted printing in one of several different styles on an ordinary low-cost dot matrix printer.

And those print styles include several that approximate to what you'll find on a daisywheel printer — plus some fancy display typefaces.

There is a mild restriction on what printers you can use: your printer must be able to do dot-by-dot graphics (but these days most matrix printers can) and PolyPrint must be set up to send the appropriate control codes to that printer (but the supplier can almost certainly handle that for your printer).

PolyPrint can accept ready-to-print output from any word processor; it will also work directly from WordStar files without your having to save them on disk first. It's up to you to format your text as you want to have it printed and to take account of the differing character widths associated with each typeface.

Print styles

Arcom says it expects to extend the range of typefaces. But at present the 12 print styles available include faces such as those used for magazines, books and manuals as well as elaborately decorative faces like Commercial Script and Flash Bold. Among the plainer typefaces are a light and medium Helvetica, standard and italic Bodoni, standard and double-height Broadway, and American Typewriter (more light than medium).

Four of the typefaces faces may be used in non-proportional mode, and three faces are double-height. All provide for eight national character sets. The UK set includes the Sterling symbol in place of the hash mark at ASCII decimal 35.

Characters are nicely made, densely printed and have acceptably clean perimeters, though descenders are a bit mean and underlines rather tight.

Using asterisk-led codes at the start of a line, you can specify whether the text following is to be printed underlined (including spaces), double-width, or centered within the text width. Some of the faces in double-width are really very handsome indeed, especially the double-height Cooper Black.

You can freely combine the three codes with each other; and if you have the colour printer, you can also mix them with other codes that switch colours.

The proportional spacing works well, especially when the line is left to find its own length from the left. When justified to both margins, space padding is evenly distributed along the line — though if a line of text is more than one or two characters shorter than nicely fills out your specified line width, you tend to get a rather windy appearance between words.

The best results in justified mode are when you've lines of more or less equal length which are just fitting your physical text width. This takes a bit of experimentation at both the text-formatting and PolyPrint stages; and in the end, the very best appearance will need changes to the text itself.

Installation

Printer and other defaults can be permanently changed using the PolyPrint install program PPINST.COM; some can be temporarily reset just for a particular run from the main program itself.

The manual for PPINST comes on disk. It's essential reading during installation and should have been reproduced in print.

PPINST allows you to set the obvious items such as the default typeface, language, page format and printer codes. But it also asks you for the highest legitimate disk drive letter, allows you to change menu border characters (are either of these options really necessary?), and records codes for the nine keys you need to designate as PolyPrint function keys — that's handled rather neatly. On the other hand, it's confusing to be asked if you want non-proportional mode when you've just specified a typeface in which this is not available ...

You can specify the width of a left margin in millimetres (or that text is to be centered), the

PolyPrint typefaces

Bodoni
!"£$%&'()*+,-./@ABCDEFGHIJKLMNOI
XYZ[\]^_'abcdefghijklmnopqrstuvw>

Flash Bold
!"£$%&'()*+,-./@ABCDEFGHIJKLMN
XYZ[\]^_'abcdefghijklmnopqrstuv

Commercial Script
!"£$%&'()*+,-./@ABCDEFGHIJKLM
XYZ[\]^_'abcdefghijklmnopqrstu

Helvetica Light
!"£$%&'()*+,-./@ABCDEFGHIJKLMNOPQRSTU
XYZ[\]^_'abcdefghijklmnopqrstuvwxyz{|}~

Broadway
!"£$%&'()*+,-./@ABCDEFGHIJKLM
XYZ[\]^_'abcdefghijklmnopqrst

American Typewriter Medium
!"£$%&'()*+,-./@ABCDEFGHIJKLMNOPQR
XYZ[\]^_'abcdefghijklmnopqrstuvwx

Helvetica Medium
!"£$%&'()*+,-./@ABCDEFGHIJKLMNO
XYZ[\]^_'abcdefghijklmnopqrstuvw>

Bodoni Italic
!"£$%&'()+,-./@ABCDEFGHIJKLM*
XYZ[\]^_'abcdefghijklmnopqrstuv

Tiny Print
!"£$%&'()*+,-./@ABCDEFGHIJKLMNOPQRSTUVW
XYZ[\]^_'abcdefghijklmnopqrstuvwxyz{|}~

Cooper Black(Large)
!"£$%&'()*+,-./@ABCDEFGHI
XYZ[\]^_'abcdefghijklmn

Broadway (Large)
!"£$%&'()*+,-./@ABCDEFGH
XYZ[\]^_'abcdefghijklmnopqr

Compacta(Large)
!"£$%&'()*+,-./@ABCDEFGHIJKLMNOPQRSTUVW

Bodoni on an Epson

REVIEW OF POLYPR

[A] INTRODUCTION

POLYPRINT (PP) is an appealing documents in 12 stylish proportional a and other dot matrix printers (includ good value for money indeed (£29.90 a sufficiently good to substitute for, if purposes. The program works as promi easy to use. Even the time taken

It rather goes against the grain price level. But there are quite a few aggravate users once past the dabblin and even better value with these att

[B] WHAT YOU GET

PP accepts ready-to-print outpu from Wordstar document and non-docu line re-formatting is attempted. The 1 those used for magazines, books and Medium and Helvetica Light), as well Commercial Script and Flash Bold

Four faces may be used in non- double-height. Characters are nicely r clean perimeters, though descenders a

The proportional spacing works find its own length from the left. Wh is evenly-distributed, but in Tiny Pri between words, rather than between c appearance. The best results are give number of characterss nicely fitting takes a bit of experimentation. In the justified mode will require changes

All faces provide for 8 country Helvetica, a standard and italic Bodo Broadway. The American Typewriter f sequence is ASCII and the UK set inc mark at ASCII decimal 35.

width of text to which copy is be fitted (again in millimetres), the number of characters per line, and whether justification is to be on or off.

If you have an Epson-compatible or a Star printer, the printer-driver codes come preset for you (though they can be changed).

If you don't have one of these, a useful crib sheet supplied will help you set up your printer — so make sure you tell Arcom about your printer when ordering! (Arcom reckons it can set PolyPrint up for some two dozen or more different printers, but there will always be a few rarities that it hasn't yet come across.)

Changing the relevant codes from dual density graphics (27 76 0 0 for Epson FX printers) to single density (27 75 0 0) works just as you'd expect, giving an expanded, paler and dottier print. Quadruple-density graphics mode (27 90 0 0) neatly condenses output to half-width. And yes, the asterisk code *W still doubles the width of characters: there's no mention of this useful additional versatility in the PolyPrint manual.

In use

The manual is a well-written, legible and useful document; it's produced from a PolyPrint master using Bodoni (reduced in size by about 20 per cent) — though it cheats by including some single lines and blocks of non-proportional text in Helvetica Light. This is actually impossible with PolyPrint, except by applying the program to each chunk of text previously split up into separate files.

Typing "PolyPrint" at the MS-DOS prompt runs an initial program that patches MS-DOS to enhance print speed (very effectively, too). PolyPrint then shows you a main menu from which the only essential entry is the filename of the document to be printed.

At this point you can rely on the PPINST-installed defaults, or you can change some just for that run. You can call up sub-menus and disk directories, and some rather feeble on-line help is also available.

You soon get the hang of the menus, although their design is inconsistent: for example, it delays you to have to wait for a directory display and to use cursor movements to specify your document filename, whereas the font filename can be entered straightaway on a command line.

It's also odd that you're allowed to print a file only if it has an extension which PolyPrint considers indicates a printable file — you can't print CONFIG.SYS or .BAT or .HLP files, for example, unless you remove or alter the extension. You can, however, specify the name of the document to be printed from the MS-DOS command line (including those with illegal file extensions); but you still need to press Enter actually to set things moving. Re-direction from a file containing just the CR/LF gets over the problem, however, so that you can set up .BAT files for multiple printing.

Lines are printed in four passes of the print head at double-density (for Tiny Print there's only two passes) at rates varying between about ten characters per second (Tiny Print) and three characters a second (Cooper Black Large and Broadway Large).

Using Bodoni, probably the most suitable typeface for ordinary English text, a page of 52 lines (including lines between paragraphs) takes about six minutes to print — an average of about seven characters/second.

To say the least, it can take some effort to master matching a typeface to fit the line width of an incoming document *and* the printed page format *and* the left margin. In the first place, there's an absolute limit of 202mm (8ins) on the width of text that can be printed — even though you may have told Arcom that you have a wide-carriage printer.

Secondly, since the printed width of the same line of text differs with the typeface in use, the maximum number of characters that can be printed on a line also varies. These limits are not given in the manual, and to some extent they also depend on the mix of characters.

Experiments with the page format menu in PolyPrint indicate a range of between 59 (Cooper Black Large) and 109 (Tiny Print) characters per 202mm line in the proportional faces, with a left margin of 0mm. Printed tests seem to show that you can in fact get PolyPrint to print some faces at a higher density, but not to the full 202mm text width. The manuals should be much more helpful on the subject.

Drawbacks

Now for some snags. The asterisk codes may only be used at the start of a line, and affect the whole line and nothing but that line. Vertical spacing is fixed at six lines per inch (three lines per inch for the double-height faces). Since the end-of-line sequence is made using vertical graphics movements (rather than by a simple CR/LF), you can't preset the printer to any other vertical pitch before you run the program.

Typefaces can't be changed within a document or even switched between proportional and non-proportional modes: you might want to do that when printing a computer manual or a table of numbers within a document, say.

Super- and subscripts are not provided for; and PolyPrint offers no way of emphasizing or lightening, or slanting characters. Nor can you over-print or construct novel characters of your own.

In addition, the manual very decently warns you that you can't print text justified to the right only, nor vertically-aligned columns of figures — though actually you can if you use the four faces offering a fixed-spacing mode.

PolyPrint does some simple checks of the consistency of your print instructions. But you are still allowed to specify an impossible combination of typeface, characters per line and text width in both PPINST and PolyPrint; the program only baulks once you've told it to start printing

Nor does PolyPrint scan your document first to make sure that lines all fall within the specified width limits, though it could do this at the cost of a relatively small delay to proceedings. As it is, the program just gives up at a longer line than it expects, leaving you stranded; reprinting by page number ranges is provided for, however, which helps to cope with the predicament.

Any specified left margin is rather inexplicably taken out of the 202mm maximum text width, rather than being sent out as a series of dot or text space characters ahead of each line. A typical one-inch margin loses you between about seven and 13 characters. If your printer accepts a global left margin specification, you can remove PPINST's printer initialisation sequence (set all four bytes to 0) and preset the margin before running PolyPrint; then you'll find that your text is shifted correctly over throughout the document.

An aggravating omission is that no provision is made for writing even small-scale output to disk. That would be useful for example to allow later background printing, assembling files from shorter chunks using different typefaces, incorporating printed output into other documents or vice-versa, repetitive printing of top copies, or dry runs to make sure that no line of text exceeds the specified line width. (Arcom points out that for most typefaces PolyPrint is issuing around 4,000 bytes for each line of text: so you'd be lucky to get the equivalent of two sides of A4 text on a single-sided Apricot disk).

In conclusion

It goes against the grain to criticise any useable software at this price. But there are some weaknesses which sooner or later will aggravate users past the dabbling stage. PolyPrint would be much more useable, versatile and yet better value with these attended to — even if it means £10 or £20 on the price.

Arcom should certainly be encouraged to develop PolyPrint to provide more control over the capabilities it already offers, a wider range of typefaces, and more consideration for the way the program could be exploited in practice by the more enterprising.

But PolyPrint is an amusing and useful utility, well capable of sharpening up a company report, restaurant menu, letter-head, name-badge, handbill or club notice board. The program works as promised, is nicely presented and is generally easy to use; even the time taken to print is bearable. It exceeds any reasonable expectations from software of any kind at the price; and it demonstrates how much can be provided for so little from simple programming, presentation and marketing techniques. ◆

PolyPrint

Description:	MS-DOS print utility for dot matrix printers offering 12 NLQ typefaces and proportional spacing
UK supplier:	Arcom Software
Address:	45b Princess Road Branksome Poole Dorset BH12 1BG
Telephone:	0202 762236
Price:	£29.90 including VAT and P&P
Configuration:	any Apricot
Conclusion:	Excellent value — a simple but effective utility marred only by limitations likely to annoy users beyond the dabbling stage.

PolyPrint and Deals For Readers

We were so impressed with John Bunney's review of PolyPrint that we have added the product to our mail-order DEALS FOR READERS catalogue. Both the Apricot and the Sirius versions are in the catalogue at an all-in price (including VAT and P&P) of £30 — note that you have to specify your printer when ordering so that PolyPrint can be set up for it.

Deals For Readers

In the DEALS FOR READERS lists you'll find ...

- public domain software, mostly games and utilities that aren't easy to obtain elsewhere — and many of these are unique to APRICOT FILE
- deals on software, typically useful items that aren't widely available
- demo disks for packages that we regard as pretty good

How to order

Send us a note with the item's name and the reference number as quoted in the list. We will normally be able to process orders within two working days.

Note that for orders totalling less than £50 we can only accept prepayments and only in Sterling, so a cheque or money order (made out to "TP GROUP") *must* accompany your order.

We will accept purchase orders if you want goods totalling more than £50, but if it's at all possible we would prefer cash with order — we just aren't big enough to provide the effort of chasing invoices and to cover the shortfall between despatch and delivery.

The prices quoted include VAT and despatch. Your order will arrive with a VAT invoice/receipt.

Overseas orders

Orders from outside the British Isles are at the same price. Anything we send abroad is free of the VAT sales tax, and since that 15 per cent levy just about approximates to the extra cost of overseas postage we thought we might as these simplify things.

This may however mean that bulky items or long-distance despatch will go by the postage rate that most closely fits the 15 per cent allowance — and sea mail can be interminable. So if you want to ensure swift despatch, or if there's any doubt, ask us to quote ...

Contribute!

We're always looking for contributions to further £15 disks. So if you've written something — a game, a utility, a helpful batch file — why not share it with other readers?

Public Domain Disks

These disks contain programs and text files culled from a variety of sources — some converted from other formats (mostly IBM PC and Victor) and checked out or revised for the Apricot: some taken from published programs: some written by contributors. As a result the disks are a bit of a miscellany, but we'd be very surprised if you couldn't find a dozen or more useful items on each ...

Apricot Disk No. 1

Order no:	**AP001**
Price:	**£15**

Virtually the same as the 16 BIT COMPUTING A001 disk, with over 40 files aboard it. We've amended some of them and replaced a couple. Highlights now include:

AUTOEXEC an improved prompt that tells you the date, time and current directory on the current drive
SDIR and **FDIR** two handy sorted-directory utilities
VDEL and **DELQ** two selective DELete commands
SYSTAT reports disk usage and free space remaining
WAIT and **ASK** two batch file commands, both more useful than PAUSE
STRIP makes WordStar document readable by other WP packages
WC counts characters, words, lines in a text file
GRAFCHAR a kind of EFONT in Basic for defining your own characters
ESCAPES text file containing the complete ESCape code for Sirius/Victor and Apricot
DUNGEON improved version of the Adventure-style game
OTHELLO a good implementation of the well-known board game
LEADTRAK and **ACCTRACK** two sales lead management systems

Apricot Disk No. 2

Order no:	**AP002**
Price:	**£15**

As the 16 BIT COMPUTING disk no. A002 — 79 files. Highlights include:

ALTER and **CHMOD** to change file attributes
BASICKEYS single-key entry of Basic commands
CRF Basic cross-referencer
DEBUG the complete Apricot MS-DOS 2.11 version
EPSON sets alternative print modes on Epsons and compatibles
INDEX-84 complete index to 16 BIT COMPUTING 1984
KNEW backup utility that copies only new files
LOWER and **UPPER** change case of text files
MENU demo of simple batch-file menu system
MOVE moves files from one subdirectory to another
PLOT demo of plotting and screen resolution
RENDIR rename subdirectories
SORTDISK sorts directories in a variety of ways — invaluable if you have a lot of files on a disk
TREEDIR displays a DIRectory of all files on a disk — in their subdirectories

VCOPY selective COPY
WHEREIS searches disk subdirectories for named file(s)

Apricot Disk No. 3

Order no:	**AP003**
Price:	**£15**

A bumper bundle of goodies you can't live without, culled from both sides of the Atlantic. Highlights:
SWEEP excellent all-purpose file housekeeper — beats any such utility that we've seen
SCAVEN finds and locks out bad sector on microfloppies
MEMDUMP browse through memory
TURBOKEY key redefinition for aficionados of Turbo Pascal
SUPERWRITER XQT FILES a useful clutch of macros
HEXFILE create hex-coded ESCape sequences and the like — comes with some fun MicroScreen-driving demos

Games

Order no:	**AP004**
Price:	**£15**

For the first time on Apricot, a disk of public domain games — and nothing but games. Some in Basic, some in EXEcutable form. Includes the very tough flight simulator **747SIM** (fly your own Jumbo) plus many old and new favourites. A very full disk

GSX User's Disk

Order no:	**AS005**
Price:	**£15**

The complete GSX tutorial from Lindsay Doyle — includes a **library of ready-to-use Basic subroutines** that drive GSX, a set of five printable **Workshops** in text form (this series is running in APRICOT FILE now), **demo programs**, and Lindsay's improved HEXPAWN game **IMPHEX** (which among other things demonstrates something not unlike artificial intelligence).

Demo Disks

Try before you buy? Here's your opportunity to check out our selection of some of the best Apricot packages — and to do so without a salesman hassling you to write a cheque for them. All disks are priced at £15 including despatch and VAT.

BusiPost Demo

Order no:	**AD002**
Price:	**£15**

Excellent British mailing list package - fast and flexible. Not available for the Apricot F1.

ChartMaker Demo

Order no:	**AD003**
Price:	**£15**

Scientific/technically oriented graphics from keyboard or from files created by database and spreadsheet programs. Lots of graph-drawing functions built in.

Letterer-Set Demo

Order no:	**AD005**
Price:	**£15**

Interesting little utility that lets you print on any dot-matrix printer any typeface that can be displayed on your Apricot screen.

Lotus 1-2-3 Tutorial

Order no:	**AD006**
Price:	**£15**

If you're thinking of 1-2-3 for your Apricot, here's a chance to see just what it can do. Requires 256KB.

MENUGEN Demo

Order no:	**AD007**
Price:	**£15**

Demo for a fine menu-creation utility — menu selections can run programs, call other menus, execute MS-DOS commands, etc. Like ASPECT, a Microft Technology product.

MicroSight 1 Demo

Order no:	**AD008**
Price:	**£15**

Everything you need for digitizing an image and several facilities for manipulating it.

Praxis MicroChart Demo

Order no:	**AD010**
Price:	**£15**

All-British business graphics package, ideal for presentation of spreadsheet data etc. Not available for the Apricot F1.

Cardbox-Plus

Order no:	**AD011**
Price:	**£15**

Particularly good demo for one of the best 'flat-file' records-keeping packages. We use it extensively oursleves; find out why ... Not available for the F1.

Special software

This modest collection represents our first attempt at purveying conventional packages: these are not public domain items, and they are not discount deals on the kind of software your dealer will sell you. Instead we have gathered a few titles that (a) we feel will be useful to the average Apricot user and (b) you won't find easily — or at all! — anywhere else.

FAST Apricot Screen Accelerator

Order no:	**AS001**
Price:	**£25**

Have you slyly suspected that your Apricot screen was a little, er, slow?

Well, you're right. The Apricot screen is slo-o-w-w ... especially when it comes to scrolling about a screenful of text. It really can be quite excruciating scrolling off of a SuperCalc spreadsheet.

The rumour is that the designers of the Apricot chose to put various screen-handling impedimentia deep within the BIOS in order that it should not compete on performance too strongly with its then stablemate, the Sirius. And these screen delays have remained there ever since.

Until now. FAST is an accelerator for the Apricot screen: it bypasses certain screen interupts. Simply running it in an AUTOEXEC.BAT will cause your Apricot to handle screen text *twice as fast* — it will redraw your screen in half *the time it formerly took.*

FAST doesn't affect CPU clock speeds and so it does not accelerate your graphics or supercharge your program execution times. But its effect on screen performance is pretty impressive.

FAST is very compact, less than 650 bytes; and it's modest price reflects that. For the money you won't find any better way of speeding up an Apricot with a single-line AUTOEXEC command ... Available only through TP GROUP.

BPS Disk no. 2 — RAMdisk with WordScan

Order no:	**AS002**
Price:	**£40**

The **RAMdisk** is better than the Apricot's freebie; the **Wordscan** word/line/character counter is *very* fast and always useful for anyone who writes text on their computer. Normal RRP is at least £70.

POLYPRINT

Order no:	**AS003**
Price:	**£30**

Pretty printing from your Apricot on any matrix printer — POLYPRINT lets you produce near-letter-quality printing (almost typesetting quality, in fact) on ordinary dot-matrix printers that are capable of printing graphics (which these days means just about any low-cost matrix printer). Menu-driven, uses only nine keys. Comes with 12 selectable typefaces, some of them very fancy. Accepts ready-to-print output from any word processor; will also work directly from WordStar files without your having to save them on disk first. Specify which printer you have when ordering — POLYPRINT comes customised for particular printers.

The Apricot Catalogue

Order no:	**AS004**
Price:	**£50**

An ASCII-format list of all the software and hardware we know of for the Apricots 1 with prices, addresses, description (with our comments when we have experience of the package), contact names and numbers (including Telex numbers) where we know them. Based on the lists originally printed in 16 BIT COMPUTING, but regularly updated by us; currently well over a megabyte of information on six disks.

Professional Author's SuperWriter Disk

Order no:	**AS005**
Price:	**£46**

The first (and only?) independently produced package of upgrades to and tools for SuperWriter — produced by a couple of professional writers (one of them our columnist David Langford) for people who want to make the most of the word processor. Highlights:

- **two-column** formatting and print
- expansion, improvements, corrections for the **dictionary**
- full documentation of **SW.DEF** plus a collection of useful **formats**
- improved (and more legible) **display** font on the screen
- **keyboard configuration** presets to speed up keying
- many **XQT files** to speed up and simplify your routine work
- several batch files for **disk housekeeping**

A good piece of work — we're reviewing it in our next issue, but PASD is sufficiently impressive to merit an early appearance in DEALS FOR READERS.

Books

A book by the editor ...

The Good Computing Book

Order no:	**AB001**
Price:	**£3.00**

A tubby and packed paperback for beginners. Mostly a reference dictionary of computing terms, but also lots of introductory info. Claims to offer "everything you need to know and nothing you don't"; well, that may be blurb-writer's licence but Dennis is still pretty proud of the book. Outside the UK, please add £1.50 for postage to Europe, £5 for postage elsewhere (sorry about that — it's not us that sets the UK postal rates). ◆

All prices include VAT and despatch. Send your order (with payment, if it totals less than £50) to:

APRICOT FILE
TP Group
PO Box 509
London N1 1YL
England

Queries: 01-609 3800

Cheques should be made payable to "TP GROUP".

We aim to despatch within 48 hours, but please allow 21 days for delivery before calling to complain ...

The Fundamentals of GSX

Part Two: how to talk to GSX

by Lindsay Doyle

When you have called GRAPHICS, you aren't yet actually in 'Graphics' mode — you still have in the normal display mode, which for no particularly good reason other than expediency I shall choose to call 'Alpha' mode. How do the two differ?

Alpha mode

In Alpha mode the screen is effectively divided into 2,000 cells — 80 wide by 25 high — and each cell may contain a character, one of 256 from whatever font has been selected.

Each character uses part of a matrix of dots 10 wide by 16 high. If you were to print a series of characters each comprising all the 160 dots possible, the screen would be completely filled with white (well, green).

Now, if you look into this situation a little further you'll find that the memory allocated for these characters is actually composed of 16-bit words. Four of the 16 bits specify whether the character is to be displayed in normal or reverse video, in high or low intensity, with or without a strike-through bar, or with or without an underscore. A fifth bit is there to be used by the likes of word processors (for marking the beginning of a line and things like that).

This leaves 11 bits which constitute a pointer to the place in memory where the font patterns are stored. When you call up the FONTEDIT program you will see an enlarged picture of whatever character you select. The dim 6-by-16 part has two dots in it, the reason for which may have puzzled you as it did me; in fact they record just where the strike-through or underline bars are to appear, and by playing with them you can move the bars or generate multiple bars on one character.

Graphics mode

In Graphics mode the screen consists of 320,000 dots, 800 across by 400 up, and each dot can be individually addressed (which means it can be switched on or off).

The screen memory is now organized in cells each of which is 16 dots square; there are 1,250 of these cells in 50 columns by 25 rows. You won't need this information to understand what will follow; but it does help explain why when you call up normal Alpha characters while in Graphics mode they will appear with two dots between each character (the strike-through and underline dots now appearing because of the 16 by 16 cell) — and why they will be all over the screen instead of starting neatly from the lefthand edge (because there's a 50-column screen rather than the 80-column screen they expect).

320,000 bits is 40,000 bytes; and when you enter Graphics mode provision is made to store the screen contents in memory locations from 2800H to C440H. I mention this, not because you need to memorise it, but because it happens to cover the area of memory in which alternate fonts are stored when in Alpha mode. So any such fonts already in there will be overwritten.

Selecting one dot

There are two ways to specify which of the 320,000 dots you want to turn on or off. The first method is for manual input: you can have available a cursor that may be moved around the screen and placed wherever you require before you say 'ON' or 'OFF'. A cursor like this can be controlled by the cursor keys (four directions), the number pad keys (eight directions), a mouse, a tracker-ball, or a joystick.

The second way is to tell the machine the X and Y co-ordinates of the desired spot. This is what we'll concentrate on, since it works from within a program and it can rapidly generate a bunch of figures — a line from two co-ordinates, a rectangle from four, a circle from two (centre position and any point on the circumference), or text from one (the lower lefthand corner of the first letter).

GSX uses a co-ordinate system of 32767 by 32767 points. Digital Research calls this NDC (Normalized Device Co-ordinates); and I may forget myself enough to use that phrase too from time to time, so be warned.

So to indicate a spot on the screen, you must specify it on a 32767 by 32767 field. GSX will do the arithmetic to find out where it should appear on an 800 by 400 field for the screen — or if you're driving a plotter, GSX will handle the conversion for the plotter's resolution, and so forth.

The advantages of this approach are numerous, leading as it does to the ability to write device-independent programs. But it does have one small drawback: 800 doesn't go evenly into 32,768 (which of course is the number of possible points or positions from 0 to 32767). The result: we'll sometimes have to live with rounding-off effects which will displace a dot or a line by a fraction of a dot width from where it ideally should be.

That can be avoided if you feel it necessary by working in a restricted field of 32000 by 32000 with a bit of a border all round. In this case each dot will be exactly 40 counts wide by 80 high. My advice? Try the full field first and only go to the partial field if you're a perfectionist and can see the difference.

How to talk to GSX

To pass a command to GSX (or to receive the information that might fed back from GSX, in some cases) it's necessary first to create five arrays. Then, for each message, certain information is inserted in certain points of certain arrays for the specified command to use when GSX is called (and from which information may be returned by GSX).

But for each type of command or report message within GSX there is a somewhat different format — and there are 33 different messages (though not all of them are implemented for the Apricot at present). So there could be a *lot* of arrays ...

The best way to handle this messy situation and avoid the need to write a custom program for each application is to have a set of standard subroutines. These can then form the backbone of any GSX application.

We'll supply such a set of callable routines in the next instalment of the series; in the meantime, the Editor tells me that I will lose your attention if I don't provide some pretty pictures soon to prove that GSX works.

So here's a little program which should draw a lovely set of 20 vertical bars, each with a different fill pattern. After you have it transcribed to disk, run GSXLOAD.BAT (which of course you've already written according to the instructions in part one). This will put you in MSBASIC with all the bits for graphics loaded and in place; you can then run "BARDEMO" and sit back. ◆

Next instalment: instant freeze-dried ready-to-use GSX routines, plus a look at each component of the BARDEMO.BAS program. And, in the interim, if nothing else you should be able to experiment with bars. Hic.

Copyright 1985 Lindsay Doyle

BARDEMO.BAS: a GSX demonstration

```
10  REM: BARDEMO.BAS BY LINDSAY DOYLE
100 REM: DIMENSION GSX ARRAYS
110 DIM A1%(9), A2%(79), A3%(200), A4%(45), A5%(12)
120 DW = 32767: REM DISPLAY WIDTH IN 'NORMALIZED DEVICE
    COORDINATES'
130 DH = DW:    REM DISPLAY HEIGHT
140 LS = DH/25: REM LINE SPACE
150 CS = DW/80: REM CHARACTER SPACE
160 PRINT CHR$(27); "E": REM CLEAR ALPHA SCREEN
170 :
200 REM: OPEN WORKSTATION & CLEAR GRAPHICS SCREEN
210 A1%(1) = 1: A1%(2) = 0: A1%(4) = 10
220 FOR Z = 1 TO 10: A2%(Z) = 1: NEXT Z: A2%(6) = 2: GOSUB
    54000
230 :
300 REM: DISPLAY HOLLOW, SOLID, HALFTONE, AND HATCHED BARS
310 T$ = "BAR FILLS": X1 = 2 * CS: Y1 = DH - 2 * LS: GOSUB
    50800
320 BS = DW/22: REM BAR SPACING
330 BW = 3 * BS/4: REM BAR WIDTH
340 Y1 = 0: REM BOTTOM OF ALL BARS
350 Y2 = DH - 3 * LS: REM TOP OF ALL BARS
360 FS = 0: GOSUB 52300: REM SELECT HOLLOW FILL STYLE
370 X1 = BW: X2 = X1 + BW: GOSUB 51110: REM DRAW BAR
380 FS = 2: GOSUB 52300: REM SELECT HALFTONE FILL STYLES
390 FOR I = 1 TO 10
400 FT = I: GOSUB 52400: REM SELECT FILL TYPE
410 X1 = X1 + BS: X2 = X1 + BW: GOSUB 51110: REM DRAW BAR
420 NEXT I
430 FS = 3: GOSUB 52300: REM SELECT HATCH FILL STYLES
440 FOR I = 1 TO 10
450 FT = I: GOSUB 52400: REM SELECT FILL TYPE
460 X1 = X1 + BS: X2 = X1 + BW: GOSUB 51110: REM DRAW BAR
470 NEXT I
480 GOSUB 54100: REM WAIT FOR KEY
490 :
500 REM: CLOSE WORKSTATION
510 A1%(1) = 2: A1%(2) = 0: GOSUB 54000: END
50000 REM: A SELECTION OF GSX SUBROUTINES
50010 :
50800 REM: WRITE TEXT
50810 A1%(1) = 8: A1%(2) = 1: A1%(4) = LEN(T$): A3%(1) =
      X1: A3%(2) = Y1
50820 FOR Z = 1 TO LEN(T$): A2%(Z) = ASC(MID$(T$,Z,1)):
      NEXT Z: GOSUB 54000
50830 RETURN
50840 :
51110 REM: DRAW BAR
51111 A1%(1) = 11: A1%(2) = 2: A1%(6) = 1: A3%(1) = X1:
      A3%(2) = Y1
51112 A3%(3) = X2: A3%(4) = Y2: GOSUB 54000: RETURN
51113 :
52300 REM: SELECT FILL STYLE
52310 A1%(1) = 23: A1%(2) = 0: A2%(1) = FS: GOSUB 54000:
      RETURN
52320 :
52400 REM: SELECT FILL TYPE
52410 A1%(1) = 24: A1%(2) = 0: A2%(1) = FT: GOSUB 54000:
      RETURN
52420 .
54000 REM: CALL GSX ASSEMBLY SUBROUTINE
54010 DEF SEG = &H61: GSX% = 0: REM SEGMENT & OFFSET
54020 CALL GSX%(A1%(1), A2%(1), A3%(1), A4%(1), A5%(1))
54030 DEF SEG: RETURN
54040 :
54100 REM: WAIT FOR KEY
54110 T$ = " PRESS SPACE KEY TO CONTINUE "
54120 Y1 = LS: X1 = DW/2 - LEN(T$)/2 * CS: GOSUB 50800:
      REM CENTRE LABEL
54130 IF INKEY$ <> "" THEN RETURN
54140 GOTO 54130
```

Readers' Writes

This is a page for your comments — we want to maintain a strong link between ourselves and you, and we also want to provide a way for readers to get in touch with each other.

If you have something to say, just write to us. Letters are usually edited for length, and we won't be printing addresses in full unless the writer expects some direct response from readers. Otherwise we're expecting to run in each issue a selection of the more interesting mail that arrives in PO Box 509, London N1 1YL.

FILE duplication

I have an Apricot *and* a Sirius, but I am reluctant to subscribe to *both* of your newsletters if they will be substantially the same ...

Derek Ramsdale, Stratford upon Avon

Yes, we can see you might have a problem about subscribing to one or both newsletters: but let us assure you that our intention is to run *no* duplicated articles if we can help it, with the possible exception of short news items with dual relevance.

Some of the *subjects* may be the same, and some of the preliminary words used will therefore be similar: for instance, issue 1 of VICTOR/SIRIUS FILE carries a piece about using hard disks with emphasis on backup procedures and incorporating a review of Keele Codes' CLIP utility. Because the subject is relevant to owners of Apricot hard disk systems, a similar piece will appear in APRICOT FILE. But the 'how to do it' sections will be rewritten to be specific to the Apricot.

This approach will however be the exception. In the first issues of the two newsletters, *none* of the features and reviews are duplicated. It's our intention that the two should be seen as separate, not least because the feedback from subscribers suggests that they will rather different kinds of reader — for the Apricot there seems to be a much greater requirement for practicalities, like more on applications and Workshops on particular packages; Sirius users on balance seem to require a slightly more technical approach.

That suggests the relative longevity of the Sirius community, with Apricot owners being at an earlier stage in their use of the machine; and we will try to ensure that this (mild) divergence is reflected in the content of the newsletters over the twelve issues of the coming year. ◆



Apricot Computers p.l.c.
Apricot House, 17 Westbourne Road,
Edgbaston, Birmingham B15 3TR
Telephone: 021-454 9091
Fax: 021-455 8427

10 October 1985

Mr.Dennis Jarrett
Editor in Chief
TP Group
51 Thornhill Road
London N1 1JT

Dear Dennis,

re: THE APRICOT FILE

I have received your mailshot and wish to raise an important issue with you.

In the second paragraph you attribute the failure of the 16-bit on "declining advertising revenues and an ACT-sponsored competitor."

I must point out that Apricot User is not ACT sponsored. Your inference that Apricot User is not independent is totally erroneous and I feel well within my rights in asking for a retraction of this statement.

Yours sincerely

John Leftwich
Group Marketing Director

Copy: Tony Westbrook
Editor
Apricdot User

Registered in England No. 844354
Reg. Office: 4 Vicarage Road, Edgbaston,
Birmingham B15 3ES.

Software round-up

These pages represent a round-up of activity in the software market — new players, new products, and news (some of it bad) about the existing participants.

Going, going ...

It's been a tough year all round, it seems, with much contraction in the UK micro market. Fewer dealers than usual seem to have gone down the pan; but the mortality rate among vendors of accounts packages appears to have drifted up. And now the good news:

■ **Scorpion Computing** went into liquidation in April, leaving a few users of the **Easy** series accounting packages high and dry. But we're told that a new company called **Subject Software** is able to provide on-going support for around £50 a year — Subject, which was formed by one of Scorpion's creditors, is on 025 126 3706.

■ **Compact Software** also crashed in April with a claimed 12,000 users of its accounting software. The operation has been acquired more or less complete by a consortium featuring one of its dealers, however, and things should be able to continue as before — the name, phone numbers and most of the key people have been retained.

■ The departure of **Peachtree** from the microcomputer packages market earlier this year left a few poor souls with no support for the company's products. In stepped ex-MD Dick Moore, setting up **Plusmark Business Systems** to take on Peachtree's larger Cobol-based packages (and incidentally to develop them quite substantially); so PBMS users can call Plusmark for support. The smaller Basic-based PBAS systems haven't found a home, but we're told that **Evan Jackson** on 0522 42399 is offering both support and upgrades.

... and coming

None of this attrition has stopped the usual clutch of announcements and enhancements, though. Among the products that caught our eye:

■ **PowerLink** is new from **Omicron Management Software**, a software interface permitting its good (but pricey — yer pays for quality) **PowerLedger** suite of financial packages to be linked with spreadsheets. Symphony, 1-2-3 and VisiCalc are mentioned specifically (does anyone still use VisiCalc?) and it enables PowerLedger data to be used for the consolidation of multiple ledgers, currency conversion, monthly accounts reporting, budget preparation and forecasting. PowerLink is £348, Omicron is on 01-938 2244.

■ A US company called **WordTech Systems** is making a business out of filling the gaps in **dBASE** by producing software tools that make the best-selling database manager a lot easier and more useful. The family includes a collection of index speed-up routines, a symbolic debugger, and (primarily) the **dB Compiler** for dBASE II which we regard very highly. Now there's **dbIII Compiler** too; requires less memory than dBASE itself, compiles dBASE III programs using II or III data files, results in virtually uncopyable code, involves no licence fees payable by system developers, runs fast ... and if you have dBASE III on an IBM PC or a compatible, you can develop your original programs there and port them over to an Apricot for compilation. That way you can have dBASE III programs running on them without actually needing dBASE III. Sounds good, costs £695 in Britain from **Micro Minder Consultants** on 01-870 7431.

■ **Kewill Systems** reckons it has "the first ever micro-based software that allows easy modelling of complex 3-D shapes and production of complex moulds". It's called **CHEOPS**, looks good for designing things like bottles and watch cases, and at a total price of some £35,000 for hardware (Apricot is one option) and software (you'll need NC programming and 2D draughting input too) must be a cheap alternative for the smaller manufacturer. CHEOPS can be tied to the modestly-priced **MLD** draughting package from **Mountford & Laxon** if you want to use parametric varying of sections. The pedigree is excellent — Kewill is best-known for the **MICROSS** production control system. Details: 0932 248328.

■ Incidentally, it seems the Stock Market isn't so disenchanted with high-tech shares after all — 13-year-old **Kewill** made a successful USM debut a couple of weeks ago, shifting 600,000 shares at just under 80p each.

■ **FT Moneywise**, the classy British business-oriented spreadsheet, has acquired an add-on that enables it to read data from other spreadsheets (at last). **Moneywise Bridge** costs £49, Moneywise itself is £395; both from **Strategem** on 01-289 5486.

■ **Sapphire Systems** has added **Report Manager** to its business

software portfolio (the **Datamaster** file manager is probably the best-known member). The new package isn't an in-house development: authors **Datamension** claim to have sold 60,000 copies in the States. Basically it's a classy spreadsheet in *three* dimensions — up to 255 rows, 255 columns, and 255 'pages' — with good graphics, the ability to read and write files in other formats (like 1-2-3 and SuperCalc), and a good macro programming language called EXEC. Report Manager is £495, Sapphire is on 01-554 0582.

■ One of the earliest and most successful of Unix business accounting suites is now available in MS-DOS versions. The vendor is **Tetra Business Systems**, the packages cover the usual gamut (ledger accounting with sales and purchase order processing, inventory, job costing and payroll), they can handle "any number" of companies and multiple currencies, and Tetra reckons that having established market leadership in the multi-user Unix world it now intends to do the same for single-user systems. Tetra is on 0494 452001.

■ Networking has been added to **QED+**, to our mind one of the best 'Executive Desktop' program currently available. It's certainly full of features already. Under QED+ you can run any one of up to 10 other programs; it also has an appointments diary (software people are being a mite pretentious in calling this 'time management'), address book (including printing of mail labels, and autodialling too if a suitable modem is connected), project planner, and a full-function calculator with five memories.

The new version makes this available to more users simultaneously — developer **Quantec Systems** claims it will run under "most" network operating systems. We haven't seen it running, and we'd suspect there might be some noticeable slowing-down if too many people are using it. But at least all the right ideas are in there. For instance, files accessed under QED+ can be designated public or private, to prevent unauthorised access, and may also be designated 'read-only'. You get file locking, too — when any file page is already being accessed by another user, you can view the same page but you cannot make any changes until the first user has finished.

Any user can access any files anywhere on the network. So a secretary, for instance, could maintain the engagement schedules for an entire department, or a number of individuals could pool their work on a joint project schedule.

A number of other enhancements have been added to make DOS functions and the QED+ calculator available even when third-party software is running concurrently under QED+. There are improvements on the text processing side as well, with up to 60 lines of A4 text available for a document. And the calculator now gets average and standard deviation functions.

QED+ costs £295 from distributors (**Computerland, Software Limited, Thorn-EMI, MBS Rentals**); Quantec itself is on 01-228 7507.

■ **Cardbox-Plus** from **Business Simulations** is one of our favourite file manager packages — we're using it extensively, and in particular we hold the CATALOGUE on Cardbox. It's simple to use, yet it can also be quite clever; the only major disadvantage we see is its inability to do sums. And the latest releases look even better: for a start there's a £500 multi-user version for **Concurrent DOS** (and DR-NET if anyone's got it). The **single-user** Cardbox has been spruced up a bit with small changes including an 'oops' command for undeleting that last record you inadvertently wiped. And there's a **new** 'Extended version' that allows two files to be open and accessible concurrently. Price for that is £350, £50 more than the original; Business Simulations is on 0892 863105.

■ Not a new product, but a good deal on a good-looking current range — **Prospero Software** has extending its discount-for-bulk **site-licensing** deals to anyone who wants a lots of copies of its compilers. Formerly limited to higher education, the agreement lets you copy the package for 33 per cent of its price (20 per cent for 17 or more copies). Since the **Pro** compilers aren't copy-protected anyway, this deal just buys extra copies of Prospero's (rather good) documentation. Why don't more suppliers adopt this approach? Details: 01-741 8531.

■ New from **Financial and Corporate Modelling Consultants** (formerly known just as Corporate Modelling Consultants) is a **corporation tax** system called **FINTAX**. This is built around FCMC's mainline product, the classy **FINAR** modelling system, and was produced in collaboration with tax consultancy **Taxsolver**; it promises to automate "a great deal of the routine work" in calculation of corporate taxation", includes the Apricot among its implementation, costs £2,500. FCMC is on 01-262 1021.

■ **Owl Computer Training** (practically the only training company about which we've had only good reports) this month starts a new three-day **Lotus 1-2-3** course on the Apricot — two days of introduction, an optional third day 'intermediate' course. Call 044 282 7302 for details. ◆

Q&A: Question and Answer Service

Your queries answered

Q&A is your chance to make APRICOT FILE work specifically for you. On tap we have a panel of knowledgeable people all waiting for your queries on any matter affecting Apricots ...

Basic bug

When I put the following line into my program, I get a SYNTAX ERROR. Why?

```
MID$(X$,X%,1)=FNUP$(MID$(X$,X%,1))
```

There appears to be two problems here — the integer variable on the lefthand side and the use of an expression on the right. The correct use of MID$ as a statement is ...

```
MID$(a$,b[,c]) = y$
```

... where:

- a$ is a string variable or array element which will have its characters replaced
- b is an integer expression in the range 1 to 255
- c is an integer expression in the range 0 to 255
- y$ is a string expression

So for a start the righthand side of the statement *must* be a valid *string* expression — and not a function.

On the lefthand side, the very nature of MID$ requires that b and c are integers; any attempt to *declare* them as such by the use of '%' will fail. (In fact, if you were to assign a *floating point* value to b or c you'd find that the Basic interpreter will convert them automatically to integers.)

If you *must* declare these variables as integers, use a DEFINT statement at the beginning of the program. For instance, something like this should overcome the problem:

```
10 DEFINT X
20 DEF FNUP$(X$,X) = MID$(X$,X,1)
30 Y$ = FNUP$(X$,X)
40 MID$(X$,X,1) = Y$
```

Hard disk COMMAND

I have a Plus 5 external hard disk (10MB) which I run with my Apricot. Needless to say, I can't start the system from the hard disk: but I do use the hard disk almost to the exclusion of the floppies, except of course for backing up. But some programs continually look for COMMAND on floppy drive A — and often as not I've taken the damn disk out! Is there a solution?

There *is* a solution, and we have Dr Guy Gordon of White Crane Systems to thank for it.

Create a subdirectory on the hard disk called (say) C:\BIN. Then amend your CONFIG.SYS file to include this line:

```
SHELL = C:\BIN\COMMAND.COM/P
```

This will cause COMMAND.COM to be loaded from your hard disk and reloaded from there whenever necessary.

Star Master Mode

I have a Star matrix printer and in general I am very pleased with it — especially the degree of software control. But one of those controls, 'Master Select' has me baffled; nor can my dealer cast any light on it. This facility would be very useful; but I don't seem to be able to make it work.

We too like the Star among Epson-compatibles, and we agree about its versatility.

Individual print modes (Pica, Elite, condensed, expanded, condensed and expanded) can be selected by individual commands; the Master Select command is an Escape code followed by an ASCII value which allows you to select just about any combination of print modes — mixing them with normal, emphasised, double-strike and/or emphasised double-strike printing. Master Select is one of the clever things about the Star.

On the other hand, one of the less-than-clever things about Star manuals is the way Master Select is documented in the manuals for all current models ... while it's provided *only* with Star's **SR** series. Master Select isn't implemented on the cheaper Star printers, the **SD** and **SG** ranges.

If you do have an SR10 or SR15, the function works as documented. Otherwise the problem could be simply that (a) you didn't pay enough for your printer and (b) Star's technical writers didn't pay sufficiently close attention to their work.

Changing drives

In the July issue of 16 BIT COMPUTING we carried a query about changing drives from within an MSBASIC program. Quentin King has come up with an answer ...

Assuming that the enquirer wants to log in a new drive and isn't simply trying to get Basic to *read* from a specific drive, this should be quite easy to do: just set up a 'drive' variable called say DRIVE$ and preset it to "A:" or "B:". Then a drive reference can simply be replaced by DRIVE$, as in ...

```
FILES DRIVE$
```

... or ...

```
OPEN "R",#1,DRIVE$+"STOCK",128
```

DOS 2 dBASE

I use dBASE II quite a bit but was quite annoyed to find that it was one of the packages that doesn't run unaltered under MS-DOS 2.11. Can it be patched?

The version that *does* work is 2.4a, and if you have that one it's quite straightforward to DEBUG it. Make a copy of DBASE.COM to and work on that one; type 'DEBUG DBASE.COM' and when the arrow prompt appears type in 'e 53F8'. DEBUG will display the memory address on the next line; the exact address may vary, but it will look like this ...

```
nnnn:53F8 0A.
```

You then enter '90' after the OA and hit the space bar. DEBUG comes back with 'C5' and you type another '90'. That's it for this bit, so hit Return. On the screen you should have ...

```
nnnn:53F8 0A.90 C5.90
```

Now enter 'e 7C16' and DEBUG will respond with ...

```
nnnn:7C16 20.
```

Type '41' to replace that '20' and press Return:

```
nnnn:7C16 20.41
```

All done. You can now type 'w' to write the alterations to disk and 'q' to get out of DEBUG. The revised dBASE should work ok under 2.11.

The Q&A HelpLine

We're still waiting for British Telecom to put in a phone line that can be used with an answering machine for your queries. In the meantime, call our main number (01-609 3800) or use Telecom Gold (mailbox 81:JET767); or write to us at ...

PO Box 509
London
N1 1YL

Classified

Free Small Ads

It's clear from the number of responses to the 'what else would you like to see in the Newsletter' box that many subscribers want a Small Ads section. So we'll be running one from the next issue on.

It depends on what we're asked to run, but the classifications in the Small Ads could include these:

- **Sales** especially for **secondhand equipment** and pre-owned **software**. (We'd prefer not to carry ads for commercially-available products, unless it's a kitchen-table sideline of yours)
- **Wants** — which could be anything, from individual items of software or hardware
- **Contacts** including (for instance) notice of user club meetings, clubs forming, or *cris de coeur* asking for help with specific problems
- **Misc** — well, we've got to have somewhere for ads that don't fit the other categories

How to send in your ad

Simple — just let us have a sheet of paper (or preferably a disk) with the words. To summarise the requirements:

- ☐ indicate which classification you'd like it to appear
- ☐ use a typewriter or printer if possible — disk is even better. If you handwrite it, please try to ensure legibility
- ☐ the first couple of words will be in capital letters for emphasis. Indicate where you want the emphasis if it's not clear
- ☐ don't forget to include an address and/or phone number (we won't be operating a box number service)
- ☐ please try to limit yourself to 50 words or less

Send your ad to:

APRICOT FILE
PO BOX 509
LONDON N1 1YL

To get us started, here's a selection from our own immediate needs that also serves to illustrate the style we'll be using.

Wants

■■■ LOTUS 1-2-3 ENTHUSIAST required by APRICOT FILE to check through a heap of Public Domain utilities, spreadsheets, tutors etc (all originally for the IBM PC) with a view to constructing a DEALS FOR READERS disk for Apricot users Contact ROGER GANN on 01-609 3800.

■■■ REVIEWERS WANTED. We're looking for people prepared to write intelligent, intelligible product reviews — particularly for relatively specialised areas like comms and CAD. Contact DENNIS JARRETT on 01-609 3800.

Sales

■■■ APRICOT INTERNAL MODEM for sale — no reasonable offer refused. Comes with IMASYNC. Contact 01-609 3800.

Big dealer

Specialist Computer Centres has become the UK's second largest computer retailer chain (**MBS** is the biggest) after buying the ten **Byte Shops** from **Kode** for something over £800,000. Kode had acquired them along with **Comart** early in 1984 but now says it wants to concentrate more on multi-user systems than High Street retailing. Apricots are featured ...

The Liberal edge?

For some reason we're on the mailing list for a chatty little number called MICRO NEWS, a 20-page tabloid newspaper published by 'the Conservative Computer Forum'. Lead article is about the **Tory defeat** at Brecon and Radnor earlier this year: seems the **Alliance** didn't win there because of any old-hat reasons like voter appeal or local unemployment or government policies or the candidates' taste in electioneering wear or even because the late Tom Hooson was the sitting MP. No, "a recent MICRO NEWS investigation" reveals that the Alliance scored because the Liberals have computerised the voting register to improve their canvassing. Ah, we knew there had to be *some* explanation.

Tact corner

Ashton-Tate's new Maidenhead HQ was opened a couple of weeks ago by **Cecil Parkinson**. The pianist subsequently swung into 'All you need is love' ...

AprilTory

Incidentally, MICRO NEWS also reports that "more than fifty" Apricots are now being used by **Conservative MPs and constituency associations**, mostly for membership recording with the **Fileplan** database from **Resource Management**. ACT/Apricot Ltd has long been associated with the Party, of course; it was one of half a dozen computer companies participating in a mini-exhibition at the Conservatives' Blackpool conference, and MP and Minister **John Butcher** is still on the Board.

Apricots go east?

The world's computer manufacturers are currently courting the **USSR** with a view to picking up some big business from Russia's **projected computers-in-education programme**. First fruits have gone to **Japan** for home-'n'-hobby computers, but things are still up for grabs in the 16-bit arena. **Apricots** might still be grabbed — the group's local rep was **Quest Automation**, which has three of the eight blokes in Moscow representing UK companies. "We expected them all to be expelled" said Quest, but only one of them was included in September's tit-for-tat expulsions.

Unclog your printer

If you're getting a **fuzzy print impression**, the cause may be ink and paper detritus clogging up your printwheel or the print head in a matrix printer. **PerfectData**, which is big in disk head cleaners, has now come up with a print head cleaner for daisies and matrix printers — it's a fibre sheet to which you apply the Special Cleaning Fluid. Remove the ribbon and run the self-test; the garbage is removed. Two-sheet kits cost £8.90 from **Action Computer Supplies** (01-903 3921) and other dealers.

No end in sight?

The bad news for Apricot appears to be continuing, with the closure of the **Beaugrand Datentechnik** operation in Germany that Apricot bought less than a year ago.

But as with much of the reporting of late about Apricot, this actually looks more like Good Sense than Bad News. Apricot was running two parallel operations in West Germany, with Beaugrand Datentechnik operating out of Frankfurt and selling mostly WP systems while Apricot GmbH in Munich dealt more conventionally with dealers along the lines established in the UK.

Beaugrand Datentechnik was a slightly odd group of some 23 specialised dealers selling the Alphatext WP package on a variety of machines with odd names like PRIMO and Super Smily (principally the Fortune 32.16). Alphatext has been converted to run on the Apricot and apparently does so quite happily. But running parallel operations, one of which was effectively selling direct to end-users, is not really what the New Realism demands of Apricot. So it got the chop.

Apricot will still support Alphatext, though, and the rationalisation will apparently produce a profit this year in Germany — losses on Apricot's European activities were another of the factors cited in the disappointing mid-term report.

Updates to reviews

Two amendments to products we featured in the last APRICOT FILE:

- we reviewed **Keele Codes' CLIP** hard disk backup utility — we're told the **latest version** (3.0) claims to work twice as fast and add some new facilities. We liked the package, and at £109 (inc VAT and P&P) it's cheap. More from 0782 629221.
- our look at MS-DOS tutors included a solid endorsement for the **Hands-on MS-DOS** package from **MicroCal**. We're told that the **price has now gone up** to £65 (still sounds good value) and MicroCal has a **new address** — PO Box 561, Slough, Berks SL3 9DP.

Our new address

We've fixed up a PO Box number that provides us with a much shorter and simpler address. The original address will still get to us, but the mail will arrive a day or so sooner if you use this one:

PO Box 509
London
N1 1YL