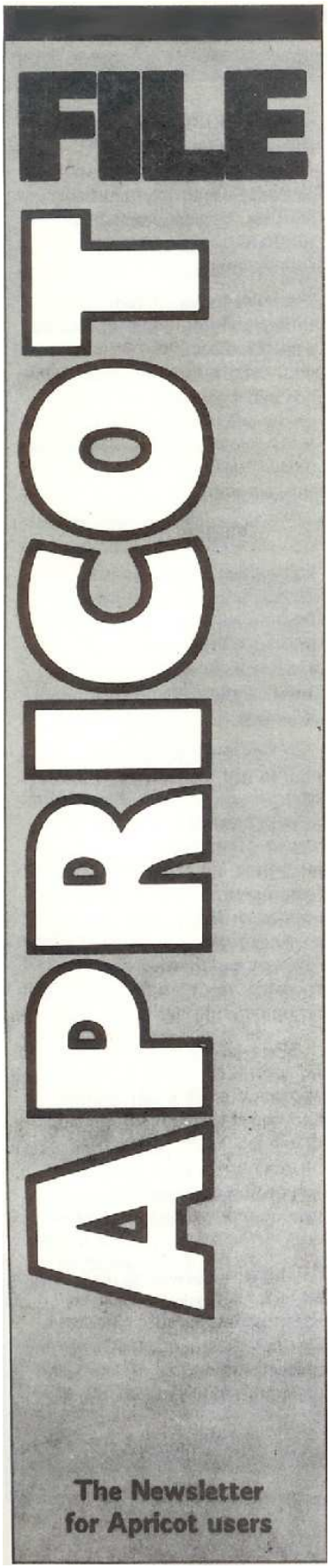
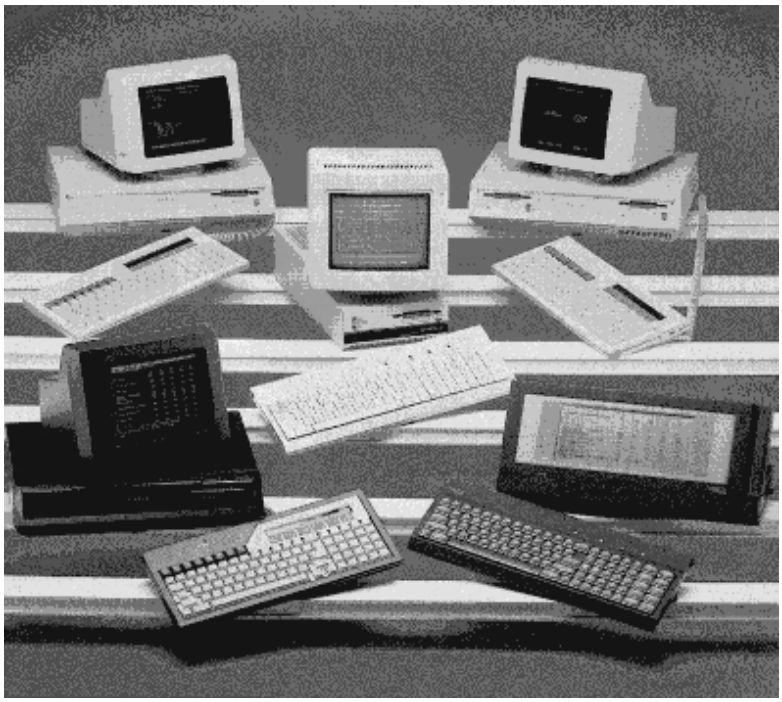


Apricot File 2_1



Scans courtesy of www.actapricot.org

Copyrighted content republished with thanks and permission from:

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

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

Release 2.1

FILE APRICOT

The Newsletter for Apricot users

October 1986

CASH customers

'Seal of Approval' scheme for accounts packages: software houses act

By Henry Deckhand

The brand spanking new Committee of Accounting Software Houses (CASH) held an inaugural meeting on 29 September. The 19 attendees reckon to represent 70 per cent of sales in the accounts package market — names like Action File, TABS, Tetra, Omicron, Pegasus, SMB, FACTS, Anagram and CMG.

The first aim appears to be establishment of a standard test procedure for accounts software, a kind of 'seal of approval' validation process; and it's obviously a reaction to the moves we reported a couple of issues ago whereby the Department of Employment (Lord knows why the DoE is involved) has given the National Computing Centre a contract to develop and operate just such a validation procedure — the Institute of Chartered Accountants will jointly administer the scheme with the NCC.

In general, we wholeheartedly approve of this kind of standardisation: 'approval' would guarantee standards of design, content and useability which are all too lacking from essential day-to-day software like ledger packages. But the validation procedure will be lengthy, and it will be expensive — getting your package approved is expected to cost between £3,000 and £8,000 a time.

Assuming some concensus on the nature and content of the testing, the short-term problem is the likelihood of a bottleneck in the validation process. The original plan was that just four people from the NCC's Microsystems Centre advisory operation should run the scheme: and the vetting is expected to take "three to five weeks" per package. Allow four weeks per application, and assume that a quarter of the 1,000 or so accounts packages on sale in the UK will be submitted. It doesn't take much pocket calculating to see that the queue could be extensive.

Two or more additional teams have been suggested by the NCC to remove any bottleneck. But re-validation will probably be necessary every time a package undergoes a major upgrade and every time the Chancellor changes statutory and tax requirements, so they'd almost certainly be running to stand still.

In fact it looks as though CASH was formed specifically to lobby against the potential delays in the process, and an early meeting with the NCC has been fixed up. Those members we spoke too all commended the scheme ... and all doubted its practicality.

The *ideal* might be independent testing: but for the industry and for the user, we feel that the more realistic option might be a kind of self-validation. That way the NCC/ICA team would develop the standard with contributions from CASH and others. This would be published and promoted heavily to establish its value in the minds of the buying public. The software houses themselves could decide whether or not to conform, and whether or not to claim conformity. And the NCC could subsequently vet any complaints from the public.

This would be quicker, more practical, and probably easier to implement and to administer. It could also work ... ◆

In this issue

The Workshop: XQT files for SuperWriter 3
When the Apricot beats the Cray 6
Notebook: MS-DOS 3.1 — what's it all about 9
The Workshop: Graphics on the F1 11
Writers in this Issue 13
Review: low-cost stock control by Dean Software 14
Software Round-Up: news and views 16
Deals For Readers 18
The Q&A Clinic 22
Review: HWa's £50 Fortran Utilities disk 23
Free Small Ads 26
Readers' Writes 27

About the newsletter

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

Except in the normal course of day-to-day contact, sub sales and the like, we are not affiliated with or supported by Apricot plc or any of its dealers. Nor for that matter are we sponsored by or directly associated with any user group or any other such 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 Publisher does not necessarily agree with the views expressed by contributors, and assumes no responsibility for errors of reproduction or interpretation in the subject matter of this newsletter or from any results arising therefrom.

Permissions

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 **by post** for queries on Editorial, Subscriptions, and DEALS FOR READERS at:

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

For **personal calls** and **deliveries**, our office address is:

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

Telephone **01-833 3501**

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

Telex:
265871 MONREF G quoting reference 81:JET767

Subscribe!

From Release 1.2, the annual subscription to APRICOT FILE buys 12 monthly issues as follows:

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

There are two ways to subscribe. Your subscription order can be accompanied by payment *in Sterling*, with cheques and money orders made payable to 'TP GROUP'.

Or use a Girobank transfer — our National Girobank account number is **59 028 1909**. We welcome this method of payment, in fact, since it's (a) cheap for both parties and (b) quick. It's particularly good for overseas customers: transfers can be made through CCP, POSTGIRO etc.

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 take Apricot, Victor and IBM PC disks. And we can handle most WP formats, but it's safest to send an accompanying hard copy 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 so that the copyright on all contents of APRICOT FILE is held by the Publishers or by the Publishers and Authors jointly. On the other hand, that's just to protect you and us: if you want to retain copyright, just say so.

■■■ ISSN 0268-6066 ■■■

The Workshop

All about .XQT

Part One: SuperWriter Execute files

By Gareth Hardwick

Using SuperWriter or SuperCalc without understanding Execute files is a bit like driving across a field without four-wheel drive — ok until you come to the muddy bit.

There has been some reference to .XQT files in previous issues of APRICOT FILE. But, for reasons best known to themselves, Apricot did not include documentation on these files in the version of SuperWriter originally supplied with the machines or in the Super Pack. If you buy SuperWriter to run on an IBM, an Amstrad or practically any other computer, you will get the full Sorcim/Computer Associates manual ... but not if you bought from Apricot.

In this article we'll look at the application of Execute files in SuperWriter. Next month we will see how they can be used with SuperCalc.

Getting started

Well, what are Execute files? Briefly, they are files — sometimes referred to as 'macros' — that enable you to repeat a given sequence of keystrokes by just typing in the name of the file. If you're clever you give the file a name consisting of only one character; that way you don't have so much to type. In addition to the file name, an Execute file has to be given the extension .XQT for it to be recognised by the program.

To use Execute files to best advantage you need to be a reasonably experienced user of SuperWriter, but there are many straightforward applications that anyone can use.

The Execute file is created like any other text file, but certain special keystrokes are used to indicate control characters and suchlike. The table at the end of this piece lists the keystrokes to type for some special characters.

Everyone knows that it's good practice frequently to save on disk the document you are working on. As a simple example let's see how this can be handled by an Execute file. First, list on a piece of paper the keystrokes you would normally use to save a file and return to the edit mode.

Esc	to access Escape options
S	to go to the 'Save' menu
Return	to Save the document to disk and return to the main menu
Return	to select the 'Edit' option at the main menu
Return	to select the default document to edit

Now choose a suitable name for the Execute file you are going to create — as it's to Save a document, call it S.XQT. Using the table of special characters, create S.XQT as you would any normal text file and you should finish up with these three lines on your edit screen:

```
^[s~
~
~
```

(Note that the special characters ^[and ~ are displayed in half intensity.) Now save S.XQT in the normal way. That's all there is to it.

Try using S.XQT. In the edit mode, call up a document you're working on. Hit the Escape key and then type 'X'. You will see this message:

```
Type name of Execute file:
```

Type 'S' and Return. Then just sit back and watch the screens flash by as the Execute file does the rest for you.

In fact this Execute file does not save you many keystrokes — you just use four instead of five — and you finish up at the top of the document. Let's see if it can be improved.

Improvement

In many cases you will be revising a document at a point somewhere in the middle of several pages of text. It would be nice if the cursor would return to the exact character you were last editing (this is known in some other programs as 'Save and Resume'). You could do this by inserting a marker character at the cursor position; and after the document has been saved, you could replace this marker with a null character or nothing. As the marker character will be saved with the document it is best to use one that is not printed, like a Spelling marker.

Take a piece of paper and list the keystrokes to do this:

Control-Q 2	to choose 'Insert Mark' options
S	to insert a Spelling marker
Escape	to access the Escape options
S	to go to the 'Save' menu
Return	to Save the document to disk and return to the main menu
Return	to select the 'Edit' option at the main menu
Return	to select the default document to edit
Escape	to access the Escape options again
R	to get the 'Replace:' prompt
Control-Q 2	to choose 'Insert Mark' options
S	to insert the Spelling marker as the character to be replaced
Return	to get the 'with:' prompt
Return	to indicate a null character as the replacement
Return	to initiate the replacement

Go back to the original S.XQT file and, using the table of special characters, edit it so that you have six lines on the edit screen like this:

```
^Q2S^[s~
~
~
^[R^Q2S~
~
~
```

Again, the control characters ^Q, ^[(for Escape) and ~ (for Return) are at half intensity.

Save S.XQT in the normal way. Call up another document to the Edit mode and try S.XQT. This time four keystrokes will replace 16.

Don't forget when you use S.XQT that it inserts a Spelling marker in the text at the point you left off editing. A Spelling marker will not affect the way the document is printed; however, if the worst happens and your computer does go down, the copy of the file on disk will have this spelling marker in it. It is, of course, easily deleted by hitting the Escape key followed by 'D' then 'S'. When you have finished editing your document you should save it in the normal way.

More uses

Execute files can be quite simple. They need not include any Control characters or other special characters at all. For example, I use an Epson dot matrix printer and like to use double-strike mode for the final copy of documents; the trouble is I can never remember the appropriate 'Out' command to use with SuperWriter. So to make life easy I have an Execute file called DS.XQT that consists of the one line ...

```
\out 27,71~
```

When I want a document to be printed in double-strike, I simply put the cursor at the top of the document, press Escape followed by 'X', and then enter 'DS'.

Another use of Execute files is for short 'boiler-plate' phrases to be stored and inserted simply into documents. The file 1.XQT might just consist of the phrase "Thank you for your letter enquiring about our products". To insert this, you could just hit Escape followed by 'X' and then enter '1'.

You could do the same sort of thing by using the 'Include' option after hitting the Escape key. However, one of the advantages of using Execute files is that the cursor ends up at the end of the text that has been added.

A key feature of Execute files is that you can halt execution at any point, insert additional keystrokes from the keyboard, and then continue with the sequence in the Execute file. This is done by inserting the ampersand character in the .XQT file. When it comes to the '&', the Execute file will suspend operation until you type "&" at the keyboard.

A useful example of this is in an Execute file to insert a comment in the Document History. The file, which I call C.XQT, looks like this:

```
^[~
^K~
^J~
^K^K~
&~
^[^[~
```

The effect of these characters is as follows:

^[~	[Escape, Return]	select Main Menu
^K~	[up one line, Return]	select UTILITIES
^J~	[down one line, Return]	select DOCUMENT History
^K^K~	[up two lines, Return]	select Update Comment
&~	[pause, Return]	pause to input data; '&' to resume
^[^[~	[Escape, Escape, Return]	select default document to edit

To use C.XQT you simply hit Escape, type 'X' and then enter the filename 'C'. At the Document History screen you type your comment and follow it with the ampersand character instead of Return. You will then be returned to your document and the cursor will be where you left it.

Remember the comment cannot be more than 24 characters. Note too that the comment will not be saved until you save your document.

One of the drawbacks of SuperWriter is that it is only partially a WYSIWYG (what you see is what you get) word processor. In particular, the effect of embedded commands for things like headers and footers cannot be seen on the edit screen. This Execute file enables you to format on screen and then quickly to return to your document in the edit mode. The file I call F.XQT consists of six lines:

```
^Q2S^[~
^J^J~
^J~
&^[^[^[R^Q2S~
~
~
```

To use F.XQT, hit Escape and 'X' and then enter the filename 'F'. You will be whisked through to the Format On-screen mode and the Execute file will then pause. Press any key to scan through the document in the normal way; when you've seen enough, hit the '&' character. The program will return you to your document at the point you left it. (Notice that this file inserts a Spelling Marker to mark the place and this is saved to disk when you Format On-screen). At the end of the editing session, save your document in the normal way.

After the file pauses at the ampersand it does not, of course, know what you have done. If you have overshot the Format On-screen and returned to the main DOCUMENT COMPOSER menu, re-enter Format On-screen before hitting '&'.

Instant execution

If you wish to use SuperWriter in the normal way from the system prompt (I assume you do not use ACTIVITY or MANAGER!) you usually just type 'SW' followed by Return. But if you follow 'SW' with a space and then the name of an .XQT file, SuperWriter will follow the instructions in the Execute file when you press Return.

As an example, here's DBAK.XQT — it enables you to check through your backup files and delete those you do not want.

```
^K~
~
^J^J~
*.BAK~
&^[^[^J~
```

You use DBAK.XQT by typing "SW DBAK" at the system prompt. The name of each file with the extension .BAK will then be displayed in turn and you will be asked to say whether you want it deleted — enter 'Y' for yes or 'N' for no.

When you've been through all the .BAK files you will find you are at the 'Display document' prompt. Hit '&' to return to the system prompt.

Execute files can be used to access other .XQT files or to repeat themselves. One of the cleverest and simplest ideas for a SuperWriter Execute file came from Lindsay Doyle and was first published in the newsletter of the Apri-Soft-Exchange User Group: he called it 66.XQT and it should consist of 65 ^J sequences followed by &^[X66~.

This can be used for controlling page breaks by placing the cursor at the top of the document and then entering Escape X 66. The cursor will move down to line 66. A suitable place can then be chosen to put in a page break with Control-P P; and you then hit '&' to repeat the process. When you've finished putting in the page breaks, hit Escape X Escape to exit from the Execute file.

Obviously you need to make sure that the line width on the edit screen is the same as the line width you will use for printing.

Where you can go with Execute files is only limited by your imagination. Get SuperWriter into four-wheel drive and try a bit of cross-country motoring! ◆

Copyright 1986 Gareth Hardwick

Special characters for .XQT files

Character	Effect	Keys to type
^[	Escape	<INS MARK> ^ ESC
~	Return	Return
^H	left arrow	<INS MARK> ^ H
^J	down arrow	<INS MARK> ^ J
^K	up arrow	<INS MARK> ^ K
^L	right arrow	<INS MARK> ^ L
^P	Print directives	<INS MARK> ^ P
^Q	Function keys	<INS MARK> ^ Q
^T	top of document	<INS MARK> ^ T
^F	right one word	<INS MARK> ^ F

... and so on

In addition, you can cause a delay in the execution of an Execute file by inserting the character ^D — each ^D will usually cause a delay of a quarter of a second.

Notes:

The ^ is the backslash and shift. When you type ^ after the MicroScreen <INS MARK> key, nothing appears until you add the character. All special characters will appear at half intensity.

For Apricots without MicroScreen keys, the INS MARK function is obtained by hitting Control-Q followed by 2.

Inserting the character '&' at any point in an Execute file causes the sequence of keystrokes to be suspended at that point and control to go back to the keyboard. The Execute file sequence recommences when '&' is next entered from the keyboard.

To exit an Execute file when suspended hit Escape X Escape.

User's view

Precision, accuracy and the right answer

Why the Apricot beats the Cray

By Edward N. Bromhead

A couple of weeks ago I went to a conference at which a distinguished Professor stated that you only get the right answer if you use a big computer (costing more than £10 million!). Small ones just didn't give the correct results.

This got me thinking deeply about the whole business of precision, accuracy, and what is the right answer. The following article is the result.

Basics

Memory on any computer is a series of 'on-off' switches. In an Apricot (or for that matter, millions of other machines) these are organised into groups of eight and called 'bytes'. To store any information, you take as many bytes as you need and set the little switches to represent that information.

This of course presupposes that you have a coding and decoding system. As far as computing is concerned, one of the greatest intellectual leaps of all time was the realisation by Von Neumann that the information stored inside the memory of the computer could equally well be program (code) as data (coded).

Now, if on=1 and off=0, the switch pattern can be viewed as a binary number; and the eight 'binary' switches in a byte can be set in 256 different ways — 2 to the power 8 different combinations. If we wished to hold information which represents *characters*, we could store one of 256 different characters in a byte. Since this exceeds the number of characters in (most) alphabets, a byte is a good character holder. Indeed, if you just have upper- and lower-case letters plus punctuation marks, you probably don't need to use up all of the switches — or bits — in your byte.

In the same way, if a byte was used to store a program *instruction*, we have a potential instruction set of 256 different processor operations. We won't complicate matters by suggesting that the instruction in a particular byte could be somewhat conditional on the values in preceding bytes; but that would then open up a whole new range of possibilities. The mind boggles.

Logical data (TRUE or FALSE) only needs one byte: and strictly, only one bit is required. (I wonder if we could have 'relatively true or false on a scale of 1 ... 256'?)

Numbers

It's when we come to *numbers* that bytes aren't good enough. Since we can have only 256 different bit patterns, we can only cope with 256 different numbers — and they all have to have the same sign (positive), or else we'd have to use up one of our precious bits to represent the '+' or '—' value. For most numbers we also need to know where the decimal point lies, and this robs us of more bits.

Assume for the start that we're storing only integer numbers which don't have a decimal fraction. Even then, we will certainly need more than 256 different numbers.

The answer is to use more bits, and that means more bytes. Two bytes (a 16-bit word) can store -32768 to +32767. For three bytes (a 24-bit word — not common today, though ICL mainframes used this at one time) the range is plus or minus 8½ million. Four bytes gives about 4,000 million possible values, five allows an enormous range.

But there are only a limited number of sensible strategies for holding an integer. By contrast, we have much more flexibility in storing *floating point* numbers.

Assume for now that these numbers are stored by using some of the bits to store the characteristic, some to hold the mantissa. For those to whom this is Greek, the number is represented in the form ...

```
0.345678 x 10x10x10 ... `n' times
```

The mantissa is the .345678 part, and the characteristic is 'n'. As an example, if 'n' was 4, we would have `0.345678 x 10x10x10x10`, or 3456.78. (In fact, both the mantissa and characteristic will be in binary; but take them in decimal for the time being, as this makes things easier to comprehend.) We also need sign bits for both the characteristic and the mantissa — a negative characteristic implies dividing by 10 'n' times.

Taking a certain number of bytes, we can trade off bits in the characteristic for those in the mantissa. That effectively changes the precision with which a number is held in inverse ratio to its range.

One further refinement is whether or not we attempt to hold the '0.' part of the mantissa: normally we don't. Also, numbers in the mantissa need only be chosen in the range 0.100000 ... to 0.9999999 .... That's because 0.0999999 can be shifted with the appropriate modification to the characteristic. This is called 'normalisation'.

Remember also that all this is done in the binary system, and is just a coding system for the binary switches in the bytes allocated to store the number. The Apricot itself makes no assumption about how data is coded into the bytes, that is entirely software dependent. (There is, however, a standard for this coding: the IEEE standard. Of course, there's no guarantee that a particular piece of software adheres to this standard — or any other.)

Typically, we use at least four bytes to store a single floating-point number; and we can employ six, eight, ten or even more in particular circumstances. Let's examine the sort of precision and range that each of these options gives us:

No of bytes	Significant figures	Range per data value ie 10 to the power of
4	typically 7	-38 to +38
6	10 - 11	-76 to +76
8	16 - 17	-308 to +308
10	about 20	-2466 to +2466

Most languages for micros give us the option of using the four- or eight-byte forms: readers of APRICOT FILE will know that I am a Fortran enthusiast, and one compiler — Digital Research Fortran-77 — allows the ten-byte form too (but only in special circumstances).

Bugs

So where is all this leading us? Well, if we realise that the computer is only holding real numbers approximately, it becomes easier to see why some errors creep in. Take a simple test: make a loop to add 0.0000001 to 1.0 ten million times. The result *ought* to be 2.0, but it isn't; the answer you *do* get will surprise you. (Do it with four-byte integers for the loop count, and four-byte reals). Repeat the test first by taking eight-byte reals, and then by doing a multiply instead of the repeated add.

In the first case, adding the 0.0000001 was effectively adding zero each time: it didn't matter that you did it so many times, the answer was still 1. Even with the better precision, the answer still isn't perfect; and it's nowhere near as good as the multiply — especially if you do *that* with eight-byte precision too.

The end result of this is a set of observations:

- The answer is better if you use higher precision
- The errors are dependent on the algorithm (calculation sequence) used
- Addition of very different sized numbers is particularly bad

Frankly, four-byte precision is just on the margins of acceptability for many purposes. Sometimes it works and sometimes it doesn't. It's fine, for instance, if you want to locate the coordinates of a pixel on the screen, or a pen on a piece of paper — even two-byte integers can be good enough for that. What may be fine for playing computer games, however, often isn't good enough for many engineering calculations.

Survey the English Channel and do the calculations in four-byte precision, and the two sets of tunnellers will almost certainly miss each other: locate the position of an aeroplane with respect to the National Grid in four-byte precision, and it will be ok in the Scillies but you will surely hit a mountain when flying across the North of Scotland. No, a better precision is certainly required.

So look at eight-byte precision. I'll give you an example of what this can do. It can represent the distance from the Earth to the Moon to better than 0.00005mm in the 400,000km distance — or to better than a thousandth of the diameter of a speck of dust! Do we need that? And to make it worse, every variable in the program takes up twice the space, the program will be longer, and will run slower. Damn! (I also looked at 10 to the power 308, to see what that implied — I somewhat doubt if there's that many atoms in the observable Universe ....)

The ideal compromise is six-byte precision. All British mainframe computers used this (as 48 bits) from the original Atlas until the industry was destroyed by the ineptness of politicians and suicidal tendencies among its own staff. When using six bytes, your results are always 'correct' — meaning 'more or less good enough' — without too much worry at the coding stage, and programs are as compact as they can reasonably be expected to be.

Now look around the micro scene for six-byte precision — any Fortrans? No. Any versions of Basic? Only one, the interpreter on the BBC micro (which also has five-byte integers — wonderful!). Any Pascals? Yes, Turbo Pascal (those Borland geniuses get it right again: when are they going to do a Fortran for the Lost Boys in the Never-Never Land of engineering?).

Just holding a number in encoded form to a fixed precision causes a loss of accuracy. Now do some calculations with it, and the errors grow. This is because we do not compute to enough digits, and get an additional error or 'round-off'. All computations are subject to this to a certain extent, but some computational procedures are more prone to error than others. (A few trigonometry exercises on a right-angled triangle with one sharp apex using Pythagoras or the sine and cosine formulae will convince the numerate: it won't matter to anybody else!).

Now comes a big difference — do you have a numeric co-processor on your Apricot or not? If you do, this is what happens. Your application takes your 32- or 64-bit number and expands it to a special 80-bit form for the numeric operation in the co-processor. The rounding down to the appropriate precision is done after the completion of the computation. Effectively, this 80-bit internal form has a 64-bit mantissa.

Mathematically, this is much better than many supercomputers (among them the much-vaunted Crays). So the eminent Professor was talking through his pipelined processor! Some supercomputers use a 60-bit data form, others degrade precision to achieve extended range. The Cray, for instance, has only about 15 decimal places of precision. The Apricot is at least ten times better — not that it matters ...

If you do not use a co-processor, the main processor has to be instructed just how to do the operation. This takes a number of steps, and you

have a choice for how you do it. When using 32-bit numbers (24-bit mantissa) you can stop at 24 bits; or you can carry on and round later. With 64-bit reals, you can stop when the 53-bit mantissas have been dealt with, or go on and completely emulate the co-processor's function.

This is a software choice; and even if the high-level language programmer doesn't have control over this (you do with some compilers), the author of the compiler itself has. Stopping the arithmetic short accelerates the growth of round-off errors and needs to be watched, although it does speed up the computations. 'Short' arithmetic is faster than 'long' if you don't have a co-processor; but the difference is very small, and has more to do with fetching and carrying the data in eight-byte blocks rather than four-byte chunks.

The last thing that causes problems with precision is the Trojan Horse left by the compiler writer. In most microcomputer languages, 32-bit precision is the default. So to operate in 64-bit precision, variables need to be TYPEd explicitly in the program — and so do all constants. In the Fortran context, this means writing *every* constant with a D exponent; for instance, '0.5' becomes '0.5D0' and '1.0E-8' becomes '1.0D-08' — and so on. Otherwise, the compiler is likely to interpret this as an invitation to insert single-precision code there in the program.

Putting expressions in subroutine calls is another bugbear. Some languages compute the result to single precision before passing it over *even if the result is obtained by operating on double-precision variables and is picked up as double-precision in the subroutine.*

Single-precision trigonometric functions are also a pitfall: Fortran 77 is supposed to be context-sensitive such that it will put in the correct precision function when a generic name is used — but many Apricot implementations aren't.

Last words

My friend the Professor had forgotten that his carelessly-coded programs simply weren't portable between supercomputers (with a default precision of 60 or 64 bits) and other computers where the default was less.

The essence of getting the right answer can be summarised as follows:

- precision
- control of round-off
- selection of an appropriate algorithm
- careful coding

The Apricot (particularly when equipped with a co-processor chip) is capable of such precision in its calculations as to exceed virtually everyone's needs. But this can be degraded quite easily by carelessness that means the end result is just not good enough for any purpose. The solution? Programmer vigilance. ◆

Diary dates

On with the Shows

A round-up of potential relevance

October 1986

1-3 IBM System User Show *London.* Mostly for users of IBM's larger machines, but many PC-oriented suppliers will be at this show. *Organiser: EMAP, tel 01-608 1161*

15-17 Computer Graphics *London.* Conference and exhibition at Wembley, covering everything from business graphics to CAD/CAM via animation ... *Organiser: Online, tel 01-868 4466*

16-21 Orgatechnik *Cologne.* Big exhibition (should be over 1,500 stalls) on anything and everything for the office. *Organiser: Messe und Ausstellungs, tel Germany (0221) 8211*

21-23 AMTEC 86 *Bournemouth.* Advanced Manufacturing Technology show apparently aimed at a local audience. *Organiser: Martin Ashenden Associates, tel 01-262 3382*

21-24 Business Equipment Show *London.* Miscellany of products and services for the office; organised (as always) with BETA, the Business Equipment Trade Association. *Organiser: BED Exhibitions, tel 01-647 1001*

22-23 BOS Exhibition *London.* Small show dedicated to BOS Software's products: purpose-built operating system, lots of standard business packages, much specialised vertical software — all available on "over 80" diffierent computers, claiming "over 8,000" users in the UK. *Organiser: Focus Investments, tel 01-222 9090.*

27-29 Interface Europe *Paris.* One of an international series from the organisers of Comdex — aimed at large-scale corporate and government visitors, presumably featuring high-grade high-priced comms and office automation system. *Organiser: Interface Group, tel USA (617) 449-6600*

November 1986

10-14 Comdex Fall *Atlanta.* The more pleasant of America's crucial twice-a-year expos for dealers and VARs. Required visiting for industry-watchers, and a damn sight more palatable than the Spring version in Las Vegas. *Organiser: Interface Group, tel USA (617) 449-6600*

11-14 COMPEC *London.* The long-established COMPEC has come back rejuvenated as one of the largest and best shows for micro users. It retains its bias towards dealers and system builders, but there's still a lot there for the average punter — especially among the smaller exhibitors. *Organiser: Reed Exhibitions, tel 01-661 4900*

18-20 Computers in the City *London.* Conference and show for financial and other City companies. *Organiser: Online, tel 01-868 4466* ◆

Technote

MS-DOS 3.1

Should you upgrade from 2.11?

By Simon Potter

MS-DOS version 3.1 is now the standard operating system being shipped with *all* Apricots. That's obviously to guarantee a degree of compatibility across the families, though in practice the BIOS differences impede true compatibility.

If you're very unlucky, you might find that the VER command shows you have version 3.0 or 3.1, in which case you should ask your dealer for an update (instant check: the CLS clear screen command doesn't work on these early versions).

So what's so special about the new version? And should you move on to it?

Bug fixes

To start at the bottom, MS-DOS 3.1 fixed a few of the minor bugs in 2.11, notably an eccentricity that prevented MS-DOS from COPYing every 256th file — very irritating if you're using a homemade BATch file to do a bulk backup of a lot of small files on a hard disk.

Incidentally, the single-disk switch (COPY /S) is no longer available — you have to use SCOPY.

Directories

Subdirectory handling has been improved, particularly for applications which betray their origins in MS-DOS 1.25 (or even CP/M) by not recognising subdirectories and paths at all.

You can now run a program outside the current directory *without* first using a CD or a PATH command. The operating system will accept a string like \SYSTEM\PROGRAMS\SC3 to run SuperCalc 3 from a sub-sub-directory on the default drive.

The new external command APPEND is also useful. It's a complement to the PATH command which completes the instruction to MS-DOS about where it should try looking for files — PATH works only for executable files (those with a COM, EXE or BAT suffix), which isn't much use for the many applications that require overlays (.OVL-suffixed files) to be in the same subdirectory. APPEND lets you set a path for *non*-executable files like overlays, so you'd ordinarily have an AUTOEXEC.BAT that included both commands:

```
PATH = A:\PROGRAMS
APPEND = A:\PROGRAMS
```

SUBST is a similarly handy facility. Several of the more common applications can't handle subdirectories directly — you can't specify a pathname for a CardBox FIL file or a SuperCalc CAL worksheet, for example. Some more or less circumvent this by allowing you temporarily to step outside the program to use the MS-DOS 'change directory' command; that will redirect the program's interest, but it's not exactly an easy and elegant solution. SUBST lets you specify a disk drive identifier instead of a pathname; since those programs are able to handle different drive IDs, that gets around the problem. So before starting WordStar or whatever, you could have a command like ..

```
SUBST E: A:\PROGRAMS\WORDSTAR\LETTERS
```

Then you'd tell WordStar to save and load new documents from drive E. You can see that SUBST might also be handy to avoid repetitive typing of long pathnames.

Unless your CONFIG.SYS includes a LASTDRIVE command, the Apricot will assume that you have no more than five drive IDs available — A and B as actual disks, C probably as a RAMdisk, D and E available for adding on real drives or for using with SUBST.

The curious JOIN command reverses the effect of SUBST: it replaces a drive identifier with a subdirectory path. It's quite clever — it will create a subdirectory for you if the named one doesn't exist — but it isn't documented anywhere and we don't know anyone who's found a use for it.

CONFIG.SYS

LASTDRIVE is one of the two new entries you can put into CONFIG.SYS: you use it to tell the Apricot to recognise more drives — it's of most relevance to Point 32 networks, which come pre-configured with LASTDRIVE = Z.

The other one, FCB, will also be useful only for Point 32: it specifies the number of File Control Blocks that can be open at once. FCBs were/are used by some older programs (notably dBASE II) to control files. The default number of FCBs is four, and that seems adequate for most programs.

Disk space

Also quite important is the way MS-DOS 3.1 handles allocation of disk space. When you DELete a file with 2.11, the operating system re-uses the freed space as soon as possible: 3.1 avoids using it until it's absolutely necessary.

This means it's a lot easier and a lot safer to recover accidentally-deleted files: there's a better chance that the clusters in question haven't been overwritten by any subsequent saves.

New commands

Then there's a collection of miscellaneous new commands and utilities, some of which are well worth having ... though in some cases you can get the same facilities (or better) from our DEALS FOR READERS disks.

- You get *real* commands for FORMAT and DCOPY without the necessity for using slow, boring, multiple-keystroke front-ends; that also applies to BACKUP and RESTORE, which incidentally appear to be bug-free at last
- There's a LABEL command which lets you put a 'volume' label on an already-formatted disk. The FORMAT program includes the facility to specify a label for a disk as part of the formatting, but with LABEL you can change it subsequently
- ATTRIB does part of the job that our CHMOD public domain utility manages to complete: it sets file attributes so that a particular file has read-only or read-write status (the latter is of course the default). CHMOD goes further; you can use it to set *any* of the available file attributes, most usefully the 'hidden' status which means that the file's details won't appear in a DIR listing
- WHAT is a silly little gismo that tells you the version number of a specified program.

Networks

For the longer term, you should be aware that the *major* development in MS-DOS 3.1 was the inclusion of multi-user primitives that provide a standardised interface between network control software and applications programs.

This looks like being a *real* standard, at least for the time being. Before 3.1, most local area networks used a different interface between activity on the network and the operation of multi-user packages. So the software developers had to produce different and customised versions of their packages for each LAN they elected to support. But every LAN that supports MS-DOS 3.1 must use its in-built interface: and any multi-user applications software written for 3.1 will run on any LAN that offers compatibility with 3.1 — nearly all the major LAN suppliers are offering that compatibility now.

As far as the software houses are concerned, 3.1 standardisation means that they can reach a large market for multi-user packages with a single copy of their software. The LAN vendors are happy enough now that some order has been brought to the marketplace — in many cases, just about the time when most of them wanted to release a version 2 of their product anyhow to include their competitors' better features.

MS-DOS 3.1 itself is not a network operating system: you need several other functions — notably a file server program, which manages disk I/O and general network security: and a 'redirector', which intercepts requests from applications programs and routes them accordingly.

There are two ways to provide those extras. The classiest example of one solution is represented by Novell's NetWare: here you have a proprietary multi-tasking operating system which actually treats MS-DOS as a subsidiary task. The main job of MS-DOS then is to run a file server, a shared disk system; but because overall control is vested elsewhere, you can have multiple file servers on a Novell network — and a file server doesn't have to be a dedicated system, it can also be used as a workstation on the net. Novell has also taken the opportunity to build in extra functions, and its approach gets around the sluggishness of MS-DOS particularly when handling file directories.

The alternative approach is to use MS-NET. This incorporates MS-DOS 3.1, but it still isn't really a true network operating system — Microsoft sells it as an OEM product, expecting the buyers to customise and enhance it before something is shipped to the end user. This is what Apricot did to produce Point 32, and it's what IBM has done in producing its PC Network Program.

The MS-NET file server is run as an application of MS-DOS, and this means that the relative simplicity for the computer maker has to be weighed against some substantial deficiencies. For one, MS-NET requires that the file server is dedicated: you can't use it for anything else. And its dependence on MS-DOS 3.1 at the file server makes for bottlenecks, unnecessary overheads, and restrictions on expandability.

Counterpoints

Back to the Average Apricot User. On the downside, MS-DOS 3.1 is a bit bigger than 2.11 (about 61K for MSDOS.SYS plus IO.SYS, against 50KB); and, more important, some commands seem a lot slower. This is particularly marked with TYPE when you want to scroll through some text.

A letter from subscriber Chris Chipperfield in a recent APRICOT FILE reported a RS232 handshaking bug — RTS/DSR information appears to be ignored.

And should you upgrade to 3.1 anyway? Well, you might be attracted by the subdirectory handling and the extra commands ... but you can live without them, and besides there are MS-DOS 2.11 utilities available commercially (like SmartPath) and from DEALS FOR READERS (like DPATH and CHMOD) which will do the same kind of things.

If you are considering moving to an Apricot network you have no choice — Point 32 (and indeed any system based on MS-NET) *requires* 3.1. Otherwise, the best reasons for shifting to 3.1 come into the category of keeping up with the Joneses: all new Apricots (and IBMs, and many other machines) have 3.1. And shortly we'll start seeing applications software that demands 3.1 and which won't run on 2.11 ... ◆

The Workshop

Direct access to F1 graphics

Programming the graphics hardware

By Richard Hughes-Rowlands

Programming the Apricot Xen is a recent Apricot document subtitled 'a compatibility guide for software authors'. It includes the following advice:

> DO NOT PROGRAM DIRECTLY TO THE HARDWARE

This is equally applicable to all machines in the Apricot range, and (for what it's worth) it carries my recommendation too. Having said that, however, this article is all about how to program the graphics hardware of the F1 directly.

Why program the hardware directly, and what are the alternatives? Lindsay Doyle's APRICOT FILE series of GSX Workshops have shown how to program the GSX graphics extension, a comparatively portable way of writing graphics programs for the entire Apricot range (and for other PCs too). The introduction of BIOS 3.1 across the Apricot family has also improved matters, providing graphics compatibility at a lower level: this is achieved through a standard interface to a loadable device driver, extending the ROM BIOS's capability into graphics.

So why program the hardware when these two options are available? If your application is purely text-based, there is no need to use the hardware directly. And if your application has to interface to GEM or Microsoft Windows, you have no choice other than to avoid the hardware.

But programming the hardware can give you speed, and a greater flexibility in the manipulation of the display. The price for this is reduced portability (because it has to be hardware-specific) and increased programming difficulty (for example, there's no support for circle-drawing and the like).

Graphics hardware overview

The *Apricot F1 Technical Reference Manual* gives all the details of how to program the graphics directly. This article clarifies and explains this description with some examples in Turbo Pascal.

This description is based on a monochrome F1 with BIOS 3.1, the only machine to which I have access; machines with earlier BIOS versions may be slightly different. Much of the Workshop will also be applicable to the rest of the F series, and to F series machines with colour displays as well. The F series graphics hardware is also quite similar to that of the Xen.

The graphics hardware consists of three major components:

- the memory-mapped Display RAM
- the Pointer RAM
- the Palette RAM.

The F1 has a display resolution of 640 by 256 pixels (200 for 60Hz, and for BIOS 3.1), and makes no distinction between text and graphics display. Up to four levels of greyscale can be displayed simultaneously on the monochrome monitor, or four from 16 colours on a colour monitor.

(At a lower resolution of 320 by 256 pixels, 16 greyscales and 16 colours are displayable at once. This option is not discussed here.)

Pointer RAM

The Pointer RAM controls the sequence in which each row of 640 pixels is displayed on the screen. The idea behind the Pointer RAM is to provide a quick scrolling of any section of the display, simply by changing the RAM contents.

The Pointer RAM is a 512-byte table of addresses: each two-byte word in the table contains the 16-bit address of a row in the memory-mapped display area. The table is located at $01E00; the address of the first row in the display is at location $01E000, the second row is contained in location $01E002, and so on. The display refresh hardware automatically reads this table and displays the 640 pixels from the location addressed by each of the 256 entries in the RAM.

Figure 1 is a Turbo Pascal procedure to save the memory image of the screen to a disk file. A very similar routine could be written to restore the screen from disk. Note that the Pointer RAM is also written to the data file; this ensures that the order of each display row can be restored as well.

Figure 1: SAVESCR.PAS

```
procedure savescr;
var
  fd: file;
begin
assign(fd,'screen.img');
rewrite(fd);
blockwrite(fd,memw[0:$1e00],4);
blockwrite(fd,memw[0:$2020],320);
close(fd)
end;
```

Figure 2: SETUP.PAS and PUTBACK.PAS

```
procedure setup;
var i: integer;
begin
for i := 0 to 255 do
  begin
  ptrsav[i] := memw[0:$1e00+i*2];
  memw[0:$1e00+i*2] := $1010 + $50*i
  end
end;



procedure putback;
var i: integer;
begin
for i := 0 to 255 do
  memw[0:$1e00+i*2] := ptrsav[i]
end;
```

The two procedures in Figure 2 manipulate the Pointer RAM. The first saves the current state of the RAM into an array called ptrsav, which must be declared in an outer block. The Pointer RAM is then initialised in sequential order. Note that the value written into the Pointer RAM is the address in the Display RAM divided by two. Each 640 pixel row takes $A0 bytes of memory.

The second routine, putback, simply restores the Pointer RAM to its original order. This is required to avoid confusing MS-DOS and the BIOS.

Display RAM

The memory-mapped Display RAM is located at $02020, and is divided up into a number of planes. For the monochrome monitor each pixel in the display is represented in two planes, mapping on to four greyscales.

One word (two bytes) of the memory map denotes eight pixels: the first pixel is represented by bits 15 and 7, the second by bits 14 and 6, and so on to the eighth by bits 8 and 0.

So for example to change the state of pixel 0, the first pixel in any word, you must change *only* bits 15 and 7. These two bits become a two-bit binary number — 00, 01, 10 or 11. This is used to index the Palette RAM.

Figure 3: PIXEL.PAS

```
procedure pixel(x:integer;y,level:byte);
var
  addr,plane: integer;
  n: byte;
begin
n := x and 7;
case level of
  0:plane := $8000 shr n;
  1:plane := $0;
  2:plane := $8080 shr n;
  3:plane := $80 shr n
end;
addr := ((x shr 3) + memw[0:$1e00+2*y]) shl 1;
memw[0:addr] :=
  (memw[0:addr] and not($8080 shr n)) or plane
end;
```

The Turbo Pascal procedure PIXEL.PAS takes three parameters. The first two are the x and y co-ordinates of a pixel on the screen (in the range 0 to 639 and 0 to 255 respectively). The third, which can be 0 to 3, is the greyscale colour.

The origin (0,0) is the top left corner. The variable n is used to calculate the pixel bit number within the word; then a case statement works out what new value of the plane is to be placed in the display word (the address of the display word is calculated by indexing the y co-ordinate out of the Pointer RAM).

The selected word is then fetched, the old value ANDed out and the new value ORed in.

Figure 4: SQ.PAS

```
procedure sq(x,y,z,sz:integer);
var i,j: integer;
begin
for i := x to x+sz-1 do
  for j := y to y+sz-1 do
    pixel(i,j,z)
end;
```

SQ.PAS is a simple routine to draw a square using the previously described PIXEL.PAS procedure. The parameters are the x and y co-ordinates of the top left corner, z is the greyscale colour, and sz the size in pixels. This isn't the fastest way to draw a square, but it *is* very simple ...

Palette RAM

The output from the memory map plane (0, 1, 2 or 3) addresses the Palette RAM. The Palette RAM is responsible for mapping this index into one of four greyscale colours; it is a write-only block of 16 four-bit words located at $E0000 (you can't read the current state).

Figure 5: PAL.PAS

```
procedure pal(grey,map:byte);
var a,b,c,d: integer;
begin
case grey of
  1: begin a:=$00; b:=$02; c:=$08; d:=$0a end;
  3: begin a:=$04; b:=$06; c:=$0c; d:=$0e end;
  0: begin a:=$10; b:=$12; c:=$18; d:=$1a end;
  2: begin a:=$14; b:=$16; c:=$1c; d:=$1e end
end;
memw[$e000:a] := map*2;
memw[$e000:b] := map*2;
memw[$e000:c] := map*2;
memw[$e000:d] := map*2
end;
```

For example, simply by changing the entries in this RAM, the whole of a black-background screen can be changed to white (or rather, to green). This is done instantly without having to redraw any pixels.

PAL.PAS shows how to manipulate the Palette RAM. Each of the four greyscale colours has four corresponding entries in the RAM; the case statement works out the addresses of each four using the greyscale as an input. The value for the map is written into each of the four locations.

With the monochrome display the values of the map should be ...

0, 4, 2, 6, 1, 5, 3, 7

In this order the greyscale colour goes from brightest to darkest. The standard settings are 7, 3, 6, 0 for the greyscale values of 0, 1, 2, and 3.

Putting it all together

SPIRAL.PAS uses some of the procedures given earlier to provide a (simple) demonstration of the F1's graphics capability.

Programming directly to the hardware is usually a last resort. It can however provide an increased speed: it does give access to facilities otherwise not available (Palette RAM, pixel by pixel access): and it *is* feasible. ◆

Figure 6: SPIRAL.PAS

```
program spiral;
var
  t,r,x,y: real;
  d,k,m: integer;

{$i pixel.pas}
{$i sq.pas}
{$i pal.pas}

begin

{ clear the screen and turn the cursor off }
clrscr;
write(chr(27),'x5');

t := 0.0; k := 0; m := 1;
while (t < (3.1415 * 4.25)) do
  begin

  { Calculate the x,y co-ordinates and radius }
  r := 3.4*sqr(t);
  y := r*sin(t);
  x := r*cos(t);

  { Draw a square }
  sq(round(0.35*x)+275,round(0.2*y)+105,k,m);

  { Select next colour and size }
  k := k + 1;
  if (k = 1) then k := 2;
  if (k = 4) then k := 0;
  m := m+1;

  t := t +(3.1415*2)/30
  end;

{ Switch the colours until keypressed }
d := 300;
while not(keypressed) do
  begin
  pal(3,7); pal(2,6); pal(0,0); delay(d);
  pal(3,0); pal(2,7); pal(0,6); delay(d);
  pal(3,6); pal(2,0); pal(0,7); delay(d)
  end;

{ Reset the palette map }
pal(0,7); pal(2,6); pal(3,0);

{ Clear screen and bring back cursor }
clrscr;
write(chr(27),'y5')

end.
```

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 9000 BOOK (very much thicker), THE ELECTRONIC OFFICE – A MANAGEMENT GUIDE (fairly thick), three titles about Commodore home computers (one thick, one a bit thick, one pseudonymous), and THE GOOD COMPUTING GUIDE (*really* thick and very cheap too at £2.95 — or £3.00 inc P&P from DEALS FOR READERS).

Edward Bromhead is Reader in Geotechnical Engineering at Kingston Poly, where he uses a couple of Apricots to model the behaviour of soil slopes and groundwater (mainly using his own Fortran software and GSX). He specialises in earthworks (eg for motorways) and landslides (he predicted the collapse of Carsington dam in November 1983, it fell over eight months later).

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.

Michael Dirom was completely ignorant of anything to do with microcomputers until he purchased an Apricot PC in October 1984. Chronic heart disease forced him into early retirement in June 1985. He formed the Southern Apricot Group in October 1985 as a friendly self-help user group for those finding little help from the manufacturer, its manuals and dealers; the Group meets every two months to discuss problems and exchange knowledge. He lives off the beaten track in South Wiltshire.

Lindsay Doyle 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.

Roger Gann works full-time running the DEALS FOR READERS section of the newsletter as well as contributing words, ideas, humour, hints, legal advice, spittle for mass-mailings, etc. He was formerly the principal writer on 16 BIT COMPUTING; before that he was a builder (and a pioneer Sirius user), a lawyer, and a kind of grandiose bank clerk.

Gareth Hardwick runs WIRDS (Worldwide Information on Rural Development Sources), an information service for development workers in Third World countries. He is also a part-time lecturer at the Microcomputer Advisory Centre, Polytechnic of the South Bank.

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

Review

Stock control for the smaller business

Dean Stock Control

By Jack Flatau

Any businessman must be confused by the wide variety of accountancy packages on offer. To start, we can distinguish broadly four different types.

- The most popular packages are split into **modules** so that you can implement one section of the work at a time. When you want to introduce the next stage, the output of one module becomes the input data of the next module. So you can start with invoicing, and introduce a computerised sales ledger or a stock recording program as an integrated system when your office is ready for it. A nominal ledger and final accounts system and sales analysis can come later still.
- The **incomplete records** system lets the trader store up sales and purchase invoices and then input these on say a monthly basis to produce periodic accounts. This would not be appropriate to a trader requiring up-to-date stock figures.
- Some accounting systems are based on **spreadsheet** programs. This simplifies new developments, but leaves the system less well controlled.
- A **stand-alone** system has just one or two functions which cannot be linked straightforwardly with any other system.

Dean Stock Control falls into the last of these groups. On the face of it, the stand-alone approach might seem too restricted: but it does have advantages. Because there are few operations, the programs are smaller and faster. More important to the busy manager, you need less time to learn how to use it.

And this is the great strength of Dean Stock Control: I do not think I have ever mastered a package so easily.

Dean Stock Control is designed for small businesses that need a simple stock recording and control system. An invoice production system can provide input to the stock records. It provides no link to an accounting system — Dean Software, the supplier, offers a package called EASIVAT for those who need full accounts.

Dean Stock Control can hold 2,000 inventory lines. You can accommodate 300 supplier or customer details.

Setting up the system

The Dean software arrives on a single-sided Apricot disk with 167,424 bytes free: it's not a big package. The existing files provide space for the data about stocks, customers and suppliers, so the system should be workable on a single-drive machine. It worked ok on my F10, on floppy or hard disk, and I could copy files and data from one to the other.

Having booted up, you type 'STOCK' and are asked to specify four VAT rates which can be used when invoicing. You are then presented with the main menu.

```
Add an Item ..................... 1
View, Change, or Delete An Item .... 2
Record Sales of Stock ............ 3
Record Receipt of Stock .......... 4
Stock Report ..................... 5
Re-order Report .................. 6
Sales report ..................... 7
End Program ...................... 8
```

The first activity will normally be to enter your stock items. The data to enter is ...

- CODE, which is used to call up the item for updating
- DESCRIPTION of up to 40 characters
- SUPPLIER CODE (three characters)
- CURRENT STOCK LEVEL in the range -9999 to 32000
- RE-ORDER LEVEL
- COST PRICE
- SELLING PRICES (you may have three different prices per item)
- CURRENT MONTH'S SALES in units and value
- SALES for each of the last three months

There is a defined area for each data field and pressing the Return key moves you to the next entry. If you enter an integer in the value fields the decimal point and zero pence appear automatically. When you have finished the entries for an item, you are asked if the details are correct; a 'Y' response sets up the details on the stock file (though there is no file readable by the operating system, the data being held within the Basic program). 'N' resets the cursor to the top of the screen. Then you either press Return, to retain the data: or you make the change required.

Once you have entered all your stock items (subsequently you may be entering additions) the program will index the stock items by code number, and offer you the chance to follow one of six options: the first five are called by the initial letter.

- Next (steps forwards through the stock list)
- Last (steps backwards)
- Change
- Delete
- Get (displays an item by reference to the indexed stock code)
- MENU (for other options)

To enter details of customers (for invoicing) or suppliers, the MENU key is used.

Entering customer and supplier data follows a similar pattern, though strangely the form uses a different style. The manual says somewhat unhelpfully that "details on how to use the customer/supplier program are given in the EASIVAT manual". This was not supplied for the review, but the operation is so simple that its absence was no problem.

Using the package

With Dean Stock Control, sales and receipts start by entering the code (the difference between receipts and sales updates is that the latter require a choice of sales prices). If you forget the exact code, you can search for it: for instance, entering 'AP<' will give you the details of the first code starting with 'AP'. By typing 'N' (Next) you can run through the product codes in alphabetical order until you find the correct one. (And 'APRF10' will come before 'APRF2', which is correct in logic).

At the end of the entry the stock level is updated automatically. For sales the sales figures are also adjusted. If you run out of stock, a message is displayed; you don't get one however if the stock has only reached the re-order level.

Stock sales can also be triggered by producing an invoice, starting with the 'Miscellaneous Menu' and selecting invoice or credit note options in 'Run the Invoice Generator'.

In either case the three-letter code for the customer is entered first and brings up the name and address. If the customer is not on file, you can enter the address yourself at this point.

Stock details are called up by the code, the quantity entered, VAT rate and selling price selected. When every item sold is entered and confirmed, the values, VAT, and invoice total are calculated and then printed.

There is no provision for a separate delivery address on the invoice. And surprisingly, there's no way of printing your own company name at the top. The assumption is that you will use your own letter-headed paper; but small users often use continuous stationery all the time, since the machine may be used for a number of different applications consecutively

Despite the provision for suppliers' names and addresses, you are not able to generate a replenishment order on this system.

A number of standard reports are available:

- **Stock List**
- **Price List** for each price or all prices
- **Reorder report**, showing all items down to reorder level or below
- **Sales Report** for current month, last three months, and average
- **List of Customers and Suppliers** (jumbled together in alphabetic order, because that is how the file is kept)

The sales statistics will only be useful to people with fairly steady and regular sales. Most small firms will want to take an average of weekly rather than monthly sales as a guide to their buying.

Which Should you buy?

The Dean package does its job efficiently and speedily. But there is a lot that it does *not* do.

Most obviously, it does not link with a full accounting system. Invoices can't have your company name or a customer's delivery address. You can't enter a special price — only preset prices are used. Sales reporting is rudimentary.

And the stock re-ordering system is primitive: one would not expect a sophisticated Box-Jenkins algorithm or exponential smoothing to be included, but the ability to specify, say, a number of weeks' sales as the reorder level would be a sensible use of the computer.

But the most important deficiency is the lack of an audit trail. This is demanded by computer professionals and accountants: it means that you should be able to follow the path of a transaction from entry to final total. The Dean Stock Control system does not retain any transaction file (a list of transactions with dates); so if you suspect that something has gone wrong with one of the results after some months and some hundreds of entries, there is no way of finding the error and rectifying it.

Still, this is an extremely polished package for a simple application at a modest price (£145). The question is: will it fit your needs — and those of your accountant?

Fortunately you can find out by purchasing the Demonstration Disk for £19.50. The program is so simple to use that it might be worth trying this demo package just to see whether it does match your requirements, to get experience of computer accounting, and to establish what you really need in the way of stock control. ◆

Dean Stock Control

Description:	Stock control package
Supplier:	Dean Software
Address:	Unit 5 The Mews Brook Street Mitcheldean Glos GL17 0SL
Telephone:	0549 542643
Price:	£145
Configuration:	any Apricot
Conclusion:	Well-written and effective, though with some deficiencies. Could be useful for very limited requirements.

Software Round-Up

News and views of what's on sale (and why)

How do we do it?

We're often asked how APRICOT FILE is produced. Well, basically it's a desktop publishing operation: we prepare articles, include the formatting and font changes, and print page masters on a laser printer. These are then handed to an offset litho printer as camera-ready copy and he prints the newsletters conventionally.

The key element, of course, is the software we use. It's called NEWSWRITER and is a complete production system designed specifically for newsletters and reports — multi-page documents in a more or less fixed format, in other words. And after a year's development, with us and the *Financial Times* newsletters as Beta-test sites, the package is now ready for the market.

Tom Sealy of FT Newsletters calls it "a superb system", and we'd go along with that. Our page design and the layout formats we use work the software pretty damn hard, and we have very few complaints: it's possible for a two-fingered novice to produce quality output in a matter of hours.

Like most true desktop publishing systems, NEWSWRITER is a high-class word processor with some fancy options for formatting. So the basics will be familiar to anyone who's used a word processor. NEWSWRITER was written by Friedman Wagner-Dobler, a friend of and occasional contributor to our newsletters, and anyone who knows his stuff (like BusiPost, The FTU, MXDUMP) will know the programming quality to expect.

And as well as being a software author, Friedman is a high-grade computer-industry journalist who also runs his own company: so he knows what he's talking about — many desktop publishing packages appear to have been produced with little concern for the practical applications. NEWSWRITER was written primarily for journalists and others who understand the production process, but that's largely a matter of familiarising yourself with a few terms like 'proofing' and 'kerning' — it really isn't difficult to use.

Some desktop publishing systems force you to become a paste-up artist, designer and writer rolled into one. NEWSWRITER lets you concentrate on the words and the artwork will take care of itself: a headline is called a 'HEAD', a byline is called a 'BYLINE'. Font changes can be displayed on-screen, and you can see where the line breaks will be in the printed copy — so what you see is what you get.

NEWSWRITER really scores in what's called 'copy-fitting' — making your words fit into a given space. This NEWSWRITER does for you at the press of a key by adjusting the spacing between lines and paragraphs; no more subbing, no more last-minute rewrites to force the text to fit. Very few desktop publishing packages can offer that.

NEWSWRITER is not a hasty adaptation of an existing program — it was written as a desktop publishing system from the start. Nor is it a half-finished product rushed to the market to cash in on the desktop-publishing hype: it's been in the field for a year and is used by several organisations, including Europe's largest newsletter publisher (FT newsletters).

What's more, Friedman's company Cognita has years of experience in the publishing field. As he puts it, "We've already solved the problems our competitors don't talk about — remote production, demand publishing, and database preparation, for instance".

NEWSWRITER runs on the IBM PC/XT/AT and clones under PC-DOS or MS-DOS, and under Xenix for the PC/AT. As for the Apricot, Friedman says he'd be able to produce a version quite quickly if there was enough interest. As it stands, NEWSWRITER should work ok with Xenix on the Xens and (perhaps with a tweak or two) under MS-DOS on the Xen-i.

NEWSWRITER costs **£666.66** excluding VAT. At that price it has to be one of the best desktop publishing options on the market: call **01-221 7612** for more details.

Incidentally, we're running a small-scale NEWSWRITER **bureau service** for the production of brochures, price-lists, (non-competitive) newsletters and similar material using NEWSWRITER and our equipment. We can offer a full production service — including design, writing and printing too if you need that — at extremely competitive prices and with very fast turn-around. You're holding a sample of our work ... Call Dennis Jarrett on **01-833 3501** if you're interested.

Dealing with dealers

The annual *Business Micro Dealer Directory* from market researchers **Romtec** was published a few weeks ago. Among other interesting facts Romtec notes that nearly 60 per cent of dealers turn over less than £500,000 a year — and, not surprisingly, around one in ten are going out of business every year.

Recent withdrawals include the last two remaining **W H Smith Business Centres** at Crawley and Hounslow (the Reading store had already closed). W H Smith has already discontinued its efforts to sell business computers alongside home micros in its retail outlets.

At least **MBS** appears to be on the road back. The group, which describes itself modestly as "Britain's leading independent computer supply and services company", had a rough ride in the year to December 1985 — largely because it hadn't accommodated the sudden changes in the market for box-shifting on low-price micros (courtesy of Amstrad and all those el cheapo Far Eastern PC clones). This year MBS dropped that business in favour of more value-added system sales and underwent the usual rationalisations and other attention to overheads; the remedy seems to have worked, for interim half-year figures to June show an operating profit of £124,000 on a £38.8 million turnover.

The equivalent midpoint in 1985 did have a £2 million profit on £34 million sales, but by the end of the year MBS turned in a £462,000 loss on £67 million. The interims still show a net loss of £1.3 million because of interest charges; but the company also had a successful rights issue in June, which apart from anything else must have represented a vote of confidence from investors ...

Apricot has just signed MBS to sell the **Xen-i**; that means ten MBS branch offices join the 200 or so systems suppliers already punting the product. **DDL**, the distribution arm of MBS, will also be middlemanning the Xen line to dealers.

Incidentally, Romtec shows **IBM** continuing to dominate the market with a 43 per cent share, followed by **Apricot** (17 per cent) and **Olivetti** (7.9 per cent).

More tea?

Software house **Iansyst**, hitherto best known for a very fine typing tutor, has come up with a clutch of neat-sounding programmer's aids under the name **VICAR**. That stands for 'Variables In Context Analysis and Reporter'; we've seen a Beta-test version, and very impressive it is too.

VICAR began life as a fairly conventional cross-referencer to track the names and usage of variables, but adds the ability to handle multiple source files (handy for applications which can extend across several programs) and importantly shows the *complete* context for a variable — not just a line number. There are versions of VICAR for many popular languages; for Microsoft Bssic and Turbo Pascal the price is **£49**, for other Pascals and dBASE III it's **£79**, for C and dBASE III you'll pay **£129**. We asked for review copies and will report; meanwhile we have just added the product to our DEALS FOR READERS lists at those prices plus VAT.

Copy protection

Every time someone comes up with a new copy-protection mechanism to prevent unauthorised duplication of pruchased disks, someone else immediately takes up the challenge of beating it.

Well, here's another challenge: **Kynd Programs** reckons it has "the best media protection package available in the Apricot marketplace". It's called **DEFENSOR**, and employs the funny-formatting approach. You use it to format your distribution disks with a user-specified security code being applied as part of the process; then you compile your programs with a linkable object-code module supplied, the purpose of which is to check for the presence of the security code at run-time.

A disk formatted with DEFENSOR can't be copied on a track-by-track basis (with DCOPY or DISKCOPY) — you'll get a 'data error reading drive *n*' message. And file-by-file copying with COPY *.* won't port over the security code, so the programs won't run either.

There's a recovery program called CUSTOS which should be able to handle customer disk failures. As for security backups — well, you'll just have to issue duplicates to the purchaser yourself.

Info from Kynd's marketing people on **0274 496805**.

Not-so-trivial pursuit

Harold Gale is chief executive of the high-IQ society MENSA, so if he puts together a computer game you'd expect something a bit special. Well, he has: and it's a kind of *Trivial Pursuit*. "An entirely new concept in microcomputer entertainment ... Addictive and educational" says the publicity person at Harold Gale Associates. We'd query the first claim, but it does indeed engage the mind.

The format of **MICROTRIVIA** is modest enough. You play the computer at a version of noughts and crosses: a question and a choice of four answers is put on to the screen, you select one, and if you're right the computer gives you a 'win' entry on the grid — otherwise it hands itself one. The next question and answer choices then pop up. A winning line for either side stops the game and you get a score plus a comment on your performance.

Scoring is the clever part. For a start you pick a time limit for yourself at the beginning of the round: if you don't answer quickly enough, the computer dumps on you. The score is controlled by the speed of victory, the number of questions it took, and the time limit you chose. Score 150 or over and you get to a special 'competition' screen: this qualifies you to compete for a River Nile holiday for two next Autumn — it's worth £1,600.

It's a bit grandiose to claim that "our aim is to make learning fun", though obviously a game format with multiple-choice questions could have educational applications. In the meantime MICROTRIVIA comprises a £25 game disk with a thousand 'general knowledge' questions on it; and there are ten additional specialised-subject disks, each with another 1,000, at £10 apiece. Titles of these will be familiar to *Trivial Pursuit* players — pop music, sport, TV and radio, geography, the living world, and the like.

At those prices (which include P&P) we can recommend MICROTRIVIA (and the Nile might be nice too). What's more, if you quote APRICOT FILE when ordering you'll get the starter disk for **£20**. Harold Gale Associates is at **Bond House, St John's Square, Wolverhampton, West Midlands WV2 4AH**.

Sparks fly

The slightly daft "that's *our* software!" row between Systems Union and Simdell appears to have simmered down, not least because (a) the case looks difficult to prove either way and (b) the built-in tardiness of the British legal system is great at turning down the heat. But there appears to be a potential successor: **Tetra Business Systems** is considering legal action against **MS Associates**, and issued a fun press release to tell the world about it.

MS Associates has a Basic-to-C converter called CGEN; and MS is run by **Keith Maskell**, who used to work at Tetra. A while ago Tetra converted its accounts software from Basic to C, and for that it wrote a package called C-Base. "We believe that CGEN is in fact the same as our product C-Base" says Tetra MD **Sean Dowling**.

On the other hand, "although Maskell was part of the original C-Base project team, the project was completed without him ... Maskell made a very limited contribution to the development of the converter, and in fact he told me at the time that it couldn't be done!" This one could run and run. ◆

The Mail-Order Pages

Deals For Readers

Public Domain software, utilities, hardware bargains

This is our mail-order se .ion, exclusively for readers of APRICOT FILE. **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 Apricot hardware and software**, typically useful items that aren't widely available
- **anything else we think would be of value to readers**, such as relevant books.

Each issue has new entries and (occasionally) price changes. So keep looking! And we're always open to suggestions for additional items — if there's something you'd like to see in DEALS FOR READERS, **or if you've come across a handy utility or a good deal on hardware add-ins, let us know.**

How to order

Simple: just send us a note or a purchase order 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. There are two ways to pay: a Sterling cheque or money order (made out to "TP GROUP") *can* accompany your order, or use a Girobank transfer — our National Girobank account number is **59 028 1909**.

We can accept purchase orders *only* from companies and other organisations, and *only* 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.

Sorry, but we cannot accept telephone orders.

The prices quoted include VAT where applicable 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 well 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? And if the ideas and techniques therein would make a useful mini-article for the newsletter, we'd certainly be interested in printing it.

Public Domain Disks

These disks contain public domain programs and text files which we have sorted, selected and tested. In many cases we have debugged, amended and documented the material too.

Public Domain Library

Order no:	AP000
Price:	£5

In response to popular demand — all the contents files for all our Public Domain disks for a fiver.

Apricot Disk No. 1

Order no:	AP001
Price:	£15

Over 40 files. Highlights now include:

AUTOEXEC with an improved prompt
SDIR and FDIR sorted-directory utilities
VDEL and DELQ selective DELetes
WAIT and ASK two batch file commands, more useful than PAUSE
GRAFCHAR define your own characters
ESCAPES list of Escape codes for Sirius and Apricot
OTHELLO good version of the board game
LEADTRAK and ACCTRACK two sales lead management systems.

Apricot Disk No. 2

Order no:	AP002
Price:	£15

Includes 79 files Highlights·
ALTER and CHMOD change file attributes
BASICKEYS single-key entry of Basic commands
DEBUG the Apricot version
EPSON sets alternative print modes
KNEW copies only new files
MENU easy batch-file menu system
MOVE moves files from one subdirectory to another
RENDIR rename subdirectories
SORTDISK sorts directories
TREEDIR displays a DIRectory of all files on a disk — in their subdirectories
WHEREIS searches disk subdirectories for named file(s).

Apricot Disk No. 3

Order no:	AP003
Price:	£15

A bumper bundle. Highlights:
SWEEP all-purpose file housekeeper
SCAVEN finds and locks out bad sectors
MEMDUMP browse through memory
TURBOKEY key redefinition for aficionados of Turbo Pascal
SUPERWRITER XQT FILES a useful clutch of macros.

Apricot Disk No. 4

Order no:	AP007
Price:	£15

More indispensable tools and utilities. Highlights:

ASMGEN dissassembler
132-column screen configurator
Memory speed test benchmarks
Improved version of SETUP
DIARY year-planner
LABEL address label printer
BANNER prints fancy letterheads on daisywheel printer.

Apricot Disk No 5

Order no:	**AP009**
Price:	**£15**

A total of 38 files plus full documentation — includes John Lyth's imaginative MS-DOS menu front end, a suite of dBASE II programs to control your bank account, CHECKER to verify inputting in MSBASIC, DEBUG.PAS, a clutch of Spellbinder utilities, IBMBASIC (BASICA to MSBASIC converter) ... and much, much more!

■■■ NEW ■■■

Apricot Disk No 6

Order no:	**AP018**
Price:	**£15**

Another disk chock full of utilities, 30KB of BATch file extras, essential MS-DOS utilities like DPATH (works like PATH but for data files like overlays and Help text — no more duplicating SuperWriter/SuperCalc/WordStar OVL and HLP files across subdirectories), the latest versions of SWEEP and WHEREIS, excellent archiving utility ARC, and a handful of programs designed to enhance your data security — even a few programs for inveterate hackers!

GSX User's Disk

Order no:	**AP005**
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, many graphics demo programs, and Lindsay's improved HEXPAWN game IMPHEX. Now with an amended screen driver capable of supporting the Epson FX-80.

Games No. 1

Order no:	**AP004**
Price:	**£15**

Includes the very tough flight simulator 747SIM (fly your own Jumbo) plus old favourites like AWARI, TOWERS OF HANOI, and many more besides. A very full disk.

Games No. 2

Order no:	**AP006**
Price:	**£15**

Includes CLEWDO (solve the mystery), TRUCKER (get a cargo across the States on time), and some games for the younger Apricot user (SPELL and HANGMAN). Plus for the first time a GWBASIC version of STARTREK (noises and all) and BACKGAMMON. Some of these are unsuitable for F Series and Portables.

Games No. 3

Order no:	**AP008**
Price:	**£15**

Only two games on this disk, but both excellent and quite realistic entertainments. GREAT BRITAIN LTD is a political-economic simulation — be Prime Minister, balance the books, win the election if you can. And INHERITANCE is an interesting investment/gambling game; you could get hooked on both!

GWBASIC Games

Order no:	**AP010**
Price:	**£15**

A miscellany of graphics-based games, some old and some new (some borrowed and some blue) — including some special sound effects and graphics demos. Highlights: OPERATOR (try running a large mainframe computer system), hunt for oil with WILDCAT, depress yourself with NUKE-LA and ALIEN.

Chess

Order no:	**AP016**
Price:	**£15**

Only three games — but what games! For the first time on an Apricot, a challenging public domain chess game with superb hi-res graphics — even supports an 8087 if you have one. As an hors d'oeuvre we include a version of PACMAN, for dessert a golf game. A self-booting disk with a free bonus: the latest version of MS-DOS with BIOS 3.1. Now comes with a special chess character set for the Portable and F Series.

The dBASE II Disk

Order no:	**AP017**
Price:	**£15**

Full of dBASE II program suites. Includes programs to control membership records, a dBASE routine library, a label printing program, the ubiquitous mailing list program, a product tracking system, a form letter generator and sundry dBASE utilities. Essentially for keen dBASE programmers.

Programmer's Disk 1

Order no:	**AP011**
Price:	**£15**

Our first disk for programming fans. Star attraction is a public domain implementation of LISP called XLISP; plus fifteen Turbo Pascal routines and utilities, a C cross referencer, an MS-BASIC programmers' toolbox, a program line indenter and useful batch file utilities.

Programmer's Disk 2

Order no:	**AP012**
Price:	**£15**

Features BSU, a program to unprotect protected MSBASIC programs; plus Ron Cain's Small C compiler, a trio of cross-referencers for Basic and C, an ASCII-to-Hex converter, a GWBASIC core dumper, the appendix that Microsoft omitted from the MACRO-86 manual, and a C source code checker.

Lotus 1-2-3 Disk No 1

Order no:	**AP013**
Price:	**£15**

No less than 25 WKS templates for 1-2-3, including the Asterix suite of business worksheets, and the macro library MACLIB.WKS.

Lotus 1-2-3 Disk No 2

Order no:	**AP014**
Price:	**£15**

A heap of technical notes and external utilities for 1-2-3 plus seven WKS files including an address label macro and PRNTMENU.WKS to manipulate your Epson printer.

PC File III

Order no:	**AP015**
Price:	**£15**

Our first 'shareware' program — an Apricot version of the well-known general-purpose database/filing package. Menu-driven, great documentation (over 100K of it on the disk), superb power and speed, very easy to use; handles everything from mailing labels to multi-page reports. Exchange data with 1-2-3, MailMerge, many other WP and spreadsheet programs. Indexing on all fields, searches includes 'soundex', multiple selection criteria ... Complete with the latest IBM emulator (which is required). For BIOS 3.1 only.

Special software

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

Order no:	**AS001**
Price:	**£25**

FAST is an accelerator for the Apricot screen: it bypasses certain screen

interrupts. Simply running it in an AUTOEXEC.BAT will cause your Apricot to handle screen text up to *twice* as quickly. 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. Not available for Apricots with BIOS 3.1, the F Series, or Portables.

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. You must specify which printer you have when ordering — POLYPRINT comes customised for particular printers.

Polyprint Plus

Order no:	**AS021**
Price:	**£20**

For existing POLYPRINT users only: 13 new typefaces and additional facilities including the new capability of mixing typefaces in the middle of a document.

The Apricot Catalogue

Order no:	**AS004** [ASCII]
Order no:	**AS005** [Cardbox]
Price:	**£50**

A list of all the software and hardware we know of for the Apricot 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. Regularly updated by us; currently well over a megabyte of information on six disks. The ASCII format can be read by most word processors or simply TYPEd, the Cardbox format can be read by Cardbox Plus and comes with a .FRM format (but not with the file manager program itself).

A.S.D. 1

Order no:	**AS006**
Price:	**£57.50**

Formerly known as P.A.S.D.: this is the latest (and much improved) version of the first (and only?) independently-produced package of upgrades to and tools for SuperWriter. A packed disk, and a good piece of work. 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
- keyboard configuration presets to speed up keying
- many XQT files to accelerate and simplify your routine work
- several batch files for disk housekeeping.

A.S.D. 2

Order no:	**AS006**
Price:	**£57.50**

A second bumper bundle of SuperWriter goodies from Ansible, essential for SuperWriting but full of generally helpful Apricot extras. Includes DOCTOR (easy-to-use file recovery), MAILER mail-list program, a whole bunch of useful .XQT and BATch files, a disk labeller ... some 90 files in all. Enthusaistically reviewed by Lindsay Doyle in Release 1.11.

AnsibleIndex

Order no:	**AS022**
Price:	**£86.50**

Sophisticated SuperWriter indexing utility, even permits indexing of subjects and themes. Can handle documents chained across more than one disk, too. Only requires two marks per index item.

AnsiblePatch

Order no:	**AS023**
Price:	**£15**

Patches SuperWriter to improve the legibility of displayed text on F series Apricots and Portables.

■■■ REDUCED ■■■

IANKEY Crash Course

Order no:	**AS007**
Price:	**£30**

The very best keyboard tutor we've seen for newcomers — beats TOUCH 'N' GO into a cocked hat. Sensible design based on serious research; clear on-screen instructions; takes only three or four hours to teach the basics of touch-typing. Our review concluded "I strongly recommend it as excellent value for money" — especially from DEALS FOR READERS, because the normal RRP with VAT and P&P is £58.65.

■■■ REDUCED ■■■

IANKEY Two-Fingers to Touch-Typing

Order no:	**AS008**
Price:	**£30**

An alternative keyboard tutor — converts a two-fingered 'hunt and peck' typist into a fast user; and it does so without wasting your time, forcing you to relearn the keyboard, or boring you stiff. Included in the enthusiastic review conclusions, and again available from us at a price £10 below RRP.

■■■ NEW ■■■

VICAR

BASIC:	**AS026**
Price:	**£56**
TURBO PASCAL:	**AS027**
Price:	**£56**
MS PASCAL:	**AS028**
Price:	**£90**
dBASE II:	**AS029**
Price:	**£90**
dBASE III:	**AS030**
Price:	**£148**
C:	**AS031**
Price:	**£148**

Variables In Context Analysis and Reporter: another good piece of work from Iansyst — a clutch of neat programmer's aids that began life as a fairly conventional cross-referencer to track the names and usage of variables, but adds the ability to handle multiple source files (handy for applications which can extend across several programs) and importantly shows the complete context for a variable — not just a line number. Note that there are versions for several popular languages.

FORTRAN Utilities

Order no:	**AS009**
Price:	**£50**

A collection of assembler utilities for MS-FORTRAN applications, complete with test programs and link instructions. Now features direct access to system calls and an interfaces to GSX. Also includes:

- set date and time
- current date and time
- get/change default drive
- delete/rename disk file
- read/get character from keyboard buffer
- print characters on printer.

SSDISK

Order no:	**AS024**
Price:	**£15**

Fed up with the inadequate RAMdisk software that came with your Apricot? S&S has produced a RAMdisk which is configurable up to a full 896KB with 128 directory entries per 'drive'.

DO-TO-ALL

Order no:	AS010
Price:	£15

Handy for hard disk users and anyone with multiple subdirectories: DO-TO-ALL lets you issue just one command that is to be applied to all directories — to delete all BAK files in your subdirectories, for instance.

TRAN-SEND

Order no:	AS025
Price:	£86.25

Replaces The FTU in our lists because it's cheaper and better. A superior file transfer package enabling you to transfer data between Sirius, Apricot and/or IBM via the RS232 serial port with speed and ease — and Viewdata, too. Handles subdirectories and features remote operation with the option of transferring files using XMODEM protocols. Price is for three disks covering those three formats.

The File Protect Pack

Order no:	AS018
Price:	£50

SECRET makes files invisible (or visible again) to DIRectory searches, READONLY sets the attribute bit so that they can't be deleted or overwritten, ENCRYPT encrypts and decrypts files according to your password, ZAP deletes them so thoroughly they can't be 'undeleted' at all.

UNDELETE

Order no:	AS020
Price:	£20

The essential brow-soother — will attempt to recover files that have accidentally been DELeted, and does so with gratifying frequency.

SPLIT Disk Partitioner

Order no:	AS021
Price:	£56

Infosoft's clever disk partitioning utility for Xi hard disk Apricots — lets you 'partition' the hard disk so that it looks like up to 24 separate disks, each with a drive identifier and each with a size you specify. Thereafter the 'drives' can be treated as if they were real ones for reading, writing, copying, renaming, deleting files etc.

MXDUMP

Order no:	AS012
Price:	£75

Another classy little number from the programming pen of Friedman — produces exact screen dumps from your Apricot display on a standard Epson MX dot-matrix printer. Once loaded it sits quietly in MS-DOS waiting for you to call it (with a single keystroke). Lots of options: specify text or high-resolution graphics, normal or bold, standard- or double-width: and more ...

The Lotus Pack

Order no:	AS013
Price:	£57.50

Useful utilities from S&S Enterprises for Lotus users:

- THEPOUND lets you use all the displayable Apricot characters in worksheets — including the pound sign
- SOLID123 consolidates several 1-2-3 or Symphony worksheets into a total
- TRANSPOS transposes rows and columns
- 123RANGE lists worksheet ranges.

The Printer Pack

Order no:	AS015
Price:	£40

Mainly for Epson-compatibles, mainly for spreadsheets:

- SIDELINE prints worksheets sideways so that wide spreadsheets fit on to the paper
- FXPOUND prints the pound sign on FX-type printers
- SETFXPR provides a set-up menu for printer functions.

The WP Pack

Order no:	AS016
Price:	£50

Four handy S&S utilities for word processing:

- WSASCII converts WordStar files to ASCII format so that they can be read by other word processors
- WORDNO word counter
- READAGE calculates the reading age (ie the comprehensibility) of text
- SEARCH searches text for a given phrase.

The First-Aid Kit

Order no:	AS017
Price:	£25

Just what you need to make your computer more palatable to novice uses — the DOSMENU front-end option-choice menu and the HELP dial-an-explanation facility.

The Games Box

Order no:	AS019
Price:	£25

Two games from Alan Solomon: ADVENTURE is the original 'Colossal Cave', CASH CRISIS is 'Space Invaders' with money

■■■ NEW ■■■

A La Carte

Order no:	AS032
Price:	£340

An exclusive offer to APRICOT FILE readers at £100 off the recommended price: A La Carte is a superior windows-based 'executive desktop' system which has a database management system, word processing and diary. In her review in Personal Computer World magazine, Kathy Lang called it "neat and easy to understand ... full of thoughtful and unusual features". All this and you can also use A La Carte to 'front-end' other software. To be reviewed in APRICOT FILE too. Requires a minimum of 512KB, works best on a hard disk. Not for Portables or F Series.

Other extras

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".

Expansion Memory

256KB	AH004
	£130
512KB	AH005
	£185

High-quality add-in memory cards from a well-known manufacturer at prices we think you'll find hard to beat — bearing in mind that they include VAT and despatch. The 512K board is particularly good value, perfect for RAM disks and big SuperCalc worksheets.

All prices include VAT and despatch. Send your order with payment to:

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

Queries: 01-833 3501

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 ...

Note that we can only accept payments, that we must have PREPAYMENT except for official Purchase Orders, and that PURCHASE ORDERS can only be accepted for orders totalling £50 or more.

We can accept payment in Sterling (cheque or money order): or use a Girobank transfer — our National Girobank account number is 59 028 1909.

The Q&A Clinic

Your questions answered

INT 21 and Trace

I'm still trying to fathom the deep mysteries of the Apricot Xi with the invaluable aid of DEBUG, APRICOT FILE, and a book on the 8086. What I would like **to know is why the Trace command goes crazy after executing an** INT 21**? Why doesn't the program follow on with the next instruction?**

Basically, the problem is that MS-DOS is not re-entrant. When you've Traced through an INT 21, you have a breakpoint in MS-DOS; then when DEBUG calls the operating system, MS-DOS has effectively been called twice — and that's not allowed. There is also some very weird stack-handling in the Apricot BIOS and that probably doesn't help either.

It is sometimes possible to set breakpoints in MS-DOS or BIOS code, but that is not reliable — and even if it does sometimes do the job, you may find that the display will work but not the keyboard (or whatever). So it's a bit like trying to catch a lift from a Vogon.

On the other hand, you might try a different debugger. We don't know whether such an animal is available for the Apricot (all suggestions and information welcome) but you can get them for most other machines like the IBM PC and even the Sirius. Basically they can do one of two things:

- skip over interrupts (this makes a lot of sense if you are debugging applications software)
- use direct hardware access to get round the re-entrancy problem.

PATH of righteousness

I had an F1 for a year, primarly for use with SuperWriter, and I was quite content with an arrangement whereby different disks were used for different classes of work. Each disk had an AUTOEXEC **to set up the F1 and SuperWriter for the appropriate task; with meaningful filenames and a 'memory jogger' in the 'author' spot, we could pick up a disk and just check the contents before getting on with the job.**

Alas, I was then tempted to by a cheap Xi hard disk machine. I dispensed with the floppies and started using subdirectories. But now I find that I must put a copy of SuperWriter (or part of it) in each subdirectory; that gobbles up memory, and there must be a better way. Using the PATH **command will direct the system to a master copy of SuperWriter, but the command 'SW' just brings up the complaint that the overlay files are missing. Lodging copies of** SW.HLP, SW1.OVL **and** SW2.OVL **in each subdirectory will work, but this does not seem very economical.**

Indeed it isn't. The problem with PATH is that it will only work for executable files, those with a .COM or .EXE extension.

There are two possible solutions, both of which will involve spending money with DEALS FOR READERS. First, you could pay £56 for the SPLIT disk partitoner and reorganise your hard disk to appear as multiple separate volumes; but it sounds as though you might run out of space and volume assignments (SPLIT will allow you only 24 individual volumes on the disk).

The preferrable (and much cheaper) alternative is DPATH.COM, which makes a starring appearance in this issue on the new DEALS FOR READERS Utilities disk no. AP018. This will do precisely what you want — it's exactly like PATH, except that it works for .OVL files and the like. You'll need a PATH command in your AUTOXEC or CONFIG file to tell the Apricot where to look for SW.COM, and you can follow that with a DPATH command to the same spot to get at the overlays.

SuperCalc lock-out

There seems to be an occasional locking-up of SuperCalc when a worksheet starts loading. After a few seconds SuperCalc is obviously unable to complete the load, leaving the Apricot's screen empty apart from the 'loading' message.

The obvious consideration is that one or other of the disks has somehow become corrupted. To check this it's worth running the Sorcim MAINTAIN utility on each part of the SuperCalc suite and CHKDSK on both the program and data disks.

Another possibility is that some odd characters have somehow become attached to the start or the end of the .CAL file. Judicious use of EDLIN might work for removing them. ◆

The Q&A Clinic

This section is free to APRICOT FILE **subscribers; we have a number of experts and enthusiasts on tap to tackle any and all problems — and we print those that we think would be of general relevance.**

Send your queries to:

APRICOT FILE
P O BOX 509
LONDON N1 1YL

Review

The best £50 I ever spent?

Harrison Ward's Utilities Disk for the Fortran programmer

By Edward N Bromhead

Harrison Ward Associates markets a number of utility disks for the Microsoft Fortran programmer. Some of these betray their origins in that they're Sirius-specific (providing interfaces to FABS, GRAFIX and AUTOSORT — whatever they are); but they also include an Apricot disk containing a set of useful Assembler-coded facilities that allow you to access system functions. The Utilities disk (available through DEALS FOR READERS at £50) has recently been upgraded and considerably enlarged, and now includes a simple-to-use GSX interface.

What you get

The Utilities for the Apricot come as a single-sided microfloppy, but they fill only a third of it:

Documentation files .	54.4KB
Utilities in Microsoft Library form	18.4KB
Demonstration Fortran source code	32KB

At first sight this might seem a little mean, but the facilities offered far outweigh the empty space on the diskette. Assembler-coded routines are very compact, and the three libraries contain some 65 separate subroutines — two systems libraries with 15 and 27 routines respectively, and a GSX library which has 23 routines. Each routine is callable from an MS-FORTRAN program (or indeed from MS-PASCAL) using conventional syntax.

Both libraries are easy to use, even for those functions which are catered for with the latest version of MS-FORTRAN with its (non-standard and hence non-portable) code interfaces to Microsoft's C and Pascal compilers.

The original library

This does many of the things you might want in your Microsoft Fortran program but which aren't supported as standard library functions. You can set the system date and time, or read the current date or time into your program. You can interrogate the operating system to see which version of MS-DOS it runs, see if a named file is on a disk, get or set the default directory, find the amount of free space on a disk ... and so on.

All very useful stuff. There's even a Fortran version of the handy Basic INKEY$ function.

The expert Fortran programmer will unfortunately be appalled by the way some of these utilities have been set up. They have been implemented as both Functions and Subroutines, and they do some dreadful things. Functions change their parameters, or do not return a value. Cripes! A FUNCTION subprogram is normally expected to return a single result. Good programming practice forbids a FUNCTION to alter its parameters, affect COMMON, or to do any I/O except display error messages. A SUBROUTINE is the way to do all the things a FUNCTION shouldn't, but cannot pass back a single value the way a FUNCTION can.

Apart from that, the utilities do work — even though serious Fortran programmers would probably disguise them slightly in their own subroutine calls. (And I'm told, HWa are going to recast many of them in more conventional form in the next release).

The library makes a nod in the direction of screen-handling with one routine for passing strings to the screen, bypassing the normal Fortran WRITE statement. It remains to be seen how useful that is. In principle, though, you could develop a Fortran program which apparently did no input/output. That straightaway removes a huge block of standard routines, and means that programs shorten by 20K bytes or more. (It's embarassing to write a five-liner that compiles to 37K bytes!) Programs also speed up.

The revised library

This is more esoteric, and allows you to access some of the fancier facilities in MS-DOS 2.11 and 3.1. Just about all of the MS-DOS system calls are implemented, this time through more conventional subroutine calls.

The revised library is a very competent and well-structured piece of work, though for the average Fortran application it's probably slightly less useful than the original library. To use it fully, the programmer would need to know his way round the *MS-DOS Programmer's Guide*. But if you want to create or delete subdirectories, use file 'handles', tinker with file attributes and so on, these are the routines for you.

The GSX library

Ah, the wonders of GSX! Originally intended to provide a standard environment for graphics software on the IBM PC and taken up by Apricot for its pre-Xen range, GSX now appears to have been abandoned by all but a few stalwarts.

The worst thing about GSX is all the bumf you have to wade through to get to the essentials of the system. In essence, all you have to do is call a single subroutine with five parameters to do any one of the many graphics options: but each parameter is an array, and you can end up having to set dozens of the array elements to do even a simple job like drawing a line! Avid APRICOT FILE readers will recall seeing entire articles on this subject from time to time ...

Prospero Software were the first out with a set of GSX-interfacing subroutines callable from its own Pascal and Fortran compilers. But Propero's PROSPECT package is (fairly) expensive, and obviously interfaces only with Prospero languages.

As an alternative, and if you are *really* rich, you can buy GINO-F and forget GSX (it might be cheaper to buy a small Rolls-Royce).

Oddly enough, Digital Research does not include with DR Fortran 77 the interface to its own GSX system, though the GSX toolkit (for PC-DOS only, but haven't you heard of the Alpha Disk?) does have some assembler interface modules in it. Those of course are in Digital Research's RASM86 assembler, and the non-assemblerate like myself cannot even begin to do the translation into Microsoft's assembler. That's a pity, because MS assembler often falls conveniently off the back of a PC-DOS floppy ... (I can just about copy an assembler routine from a listing and assemble it; but as for what it does, who knows?)

Incidentally, when you buy your GSX manual from Apricot, a number of important pages are left out — but the rubbish on how to use the GSX installation program, which doesn't work on the Apricot, is left in! Buy Microsoft Fortran for the Apricot, and what do you get? A GSX interface. Confusing, innit?

The free-with-MS-FORTRAN interface allows you to write GSX-using programs, utilising all those horrible subroutine calls, without spending any extra money. The facilities in the HWa disk mean that you not only save most of the cash, but you don't have to put up with the bumf either — you can just call a separate subroutine for each major GSX function, and these have the minimum of parameters. Nice.

Finally, if you have to step outside the supplied routines and call GSX itself with those nasty arrays, the HWa library has two function calls for that too. One lets GSX know where the arrays actually are, the other allows you to invoke GSX without giving the arrays as parameters again.

The future

My only adverse comments are that there are some very useful functions left off, notably on the GSX side: and that, in common with virtually all software, the documentation is somewhat incomplete.

Chief among the omissions from the GSX routines are the ability to set fill, line and text colours, and to choose a hardware text font. You can still get these functions using the full GSX rigmarole, but it would be nice not to. (This is really looking a gift horse in the mouth.)

In the ACTIVITY GSX screen drivers there are a number of so-called 'Escape' operations. These are Apricot specific, they aren't part of the Digital Research software: they do fancy things like moving bit-maps about on the screen. They would appear to be difficult to use from Fortran as it is difficult to access segment/offset data. I would have thought HWa might have given one or two little routines to do these functions. When using MS-FORTRAN version 3.30 or 3.31 there are (strictly non-standard Fortran) facilities for getting and passing segmented addresses, but that plunges you back into the GSX morass once you have them.

Quick Tip

To get the flavour of 'black-on-white' graphics, start your program with this:

```
CALL GSXDAT
CALL GSXOPE(1)
CALL FILTYP(1)
CALL GSXBAR(0,0,32767,32767)
CALL GSXMOD(3)
```

This sequence opens GSX, sets solid fill mode, fills the whole screen as if it were a bar, and turns on XOR writing mode. That puts a black line if on green, or green if on a black background. The 'clear screen' function, by the way, is GSXCLR not GSXCLE as per documentation. The graphics then come alive! Forget it, though, if you have a Portable ...

Obviously, utilities like this develop and grow — as much by feedback from users as from development initiatives by their originators. The sort of features one could expect to appear in the future are a way for the programmer to find the amount of apparently free memory in the computer and to make this useable. The interested reader will find an Assembler routine to do that on pages 211 and 212 of the July 1986 *BYTE* magazine; sadly that routine is IBM-specific and would need some alteration to work on the Apricot. Expect something like this on the next HWa disk.

Similarly, it would be handy to have a routine for picking parameters off the command line. This makes the Fortran program easier to use from a BAT file if it has options within it. (I know you can put responses in a file and use I/O redirection, but this way would be easier.)

A few moments' thought will throw up a whole handful of other things that Harrison Ward might implement — PEEK and POKE, for instance: an interface to the mouse driver, or to the Apricot Control Device. It looks to me as though HWa could have at least as big a market upgrading existing users as winning new ones!

Documentation

Most of the documentation comprises text files on disk. These have the odd typographical error (don't we all), but are generally readable and understandable.

I corrected my documentation files before printing them out, and I reformatted them so that they matched in size the Microsoft Fortran manuals and could be kept in the same binders (other software houses and writers of README files, please note).

The documentation for GSX functions refers to really obsolete versions of the screen device drivers; the update document seems to be a little behind the times on this. For instance, it appears unaware of some of the details of the text fonts which have a much better range of sizes and types than you would think from this account.

Last words

You probably won't need much support (you've a cheek expecting much for £50). And don't bother ringing the telephone number given on the UTIL1.DOC file — it's a Bovis construction site these days.

But any one of the libraries is worth the £50 to a Microsoft Fortran programmer: the three together are a bargain. If you use the Microsoft package, you're a mug if you don't buy this disk. ◆

FORTRAN Utilities

Description:	Useful utilities for MS-FORTRAN programs
Supplier:	Harrison Ward Associates (via DEALS FOR READERS)
Address:	62 Lynton Road Rayners Lane Harrow Middlesex HA2 9NN
Telephone:	01-864 6551
Price:	£50
Configuration:	any Apricot
Conclusion:	Good value.

Files on the disk

The HWa disk contains five documentation files plus test Fortran programs showing how every single one of the utilities is used. They come in the form of these two libraries:

■■■ GSX Library

GSXCAL	call the GSX system
GSXOPE	open a device for graphics use
GSXCLO	close a device
GSXCLR	clear a graphics device
GSXLIN	draw a line between end points
GSXTXT	write a string of text
GSXUPD	update a device
POLYLN	join a series of points by lines
POLYMK	put a marker at a series of points
POLYFL	draw a filled area
GSXBAR	draw a filled bar
TXTHT	set height of text characters
TXTDIR	set the direction of the text
LINTYP	set the line type
LINWID	set line width
FILTYP	set type of interior style for a filled area
FILIND	set type of pattern or hatch to be used for a filled area
GSXLOC	move a location indicator and return the final coordinates
FINVAL	change a value (cursor appears on screen)
GSXCHO	return an entered choice
GSXGET	get a string from the keyboard
GSXMOD	set mode for screen display drawing

■■■ Original Library

DATER	returns the current date
DLFILE	delete named file from disk
IDDOS	determine the version of the MS-DOS operating system (for version 2.0 or greater)
IDFS	returns the amount of free space on the disk, and also the total space on the disk (for MS-DOS 2.11 or greater)
IDRN	return the default drive number
INK	read one character from the keyboard buffer
IRENAM	rename a file
ISCHAR	looks to see if there is a character in the keyboard buffer. If there is, it is read
ISETDR	change default drive number
ISETDT	set the date
ISETIM	set the time
ISFILE	see if a file is on disk
JOINST	concatenate two strings of known length
PRCHRS	send characters to the printer
TIMER	read the current time

■■■ Revised Library

CHCDIR	change current directory
CHDIRE	change directory entry
CLHAND	close a file using the 'handle'
CNTDAT	get/set country-dependent data
CONDAT	get/set control data relevant to character or block device
CRENEW	create new file for use with a 'handle'
CRHAND	create a file for access with a 'handle'
CRSDIR	create sub-directory
CRTEMP	create temporary ('scratch') file for use with a 'handle'
CURDIR	return the current directory path
DEVDAT	get/set device data
DEVRDY	check if device with specified handle is ready for I/O
DLDIRE	delete directory entry
DRTYPE	is named drive a local or remote (MS-NET) drive?
DSKTYP	hard or floppy disk?
DTDIRE	get/set date and time of directory entry
DUPHND	obtain a duplicate handle for a device
FDUPHD	force duplicate handle of your choice for a device
FILPTR	position file pointer
HDTYPE	is file with specified 'handle' on a local or remote drive?
LOCKFL	lock part of file
NUMTRY	number of retries if a disk operation fails because file is 'shared'
OPHAND	open file for use with a 'handle'
RDHAND	read from a file with specified 'handle'
RMSDIR	remove subdirectory
WRTHND	write to file with specified 'handle'
UNLKFL	unlock part of file

Classified

Free Small Ads

This Classified Ads section is free to APRICOT FILE subscribers. Just let us have a sheet of paper with the words. To summarise the requirements:

- **indicate in which classification you'd like it to appear**
- **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**
- **let us know when you want the ad removed from the list!**

Send your ad to:

APRICOT FILE
P O BOX 509
LONDON N1 1YL

Jobs

■■■ STAFF WRITER (EDITOR-DESIGNATE) REQUIRED FOR TP GROUP to work across a range of newsletters and eventually to edit this one. News- and feature-writing, planning and research, commissioning and subbing: knowledge of the computers is more important than journalistic experience, though lucid writing is a prerequisite. Anyone interested in a full-time job (modest salary plus profit-sharing and a lot of fun) should write to Dennis Jarrett at our address.

■■■ TRAINEE WRITER/DOGSBODY REQUIRED FOR TP GROUP — interest in MS-DOS computers and/or publishing (or a willingness to learn about either) is essential. Suit school- or college-leaver; lots of variety, on the job training, and good prospects. Write to Dennis Jarrett at our address.

Sales

■■■ APRICOT XEN FD needs a good home. Six months old, one careful owner; 512KB memory, dual floppies. Includes all standard software and a copy of A LA CARTE. Price £1,500 ono. Call 01-221 7612 office hours.

■■■ LAYOUT TABLE with plate-glass top, illuminated from below by four separately-switched strip lights. Has two drawers with locks. Measures 4ft 7in wide by 3ft 10in deep and 3ft high; suitable for artist, architect, photographer or similar. Must sell — £250 ono. Phone 01-968 5448.

■■■ APRICOT PC with 512KB memory, dual 640K disk drives, standard software (MS-DOS, SuperWriter, SuperCalc). Good condition, not much used. Price £650. Contact Dave Illing, 0270 661427, evenings.

■■■ F1 INFRA-RED MOUSE with light pipe and software. Low mileage, well-behaved. £50 ono. Contact 0734 732512.

■■■ PSION ORGANISER — original program and datapacks, RS232 link-up, documentation. Boxed. £40 ono. Contact Mike Nash on 01-671 8644 (home) or 01-584 5000 x 3545 (office).

■■■ APRICOT PORTABLE with Starter Pack, MS-DOS, ACTIVITY, 640KB memory: £650 ono. PALANTIR word processor and Speller: £80 ono. Also EPSON HX20, no software, suit salesman: £150 ono. Contact T Dawson on 0292 77588.

■■■ SPARE 9in monitor for Xi for sale — green screen, black casing. Price £99, or will consider an exchange for interesting software. Also OPEN ACCESS for sale: £99 or exchange (now using KMan/2). Contact Charles Stirton on 01-940 1171 (day) or 01-940 8428 (evenings).

■■■ APRICOT F2 with 12in green monitor. Brand new — last available example! Cased, unused condition. 512KB RAM, 12 months factory warranty, infra-red mouse, improved keyboard, GEM software, MS-DOS 2.11, ASYNC etc — with manuals and carrying case. Contact Mr Lund on 0603 630326 in working hours.

■■■ XEN HD with 20MB hard disk, 1MB RAM, paper-white monitor; plus SuperCalc and SuperWriter. Hardly used. Cost £3,500 plus VAT, will accept £2,500 plus VAT. Also available: WordStar 2000 and Epson LQ1500. Contact Mark Wardle on Gloucester 504294 (day) or 411402 (evenings).

■■■ 9in MONITOR for sale — green screen, black casing. Good condition. Price £50. Contact 06512 2667.

■■■ APRICOT Xi20S with 1MB RAM, 20MB hard disk, 12in monitor, Mouse, Internal Modem, Juki 6200 printer. Nine months old, hardly used. Price new was £7,000 — will accept £2,800. Also available: external 20MB hard disk and tons of software if required (not included in price quoted). Messages on 0772 719325.

■■■ SIRIUS TO APRICOT file transfer package. Official ACT version: cost £109, bargain at £45. Contact 061-428 5975.

■■■ APRICOT Xi, as new and boxed. 10MB hard disk, 720KB double-sided floppy, 9in monitor; SuperCalc, SuperWriter, SuperPlanner. Cost £3,444 — will accept £1,495. Dust covers and Securicor delivery included! Phone 061-428 5975.

■■■ XEN multi-user system with two 256KB Apricot PCs and Oki Microline 193 printer plus Delta 4 (networked), SuperWriter, SuperCalc. Cost £9,000 in March, will accept £5,750 now. A first-class system, but unfortunately the business did not materialise ... Details from 031-445 2218.

■■■ 9in APRICOT MONITOR green screen, black case, includes carrying case. As new, used only on Sundays, etc. Price £100. Contact Paul Lavin on 01-451 1951.

■■■ CARDBOX electronic card index system: no longer required, £75. Also once-used Apricot single-sided disks: boxed in 10s, £17.50 per box. Phone 041-332 2505.

Wants

■■■ PC or XI wanted — must be cheap! Contact T Dawson on 0292 77588.

■■■ PORTABLE PROBLEMS — I have Portable without monitor interface and version 1.6 boot ROMs. Can anyone in South London offer advice on

modifying/rectifying these? Could swap software for help ... Contact Mike Nash on 01-671 8644 (home) or 01-584 5000 x 3545 (office).

■■■ SERIAL MOUSE wanted. Contact Charles Stirton on 01-940 1171 (day) or 01-940 8428 (evenings).

■■■ CHEAP TURBO PASCAL WANTED by student, to help with studies. Write to Martin Norman at 14 Spurr Court, Foxwood, York.

■■■ LOAN OF PORTABLE DISKS WANTED — Dixons muddled up the disks I should have received, say there's two months delay on replacement, and have certainly omitted at least the Voice disk. Can anyone help? Contact Paul Anderson, 01-868 4499 ext 263 (office hours) or 02518 3470 (evenings).

■■■ DR LOGO — can anyone alter DR LOGO for the F1/F1e to run on the F2? I have tried Apricot dealers without success and the long-promised £30 DR pack still hasn't materialised. Contact David Stirling at 28 Ingleston Avenue, Dunipace, Denny, Scotland.

■■■ MOUSE WANTED for Apricot F1. Serial mouse preferred, but willing to consider infra-red Apricot mouse if cheap. Contact Dr Cartwright, tel 045 383 4576.

■■■ DATABASE/FILING PACKAGE (Palantir or Cardbox preferred) and A.S.D. Superwriter disk. Exchange IANKEY touch-typing tutor, GRAMMATIK grammar checker, PROOFREADER spelling checker (full working copies with documentation) — cash difference either way. Phone 095 786 261.

Help

■■■ SIRIUS-APRICOT TRANSFER: any program and/or data transferred from Apricot to Sirius format and vice versa. Just send your disks along with £5 for the first one and £1 for each disk after that: P&R Miller, 8 Hillside Close, Hednesford, Staffs WS12 4DG.

■■■ SANYO AND APRICOT: can anyone suggest how I might format 3.5in disks so that both my Sanyo MBC-555 and my Apricot can read me? Also, does anyone know where I can obtain Concurrent CP/M-86? Contact Bob Wilcock on 01-888 8304 (evenings).

■■■ HELP FOR F1/F2/FP USERS: Gateway Users Group exists solely for users of these machines. Details from Tony Hill at 85 Edna Road, Raynes Park, London SW20 8BS.

■■■ SWITZERLAND: Apricot user would like to contact others, preferably in the Zurich area, to exchange experiences and help each other with problems. Contact John Solomir, Stockenstrasse 67, 8802 Kilchberg — phone 01/715 10 94.

■■■ HELP GROUP: Apricot users within striking distance of Reading, Berkshire, and interested in forming a friendly group to give and receive help on programming, communications, software packages and anything else legal are invited to contact David Holland on Reading (0734) 68860 — evenings only. ◆

Letters

Readers' Writes

Subscribers have their say ...

NEC V30 revisited

Having just acquired a V30 chip for my Apricot PC, I eagerly turned your pages to Paul Lavin's article in APRICOT FILE Release 1.9. Reading that it wasn't going to work in my machine with BIOS 2.7 or 3.1 was of course a letdown. But not being able to take no for an answer, I went ahead with the replacement anyway and crouched behind my desk as I switched on the power.

A very unhealthy sound emanated persistently from the speaker, and the boot PROM diagnosed an Error 31 (keyboard faulty or disconnected). Now, my keyboard is fine; so clearly something was amiss, but there was no sign of smoke, nor of fuming Apricot officials. Furthermore, the little flashing arrow prompted me to insert a disk as usual, so I did, and lo! Booting commenced, the ugly tone disappeared, and everything proceeded as normal.

Well, almost as normal. Wasn't the display a bit perkier, the DIR output a little snappier? I summoned Peter Norton's System Information utility, which informed me that my machine's performance relative to the IBM PC was now 2.1 — a nice improvement indeed from the rather embarassing 1.1 achieved with the 8086 chip!

So far I have found only three (minor) inconveniences with the chip. First, the sound chip isn't timed quite right — it makes a noise on power-up, and the bell sound is sometimes a little off-key. Second, there's the erroneous Error 31 when booting. Third, the system clock runs a few seconds fast every hour. Under BIOS 2.7, the system clock isn't initialized properly from the keyboard clock, either; under 3.1 it is.

Apart from that, all the applications programs I have tried so far work perfectly. They include Turbo Pascal programs using the 8087, GSX, FORMAT, WordStar and even Microsoft Windows. My machine is a PC with two 720K drives, 768K RAM, Apricot mouse, and one Alpha Disc XF400A external drive. I may have to put the 8086 back in when I get a hard disk, but till then, I'm sticking with the V30.

Sten Ternstrom, Solna, Sweden

PS Fujitsu recently released a chip called MBL8089, a work-alike for the 8089 I/O processor. Unlike Intel's original, this one comes in both 5MHz and 8MHz versions (the latter is the MBL8089-2). Will anybody please find out if this means that we can have 8Mhz Apricots? ◆

20MB F1

So you thought the F series was dead? Well, Apricot's Portuguese agent **Landry** doesn't: it's packaging an F1 with a **20MB** hard disk, **512KB** of RAM on the motherboard, and the hard-to-get **Factory Format** software (plus all the standard F1 appurtenances) for a price that it says is **around £700** plus VAT. Landry tells us it is looking for UK dealers to sell the upgraded system: contact Sr Carlos Nogueira at **Landry Engenheiros Consultores, Rua Tomas da Annunciaçao 53-A, 1300 Lisbon** (telex 43436 COMPUT P).

Users unite

Computer-based products for professional graphics artists have tended to come from the States and typically require specialised hardware plus a hefty cheque. **Io Research** bucks both trends: it's British, and its prices are comparitively modest. Starting with add-on black boxes to provide high-resolution and multi-colour displays for desktop computers, Io now has a raft of graphics products — hardware, software, and systems — and some 7,000 users. And there's a new **Pluto User Group**, independent of but collaborating with the manufacturer: prime mover is **Alan Sutcliffe** at **Systems House Ltd** on **0734 781283**.

And users of Mountford & Laxon's highly rated **MLD** CAD and draughting software have also formed an independent User Association "to provide a forum for discussion and a means of distributing information of interest" — primarily through a regular newsletter. Info from chairperson **Gerry Bernado** at Technique & Design Consultancy on **0295 54758**.

CD-ROM standards

CD-ROM is one of the hot topics in hardware right now, providing computer-readable non-rewriteable versions of compact discs for publishing information. The so-called **'High Sierra Group'** consists of around a hundred firms from the computing, publishing and compact disk manufacture with an interest in the subject; over the past two years they've been putting together a set of standards for volume and file structures with a view to achieving some compatibility between CD-ROM disks and drives. That work bore fruit in the summer, and if you're interested in fitting the specs the easiest access is probably via the newly-published *CD-ROM Standards: The Book*. Produced by a High Sierra subcommittee, it also provides a very good overview of the current position, prospects and potential problems for 'optical publishing'. It's available for **£50** from **Learned Information** on **0865 730275**.

Money for technology

A survey among a hundred or so City sources of venture capital suggests that there's still a lot of money available for high-tech investment — but "the vast majority of proposals for finance were not good enough to justify an investment".

Touche Ross, who conducted the survey, found that 95 per cent of respondents had "ample" finance for expanding high-technology businesses, and three-quarters said funds were available for start-ups. But the venture capitalists went to town on the reasons for *not* investing — inadequate management, little or no market research, little or no scope for product development, underestimates of the amounts required. "The main conclusion was that well-researched and comprehensive business plans and marketing policies are, sadly, few and far between ... the overwhelming opinion that inadequate management is the principal problem area should make entrepreneurs aware of the range of skills required ..."

40MB cartridge disks

Removable hard disks are often attractive for fast back-up; **Micro Technology** reckons it has sold a thousand of its **Honeycomb** units (two 10MB cartridges). There's now a 40MB version, with two 20MB cartridges; the price is **£3,750**, there's an Apricot interface, and the company is on **0892 45433**.

In-car award

The M1 modem from **Transam Microsystems** — "the world's first in-car cellular modem" — won the hardware category in the London *Standard* Micro Business Awards for 1986. Runner-up was the Dest Corporation PC Scan page reader. The software winner was the elegant **Automator Mi** from **Direct Technology**, a programming tool which can genuinely improve productivity; Intelligent Environments' 'decision modeller' Deja Vu was the runner-up.

A better buffer?

Printer buffer boxes have been around for a while: they sit between the computer and the printer, acting as a temporary memory store so that the computer can send data towards the printer at a decent speed without having to wait for the printer's electromechanical tardiness to decide how fast the printing actually happens. But printer buffers are getting cleverer — and cheaper.

Take the **X-Buffer** from distributor **Micro Control Systems**. It has four sockets, input and output for RS232 serial and Centronics parallel: data can be switched between them. So one computer can use two printers, plotters, or modems: two computers can share one such device: and serial input can be converted to parallel, or vice versa.

The X-Buffer isn't unique, of course, though it can be software-controlled for a variety of functions (multiple copies, CR with LF, hex dumps, single-sheet feed, and more). But it sounds clever, looks solid, has a two-year warranty, and is modestly priced — the basic unit with 64K memory costs **£159**. More from **0602 391204**.

Microfloppies for IBM

IBM's announcement of internal and external 3½in 720KB microfloppy disk drives for all its PC models must have produced a few quiet smiles at Olivetti — we understand that all Olivetti PCs have had the necessary drivers for 3½in disks built into their disk controllers from the start. On the other hand, of course, there were broad grins at Apricot: according to MD Roger Foster, "It's the best news for Apricot there has been for some time ... We're delighted that IBM has opted for what is clearly a better standard". ◆