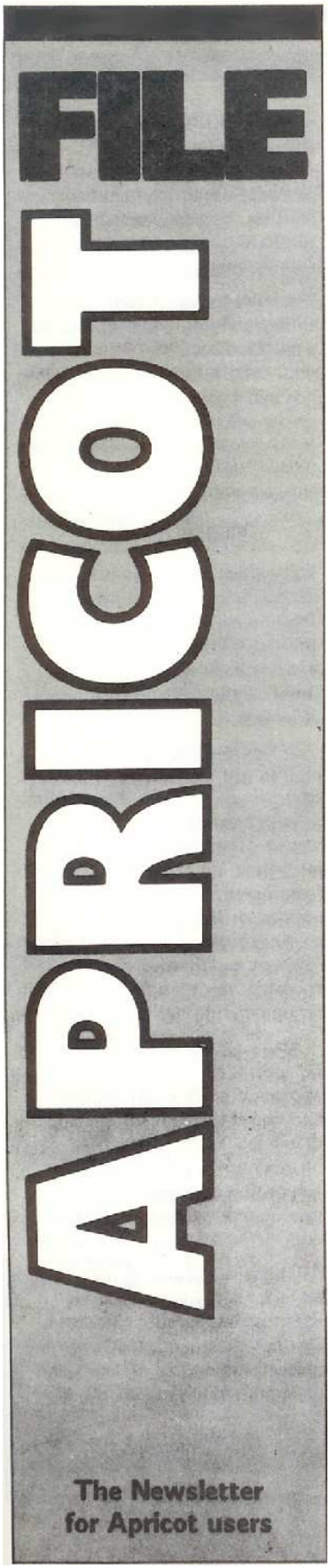
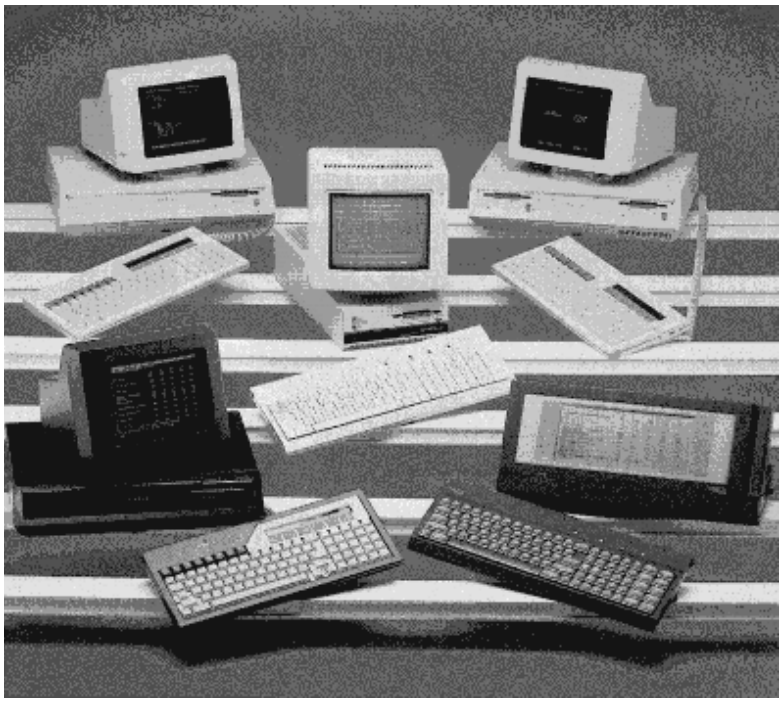


Apricot File 1_5



Scans courtesy of www.actapricot.org

Copyrighted content republished with thanks and permission from:

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

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

Release 1.5

FILE APRICOT

The Newsletter for Apricot users

February 1986

At the Apricot Show

By Henry Deckhand

The Apricot Show was a pleasure for us — good to meet so many contributors and readers, to put faces to names. We also distributed over a thousand subscription forms, so we're expecting to welcome many new subscribers ...

The Show itself wasn't packed with fascinating new products — the computer industry as a whole seems to have moved on from the first flush of technological excitements of the early 1980s. But there was much to see. Here's what struck us particularly.

Citi-Software does a rather good expansion box for Apricots, but was also talking about an interesting four-channel serial communications card with on-board Z80 controller and baud rates to 19,200. At a projected end-user price somewhere between £350 and £500, this could be a cheap option for applications like lab instrument control and point-of-sale ... We also chatted about a £500 MicroCache-plus-512KB add-on that Citi-Software is considering for big Lotus worksheets and the like, a kind of Apricot RAMpage ... Contact 01-831 8369

Alpha Disc has beaten Apricot itself in providing a degree of IBM compatibility on the Xen — Apricot is telling people to wait at least until April for its add-on IBM-format disk drive, but Alpha Disc can do you one right now for £495. You get a box that reads, writes *and* formats 40-track 360KB diskettes ... Contact 0784 35357

Air Call was demonstrating (a bit erratically) the **Transam Microsystems** M1, a £250 in-car modem that works with cellular radio systems. No executive can be without one ... Transam is on 01-4005 5240

Plus 5 had an add-on hard disk for the Xen — albeit in its Olivetti-compatible livery. No prices yet, though ... Contact 08926 63211.

Logica had a small stand with Xenix demos on the Xen, and told us (with no prompting from Apricot people) that the implementation was easy and effective. Right now it's Xenix 3.1, of course, but the New Improved Xenix System V should be ready in a matter of months. Meanwhile Xenix is reportedly just about shippable, at least as far as Logica is concerned. The basic operating system is being quoted at £695, the development software is the same again; Apricot is quoting £10,176 for an example six-user system with 40MB and F1s as terminals. ◆

In this issue

The Disinformation Column 3
GSX Workshop: Part Four 5
Psion Chess reviewed 7
Basic Workshop: CHECKER 9
Apricot news from the front line 12
Notebook: DELTA passwords 13
Software round-up 14
UNDELETE — the True Story 15
Technote: line-drawing 18
Deals for Readers 19

About the newsletter

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

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

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

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

Copyright, permissions and reprints

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

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

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

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

Contact points

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

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

Telephone **01-609 3800**

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

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

Subscribe!

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

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

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

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

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

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

Write for us

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

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

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

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

ISSN 0268-6066

The Disinformation Column

On tiny COM files

by **David Langford**

Have you noticed that there's a yawning social gap between two classes of software users?

If you sit all day bashing copy into a user-hostile word processor or laboriously simulating Sir Clive Sinclair on the spreadsheet, that's *application* software: your hands are deemed not to have been soiled (we won't discuss the state of your brain).

But should you stoop to write a one-line batch file or play around with MSBASIC, you've slithered into the murky world of software *development* — and on the whole it seems that a higher social status is enjoyed by collectors of night-soil.

I was properly put in my place during visits to AT Computerworlds, WH Smith Business Centres and the like. "You want an ... *assembler*?" they said in alarm. Pause while exquisitely clad salesperson retreats to the back room and the *Penguin Dictionary of Computing*. "Oh, that's Development Software. We don't stock that kind of thing because" (all together now) "there's no call for it."

Thus are you spared a column about the MicroSoft Macro Assembler, you lucky people, and I'm left to wonder whether it's only bootleg copies which cause the Apricot PC to hang up as a cunning means of indicating an assembly error ...

Compact COM files

In fact, though, you don't need an assembler to fudge up tiny .COM files for tasks like screen manipulation: they aren't as mysterious as they seem. Remember how .BAT files once appeared enigmatic? Remember the sudden rush of illumination on finding they contained instructions almost in plain text, and that they could be created and edited with your word processor?

Well, COM files contain plain machine code with no frills; and to create a very simple COM file you require only three assets:

- *patience*, which as a reader of Apricot plc manuals you should by now have
- a little *information*, with which I am about to bore you rigid
- a copy of the fabulous DEBUG, as supplied to all my own customers (advertisement) and available on DEALS FOR READERS Disk No. AP002 at £15 (plug).

For my money, the shortest COM file to do anything even remotely useful is BEEP.COM. It simply provokes the Apricot to toot when it's run. You can then include the line 'BEEP' at the end of long batch files to produce an audible "I've finished!' signal. BEEP.COM runs to just eight bytes, and it requires knowledge of no more than two assembler commands.

Here's how to create it from scratch.

First, load up DEBUG. At the prompt (usually a hyphen), type 'A 100' and Return; this tells DEBUG to expect assembler instructions and to start the resulting machine code at memory location 100.

Alarming numbers will now replace the prompt: four digits of current segment address (which you can ignore), followed by a colon, followed by four digits of address within the current segment.

So, in the following, it's only the stuff on the right which you actually need to type (end each line with Return):

```
209A:0100   MOV AH,2
209A:0102   MOV DL,7
209A:0104   INT 21
209A:0106   INT 20
209A:0108
-
```

Entering a blank line at the end (by pressing Return on its own) takes you back to the hyphen prompt. Now, what do these lines actually do?

- MOV AH,2 'moves' the value of 2 into the AH register: '2' is the MS-DOS function code meaning 'display the character in register DL'.
- The next line pops a '7' (that is, ASCII character 7 — or Control-G, which produces a beep) into register DL.
- All the work is then done by the INT 21, an 'interrupt' which asks MS-DOS to do what the AH function code is telling it.
- INT 20 stops the program (there are lots of ways to end COM programs, but this is the shortest)

At this point you have the entire dwarf program in memory in the eight locations from 100 to 107 inclusive. You can even run it from DEBUG, by typing 'G': that should produce a beep and the message 'Program terminated normally'.

Let's write it to a file. First type 'R CX'. This shows you the current value of the CX register and gives you a chance to change it: a colon prompts you to enter a new value. Because we have an eight-byte program and CX controls the number of bytes DEBUG will write to disk, the value to enter here is '8'. (This is always easy to work out: it's 100 less than the last memory location figure displayed when entering the code. Remember that '209A:0108' at the end of our DEBUG input?)

Now type 'N BEEP.COM' to give your file a name; and then write it to disk with the command 'W'. Leave DEBUG by typing 'Q'.

More fun

When you've wearied of the ecstasy of your very first machine-code program and can stop filling the air with hideous beeps, you might like to reflect on this: any hexadecimal number, from 0 to FF, can be substituted for the '7' in that 'MOV DL, 7' command.

This means you can write tiny .COM files to generate *any* character, including those with ASCII codes 127 to 254 (hex 7F to FE) — which with a few exceptions like the Sterling and Yen signs can't be typed at the keyboard. (Incidentally, new users might like to know that several 'specials' can be typed using Control with the keys in the top row: for example, Control-4 produces a Yen sign.)

Character codes 0 to 31 (hex 00 to 1F) produce 'control characters' like Return (hex 0D) or Linefeed (hex 0A). But the font also contains displayable characters with these codes: silly faces, playing-card graphics, male and female symbols, musical notes ...

As noted in APRICOT FILE Release 1.3, you can display these by sending Escape and 8 to the screen for the relevant character code. This suggests a tiny utility for generating them.

Use DEBUG to enter the assembler instructions as before. Here's what to type:

```
MOV AH,2          'print character' code
MOV DL,1B         Escape (1B hex, 27 decimal)
INT 21            MS-DOS call to print
MOV DL,38         character '8' ASCII code
INT 21            call MS-DOS again to print
                  it
MOV DL,[0082]     read command line
AND DL,1F         get into range 0-31
INT 21            again, print it
INT 20            end
```

Call this one LIT.COM and save it as before: it runs to 15 (hex) bytes.

Your first line is the 'print character' code, line 2 moves the Escape code (1B hex, 27 decimal) into DL, and 'INT 21' is the MS-DOS call to print. You do it again for hex 38 (the character '8' in ASCII).

Then 'MOV DL,[0082]' reads the command line; 'AND DL,1F' gets into the range 0-31 and 'INT 21' prints that; and 'INT 20' signifies the end.

The first four and last two lines should make sense. The enigmatic 'MOV DL,[0082]' gets the first character from the 'command line'!v which in practice means that if you run this program by typing 'LIT A' from the MS-DOS prompt, the 'A' (character code 41 hex) is what will go into register DL.

The command following that does what assembler programmers call 'masking': that 'AND DL,1F' chops out any bits in DL which aren't in the number 1F (hex), or 31 (decimal), or 00011111 (binary). This conveniently means that capital 'A' (binary 01000001) and small 'a' (binary 01100001) are both stripped down to '1' (in any number base you like).

Character 1 is also known as Control-A. So 'LIT A' gives you ASCII character 1, 'LIT B' gives you character 2 ... and by the time you've worked through to 'LIT _' for character 31 you'll doubtless have had quite enough.

Do you really have to waste your days doing 'INT 21' commands to send stuff to the screen one character at a time? No, you don't; and thankfully you don't have to. Our last example is a teensy file which can be adapted to send almost anything you like to the screen — anything, that is, except a dollar sign.

```
MOV AH,9          'print string' code
MOV DX,0109       set start address for string
INT 21            kick MS-DOS into life
INT 20            kick program to death
DB "Put any character string here" 0D 0A
DB "Mix quotes with ASCII codes: " 2A 7F " "
EC
DB "Finish with a dollar sign.$"
```

Since the four instructions which make up the program are always the same, the string will always start at location 0109. When you finish entering the 'DB' (which stands for Define Bytes) lines, make sure to note the last address and set CX accordingly before saving the file.

This way of writing chunks of stuff to the screen can be a lot quicker than multi-line BAT files — as well as looking more impressive.

Last word

Most irritating conversation of the month: a dealer who proudly announced that although APRICOT USER arrives in the box with every machine, he always takes the magazine out and thus prevents buyers learning how extortionate his prices are. I wonder if he visits Dixons and obliterates the price-tags with a spray-can? For a small fee I will be prepared to disclose his name and title ... ◆

Workshop

The Fundamentals of GSX

Part Four: A sketching program

by Lindsay Doyle

This series provides a pragmatic introduction to the Apricot GSX graphics extension — how to use without necessarily needing to know how it works.

This time we have a complete sketching program that demonstrates the use of GSX drawing functions. You'll need the three previous articles in this series; and I assume you have transcribed the GSX skeleton program GSXSUBS.BAS to disk and have the other files previously described also lying in wait for use.

The listing given here has its line numbers selected so that it can be interpolated into GSXSUBS.BAS. The resulting program, SKETCH.BAS, gives you a white screen on to which you can draw black lines of seven types in eight directions at two speeds: what more can you ask for? (Don't answer that.)

The program has been heavily annotated with remarks, so it shouldn't be too difficult to follow in conjunction with the explanation of the GSX opcodes in the previous instalment.

Having called up the cursor at line 470, you can move it around the screen with the number pad keys — hitting any other key will cause GSX to drop out of the cursor routine, with cursor position and key code reported in X2, Y2, and N.

Lines 530-660 determine the meaning of the key pressed and carry out the command by calling the applicable routine. Lines 680-690 draw a line between the previous and current cursor co-ordinates, reset the co-ordinates, and loop back to repeat.

The switch to BIOS version 2.7

In the middle of 1985 ACT started shipping Apricots with a new release of the BIOS called version 2.7. Among other things, the BIOS decides how goodies buried inside the Apricot are actually called up by programmers; and this includes font selection.

These articles were written in the fond assumption that ACT wouldn't be changing anything very significant in the BIOS. But somewhat inconveniently the alterations for 2.7 include a completed reversal of the font selection numbers; up to version 2.6, the dot font was number 2 and the character font number 1 — but 2.7 has 1 as the dot font and 2 as the character font.

The listing for GSXSUBS.BAS includes REMs in 300-420 that summarise this, and there's also a REM at line 50130 which indicates how the set-up parameters should be changed if you have BIOS version 2.7 (the start-up display should tell you which BIOS you have). Briefly, you should alter 50120 to read ...

```
50120  FOR Z = 2 TO 10: G2%(Z) = 1: NEXT Z:
       G2%(6) = 1: GOSUB 54000: RETURN
```

(Actually you could just delete the 'G2%(6)' parameter altogether since that will be the default.)

Thereafter — and in subsequent articles — you BIOS 2.7 users should change FO=1 to FO=2 and vice versa.

Next issue: a real sockdolager of a graphics demonstration program for those of you too lazy or too busy to have designed your own by now ...

Copyright 1985 Lindsay Doyle

SKETCH.BAS — a complete GSX screen sketchbox

```
10      rem: SKETCH.BAS BY L.DOYLE
20      rem: must be interpolated into GSXSUBS.BAS
30      :
300     rem: Instructions Option
310     PRINT CHR$(27); "SKETCH INSTRUCTIONS? (Y
        OR N)
320     I$ = INKEY$: IF I$ = "" THEN GOTO 320
330     IF I$ = "Y" OR I$ = "y" THEN GOSUB 60000
340     :
400     rem: Initialise
410     DW = 32767: DH = DW: DV = 1: N = 2
420     GOSUB 50010: GOSUB 50100: rem DIM arrays,
        open workstation
425     :
430     rem: Re-initialise
440     GOSUB 50300: rem Clear
450     WM = 1: GOSUB 53200: rem set Fill to
        white, draw bar
460     X1 = 0: Y1 = 0: X2 = DW: Y2 = DH: GOSUB
        51110
470     WM = 4: GOSUB 53200: rem set black line
480     X1 = DW/2: Y1 = DH/2: rem start at centre
        of screen
490     GOSUB 52800: rem 'Input Locator' gives
        cursor
495     :
500     rem: Line Drawing Routine
510     rem: Press any key except numbers to drop
        out of Locator. Then:
520     X2 = G5%(1): Y2 = G5%(2): IN = G4%(1): rem
        IN = the key that was pressed
530     IF IN = 81 THEN GOSUB 50200: END: rem "Q"
        - Quit
540     IF IN = 113 THEN GOSUB 50200: END
550     IF IN = 66 THEN WM = 4: GOSUB 53200: rem
        "B" = black line
560     IF IN = 98 THEN WM = 4: GOSUB 53200
570     IF IN = 87 THEN WM = 1: GOSUB 53200: rem
        "W" = white line
580     IF IN = 119 THEN WM = 1: GOSUB 53200
```

```
590     IF IN = 67 THEN GOTO 440: rem "C" = clear
        Graphics screen
600     IF IN = 99 THEN GOTO 440
610     IF IN = 76 THEN GOSUB 700: rem "L" =
        select line type
620     IF IN = 108 THEN GOSUB 700
630     IF IN = 72 THEN GOTO 480: rem "H" = Home
        to centre of screen
640     IF IN = 104 THEN GOTO 480
650     IF IN = 77 THEN GOTO 690: rem "M" = Move
        without writing
660     IF IN = 109 THEN GOTO 690
670     :
680     GOSUB 50600: rem draw line
690     X1 = X2: Y1 = Y2: GOTO 490: rem loop
695     :
700     rem: SELECT LINE TYPE
730     LT$ = INKEY$: IF LT$ = "" THEN GOTO 730
740     LT  = VAL(LT$): IF LT < 1 OR LT > 7 THEN
            GOTO 730
750     GOSUB 51500: rem set line type
790     RETURN
795     :
800     :

60000   rem: Instructions
60010   PRINT CHR$(27); "SKETCH draws black lines
        on 800x400-dot white field";
60020   PRINT " in eight directions.
60030   PRINT "Press the initial letter to get any
        of the following functions:"
60040   PRINT: PRINT "B = BLACK LINE"; SPC(32) "W
        = WHITE LINE (erase)"
60050   PRINT: PRINT "M = MOVE WITHOUT DRAWING
        (use M at beginning and end of";
60060   PRINT " move)": PRINT
60070   PRINT "<SPACE> KEY DRAWS LINE FROM PRESENT
        TO PREVIOUS CURSOR POSITION.
60080   PRINT: PRINT "(to draw an isolated dot
        press M <SPACE> M.)"
60090   PRINT: PRINT "C = CLEAR SCREEN"; SPC(31)
        "H = HOME TO CENTRE
60100   PRINT: PRINT "5 = TOGGLE FROM HIGH TO LOW
        SPEED"
60110   PRINT: PRINT "OTHER NUMBER KEYS 1 TO 9
        MOVE CURSOR IN INDICATED DIRECTION
60120   PRINT: PRINT "L = LINE FOLLOWED BY 1 TO 7
        TO CHANGE LINE TYPE";
60130   PRINT "  (Always use L1 for erasing.)"
60140   PRINT: PRINT "Q = QUIT THE PROGRAM";
        SPC(27) "<RETURN> key not required
60150   PRINT: PRINT
60160   PRINT SPC(20) "PRESS <SPACE> KEY WHEN
        READY TO PROCEED
60170   I$ = INKEY$: IF I$ = "" THEN GOTO 60170
60180   RETURN
```

The easy way

A text file for the complete GSX series along with all the programs and demos are available on an Apricot disk from our DEALS FOR READERS mail order service — order no. AP004, price £15 including VAT and P&P.

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, PRACTICAL COMPUTING and WHICH COMPUTER? Books include THE IBM PC BOOK (not very thick), THE SIRIUS/VICTOR BOOK (very much thicker), THE ELECTRONIC OFFICE — A MANAGEMENT GUIDE (fairly thick), three titles about Commodore home computers (one thick, one a bit thick, one pseudonymous), and THE GOOD COMPUTING GUIDE (*really* thick and very cheap too at £2.95 — or £3.00 inc P&P from DEALS FOR READERS).

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

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

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

Terrance Duggan is one of the prime movers in TP Group — his responsibility is generally the office management side. He is basically an artist (great with pen and ink — give him a call if you want a signed original of your house or your office) but in the past has been an Army officer, a bookies' runner, and sundry other things.

Tony Weyman is an optometrist with a practice in North London; he uses a 10MB Apricot Xi to keep patient records. Originally he bought the machine from a software house which was to write the system with dBASE II, but after nine months he still didn't have a working program and so he decided to try it himself. "Needless to say the system now works — and I've had a lot of fun (and much misery) in doing it". ◆

Review

Mate in one

Psion Chess reviewed

by Terrance Duggan

Psion may be better known to Apricot users for its reportedly rather good XCHANGE suite of office programs — linkable file manager, word processor, graphics, spreadsheet. But the company also has the marketing rights to a chess game that was joint winner of the 1984 World Microcomputer Chess Championship.

And slightly to our surprise, we found that Psion Chess represents microcomputer chess logic at its best.

The program arrives in a neatly packaged box with decent instructions — makes a change from all those ring-bound manuals in stand-up folders. The box contains a single-sided Apricot disk and a nine-page guide; the booklet gives all the instructions you need to use the program. You also get a list of the 50 'master' games provided on the disk, along with brief reasons for their inclusion (all very sensible).

The disk is copy-protected. You can make a backup: but without the Master inserted in your disk drive, the copy simply will not run. *So don't lose the Master disk* ... Psion told me that if I had "a valid excuse" for losing the Master, its support people "would see what they could do". Hmm.

You can however play on your backup disk once you've got things running and after you have extracted the Master from the drive (so if you have more than one Apricot you can play on more than one machine at a time!).

Play away

The game itself is very realistic. The computer's own level of play is above average without being too far beyond it; the display looks realistic; and you get the clocks that some chess packages on micros omit.

And unlike some of its competitors, Psion Chess has a pretty comprehensive range of levels of play from novice to something near Grand Master. For instance, once the screen has been drawn, a beginner with no knowledge of chess could press the 'C' key continuously — whereupon the screen shows your best move in the circumstances. In effect that makes the computer play itself, which is a remarkably easy way of learning the basics of the game — and it's a good tutor for strategies, too.

Beyond that, Psion Chess has an outstanding range of features to help you become a better chess player. You can invert the board to view the game from your opponent's side of the board. You can change the level of play during the course of the game, giving yourself and the computer less or more time to think. You can ask the computer to play your move for you if you are in a difficult position, or you can freeze the game (stopping the clock) until you have made up your mind on the next move. You can ask the computer for hints, and you can even take back your previous moves to see where you made an error.

The 50 Master games on the disk which you can run are an education in themselves. You can play set pieces and ask the computer to try and checkmate in a certain number of moves — ideal for solving those irritating end-games you find in certain national newspapers. You can also force the computer to play its *second* best move rather than that lethal first choice!

Once you have progressed beyond the novice level you can request an analysis of the game. This appears at the foot of the screen and tells you precisely what desperate straits you're in ... There's the standard print option for your own games, but you can also ask for a copy of the master games; and of course you can save your current game to disk for subsequent analysis or to complete it later.

Caveats

After all that good news comes the bad: sadly, Psion Chess is not perfect. For a start, more technical information would have been useful; the program was obviously written by an aficionado, and for instance one of the Master Games could have been given an explanation of the logic behind each move — after all, the disk is not full.

The program provides the option of displaying the board in the usual two-dimensional plan, or as a three-dimensional view such as the player might actually see. Although most of its users think of Psion Chess as a 3-D game, in fact all the commands are entered on the 2-D screen. Unless you enjoy switching between displays during the game, you will certainly need the guide open beside you until you know all the commands.

And since the 3-D is drawn in one-point perspective, it's easy to obscure a pawn because it is in front of your King or Queen — thereby losing it from your sight and possibly from your game.

The 3-D screen is impressive as a graphic display, but it could so easily have been improved. An abbreviated list of commands and a small 2-D game plan in a window would have been a great help. And given the tension that can build up, I find it most unsatisfactory that beating the computer results in no more than a flashing King. Where are the heart-felt congratulations, the 3-D

handshake, the encouraging comments? you don't even get the King toppling over.

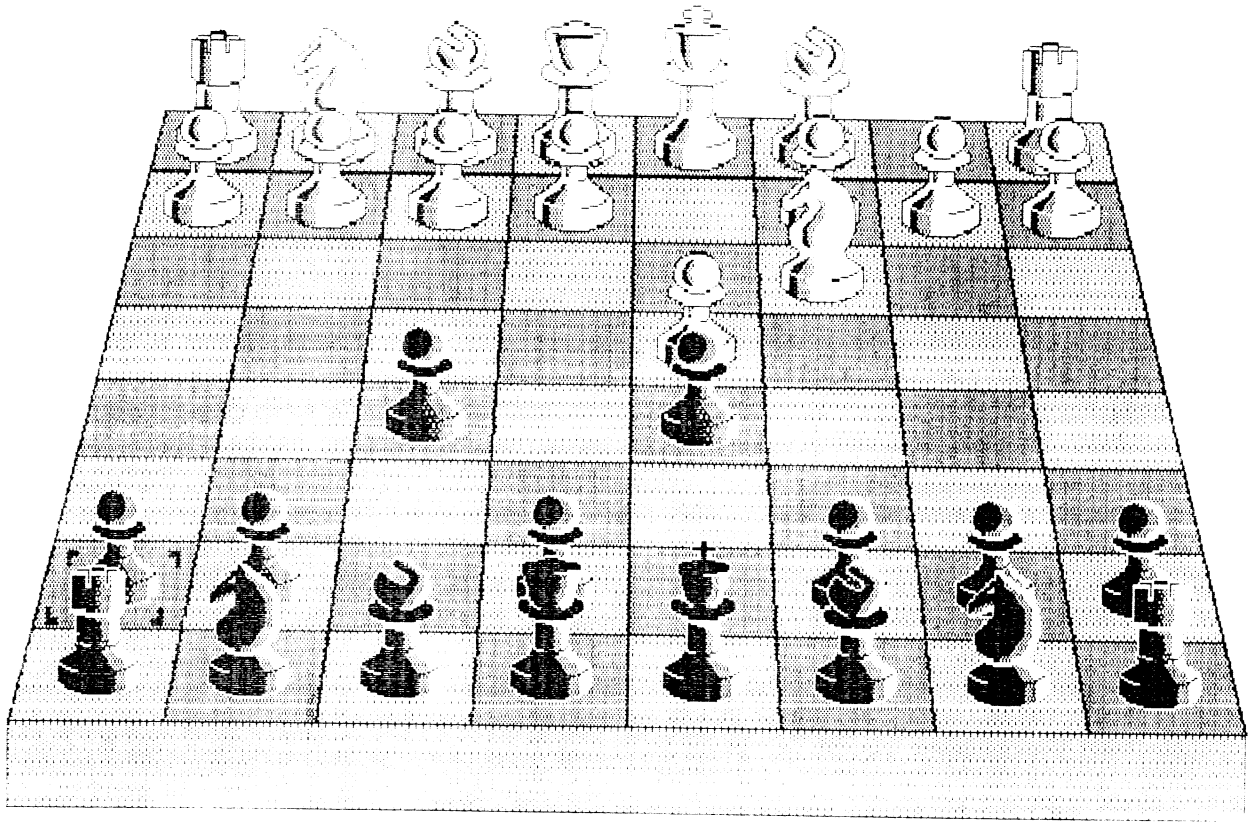
But the most irritating feature of Psion chess comes when you move a piece. That you do by resting the cursor on the piece and striking the Enter key to select it; using the arrow keys, you then move the cursor horizontally or vertically to the destination square; and you hit Enter again to indicate that this is where you're moving to, and the computer transfers the symbol to that location.

There are three problems with this: it's hardly a help in learning chess notation, for one. And the cursor is always positioned on the extreme lefthand pawn's square (H7 for white, A2 for black); so if you are playing the length of the board with a Bishop, you have to move by taking the cursor along two sides of a triangle to get to the desired square. Third, the computer doesn't have to use the cursor keys or Enter — so the program has a built-in advantage over the human when playing against the clock.

Still, even with its idiosyncrascies, Psion Chess is without equal on the Apricot. The quality of the chess played by the computer and the features at your disposal to help you understand the game and improve your skill both make it well worth the money — it could cost you more than this to buy the books to cover just the Master Games part of the program. ◆

Psion Chess

Description:	Chess game
supplier:	Psion Ltd
Address:	Psion House Harcourt Street London W1H 1DT
Telephone:	01-723 9408
Price:	£59.95
Configuration:	any Apricot
Conclusion:	Effective, impressive; looks good, plays a good game. Recommended.



Workshop

The BASIC Workshop

Simplified screen-handling plus input validation

by David Holland

Confused by control codes? Fed up with invalid data creeping into your work? Then CHECKER.BAS is for you. With this set of subroutines appended to your programs you'll be able to:

- Assign the most common escape sequence for screen control to easily-used variables
- Validate data entered at the keyboard for numeric, monetary or date format
- Prevent a user skipping past a field by hitting Return when data ought to be entered there
- Optionally amend fields by pressing the ESCape key
- Check that a valid date has been entered.

Setting it all up

Before you can start using the routines you must load MSBASIC and type in the code shown in the listing. Save this on disk in ASCII format (by using SAVE "CHECKER.BAS",A).

When you want to use CHECKER, make sure your own program is loaded and then copy the subroutines over (MERGE "CHECKER.BAS"); provided your code doesn't use line numbers higher than 40000 you should find that the routines are now appended to your program.

Finally, at the beginning of your program add the line ...

```
TRUE=-1: GOSUB 40000
```

It goes without saying that your program shouldn't use any of the line numbers in CHECKER.BAS (otherwise you'll overwrite your own work!) and that your code shouldn't use any of CHECKER's variables. Besides those listed in the 'Screen control codes' table, CHECKER.BAS uses these variable names:

H	V	FL
P$	Y$	Z$
DAY	DATEOK	FORCE
DT(1)	DT(2)	DT(3)
NUMERIC	MONEY	

How does it work?

CHECKER.BAS is divided into four sections — for SCREEN CONTROL, KEYBOARD INPUT, KEYBOARD AMEND and DATE VALIDATION.

Screen Control

BEL$	sound the bell
CBL$	start the cursor blinking
CST$	stop the cursor blinking
CLI$	clear from the current cursor position to end of the line
CLP$	clear from current cursor position to end of the page
CLS$	clear the entire screen and put the cursor in the home position
CON$	set the cursor on
COF$	set the cursor off
CRB$	move the cursor back one space
CRF$	move the cursor forward one space
CRU$	move the cursor up one line
INV$	set inverse video on
NRM$	set inverse video off
UOF$	set automatic underlining off
UON$	set automatic underlining on

The Screen Control Codes routine assigns escape code sequences to variables as shown in the table. Each of these variables must be preceded by a PRINT command when you come to use them — for example, PRINT BEL$ will sound the beeper. Similarly, to print the word the word 'Apricot' in reverse video with underlining your program should contain the line PRINT INV$ UON$ "Apricot" NRM$ UOF$. (Note the use of NRM$ and UOF$ to turn the reverse video and underlining off again.)

This section also includes the definition for a function that places the cursor at any position on the screen. It takes the form:

```
DEF FNP$(H,V)
```

... where H and V are the horizontal and vertical co-ordinates of the cursor position. So, to print 'Apricot' at a position 10 columns across the screen and 15 rows down, you'd use the statement:

```
PRINT FNP$(10,15) "Apricot"
```

And to clear the screen from column 8 in row 12, you would say:

```
PRINT FNP$(8,12) CLP$
```

Validating data

The KEYBOARD INPUT subroutine will let you define and validate data as *numeric* (integer), *money* (any value to two decimal places), or *date* formats. You will also be able to force data to be entered into any field — it won't be possible to bypass the field.

All of this is done by assigning the value TRUE to the variables NUMERIC, MONEY, DAY and FORCE before calling the subroutine.

In addition you can define the length of a field to be input by setting the variable FL. A row of dots equal to the length of the field will then be displayed on the screen; as you enter data into the field, each period will be overwritten — and if you delete any characters, they'll be replaced with

periods. The routine will not let you enter more than the defined length of the field, either.

The routine sits at line 41000, so call it with a GOSUB 41000. Once the data has been validated it is returned from the subroutine in the variable Y$; this you should then assign to a more permanent variable in your program.

Some examples should make things clearer:

- **To enter a numeric integer field six digits long:**

NUMERIC=TRUE: FL=6: GOSUB 41000

Here you won't be able to enter anything other than a digit. Of course, if you want to be able to enter any combination of alpha numeric characters in a field, don't assign the TRUE statement: just define the field length FL.

- **To enter a monetary field eight digits long (including the decimal point):**

MONEY=TRUE: FL=8: GOSUB 41000

Don't key in the decimal point itself, just the figures — the subroutine will display your entry to two decimal places.

- **To enter a date:**

DAY=TRUE: FL=6: GOSUB 41000

The field length must be six digits long, and you should enter the date in the format *ddmmyy*. The subroutine will check for a valid date and display it in the format *dd/mm/yy*.

- **To print the word 'Date' at position 1,10, input a date at position 6,10, force an entry to be made, and finally store the valid date in a permanent variable:**

```
PRINT FNP$(1,10) "Date"
DAY=TRUE: FORCE=TRUE
FL=6 H=6:V=10
GOSUB 41000
CURRENTDATE$=Y$
```

Note the use of H and V to define the cursor position for the data field. Where data entry is to take place at a specific point on the screen, it's imperative that you assign the cursor position to H and V rather than use the statement PRINT FNP$(6,10).

Note also that the value returned from the subroutine in Y$ is assigned to the permanent variable CURRENTDATE$. Otherwise the data will be lost the next time this subroutine is called.

Amending data

As you might expect, the KEYBOARD AMEND subroutine will let you make amendments to data which has already been entered and stored within your program. A 'greater than' prompt will appear to the left of the data field; you can either the ESC key to amend the data or Return to accept it as it is.

If you do press ESC, the data on the screen will be replaced by an appropriate number of dots and you'll be able to re-enter the data as described in the KEYBOARD INPUT routine above.

This is called with a GOSUB 42000 statement. Taking the example above, this is how you might write an amend procedure:

```
PRINT FNP$(1,10) "Date"
Y$=CURRENTDATE$: DAY=TRUE: FORCE=TRUE: FL=6
H=5: V=10:
GOSUB 42000
CURRENTDATE$=Y$
```

If you were to omit the statement Y$ = CURRENTDATE$ at the beginning, you'd have problems when Return is pressed to make no amendments) — Y$ would then return a null from the subroutine and the value in CURRENTDATE$ would be destroyed.

Note too that the value of H is 5 and not 6 as before. This makes room for the prompt arrow.

Validating a date

The DATE VALIDATION subroutine will check the validity of any date entered as *ddmmyy* — for instance, '281185' for 28 November 1985. You cannot use full stops or slashes within the date; and you must not express the year in full (as '1985').

Within the 'ddmmyy' format, variable DT(1) holds the value of 'dd': DT(2) holds the value of 'mm': and DT(3) holds the value of 'yy'.

Although the date validation is an overall part of CHECKER.BAS, with a little modification it could be a stand-alone procedure. ◆

CHECKER.BAS

```
40000  REM ************************
40010  REM * SCREEN CONTROL CODES *
40020  REM ************************
40030  BEL$ = CHR$(7)
40040  CBL$ = CHR$(27)+CHR$(50)
40050  CLI$ = CHR$(27)+CHR$(89)
40060  CLP$ = CHR$(27)+CHR$(74)
40070  CLS$ = CHR$(27)+CHR$(69)
40080  CON$ = CHR$(27)+CHR$(121)+CHR$(53)
40090  COF$ = CHR$(27)+CHR$(120)+CHR$(53)
40100  CRB$ = CHR$(27)+CHR$(68)
40110  CRF$ = CHR$(27)+CHR$(67)
40120  CRU$ = CHR$(27)+CHR$(65)
40130  CST$ = CHR$(27)+CHR$(51)
40140  INV$ = CHR$(27)+CHR$(112)
40150  NRM$ = CHR$(27)+CHR$(113)
40160  UOF$ = CHR$(27)+CHR$(49)
40170  UON$ = CHR$(27)+CHR$(48)
40180  DEFINT H,V
40190  DEF FNP$(H,V) =
       CHR$(27)+CHR$(89)+CHR$(31+V)+CHR$(31+H)
40200  RETURN
41000  REM ******************************
41010  REM * KEYBOARD INPUT SUBROUTINE *
41020  REM ******************************
41030  PRINT FNP$(H,V);
41040  Y$ = ""
41050  PRINT STRING$(FL,CHR$(46))" ";
41060  PRINT FNP$(H,V);
```

```
41070 Z$ = INKEY$
41080 IF Z$ = "" THEN GOTO 41070 ELSE IF Z$ =
CHR$(13) OR Z$ = CHR$(8) THEN GOTO 41100
41090 IF ((MONEY OR NUMERIC OR DAY) AND
(ASC(Z$)<48 OR ASC(Z$)>57)) THEN PRINT ".";CRB$;:
GOTO 41070
41100 IF Z$ = CHR$(13) THEN GOTO 41210 ELSE IF
Z$<>CHR$(8) THEN Y$ = Y$+Z$ ELSE IF Y$ = "" THEN
GOTO 41040
41110 IF Z$ = CHR$(8) THEN PRINT CRB$;".";CRB$;:
ELSE GOTO 41150
41120 Y$ = LEFT$(Y$,LEN(Y$)-1)
41130 ON -(LEN(Y$)<1) GOTO 41040
41140 GOTO 41070
41150 PRINT Z$;
41160 ON -(LEN(Y$)>FL) GOTO 41180
41170 GOTO 41070
41180 PRINT CRB$" "CRB$;
41190 Y$ = LEFT$((Y$),FL)
41200 GOTO 41070
41210 PRINT STRING$(FL-LEN(Y$),CHR$(32))
41220 IF (Y$ = "" AND FORCE) THEN 41030
41230 IF DAY THEN GOTO 43030
41240 IF NUMERIC THEN PRINT FNP$(H,V) SPACE$(FL)
:PRINT FNP$(H,V) Y$
41250 IF MONEY THEN PRINT FNP$(H,V) SPACE$(FL)
:PRINT FNP$(H,V);: PRINT USING "###.#";VAL(Y$)/100
41260 IF DAY THEN PRINT FNP$(H,V) SPACE$(FL)
:PRINT FNP$(H,V);: PRINT
LEFT$(Y$,2);"/";MID$(Y$,3,2);"/";RIGHT$(Y$,2)
41270 IF FORCE THEN IF Y$ = "" THEN 41030
41280 NUMERIC = NOT TRUE
41290 MONEY = NOT TRUE
41300 DAY = NOT TRUE
41310 FORCE = NOT TRUE
41320 RETURN
42000 REM ****************************
42010 REM * KEYBOARD AMEND SUBROUTINE *
42020 REM ****************************
42030 PRINT FNP$(H,V) ">"
42040 Z$ = INKEY$
42050 IF Z$ = "" THEN GOTO 42040
42060 IF Z$ = CHR$(27) THEN PRINT FNP$(H,V) " ": H =
H+1 :GOSUB 41030: ELSE IF Z$ = CHR$(13) THEN PRINT
FNP$(H,V) " " :RETURN ELSE GOTO 42040
42070 RETURN
43000 REM ****************************
43010 REM *DATE VALIDATION SUBROUTINE *
43020 REM ****************************
43030 DT(1) = VAL(LEFT$(Y$,2))
43040 DT(2) = VAL(MID$(Y$,3,2))
43050 DT(3) = VAL(RIGHT$(Y$,2))
43060 DATEOK = DT(1) * DT(2) * (DT(2)<13) *
(DT(1)<31 + ((DT(2) = (DT(2)>7)) AND 1) + (DT(2) = 2) *
(2+((DT(3) AND 3) = 0)))
43070 IF DATEOK THEN GOTO 41240 ELSE GOTO 41030
43080 RETURN
```

Classified

Free Small Ads

Classifications in this free Classified Ads section could include these:

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

Help

■■■ HELP GROUP FORMING 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.

Sales

■■■ MAGAZINE BINDERS — we have acquired the last few 16 BIT COMPUTING binders. Sturdy, black with tasteful silver logo, hold 12 A4-size magazines (or even newsletters ...) Clearance price is £1.20 each, and we've only got about 40 of them. Cheques payable to TP GROUP, please.

■■■ KODE 43 TELEPRINTER — ready for use as a typewriter (has keyboard) and/or as a 30cps matrix printer (uses standard 12in 80-column paper). Excellent working order. Offers to G T Hunter-Blair on Ross-on-Wye (0989) 62013.

Misc

■■■ IBM MAG CARD TYPEWRITER still on offer at no cost to anyone willing to collect from Islington. Aged, and not currently working, but presumably fixable (by a school's enthusiast?). Comes with manual, a few mag cards, and the Selectric typewriter unit — beautiful print quality, proportional spacing, etc. Contact 01-609 3800.

Reports from the front line

A round-up of Apricot news, views and events

Having gutted Victor of its top European mangement (Jurgen Tepper *et al*) the recently-arrived PC-compatible manufacturer **Tandon Computer** is starting on Apricot (the current austerity measures have meant that there's a lot of spare bodies around). Tandon's new UK MD is **James Minotto**, ex-Computerland and ex-Interface Group — but most recently ex-Apricot.

So he's collected several other Apricot people with him at Tandon, including some good names: Ray Parsons and Ray Rose on the support side, Mike Lucas and Tony Plackett on sales development, Ann Barnbrook on training — and, most spectacularly, **Chris Buckham**. He's the former Apricot director who set up Apricot Inc and then AT Computerworld, but no-one seems to be hoiding either of those against him ... Buckham joins Tandon on a consultancy contract to set up distribution.

Xen disk drop?

'Rumours denied' dept — we heard that the 20MB **Xen hard disk** has been the subject of a **manufacturer's recall** because of 'bad media' problems. We also heard that units already out in the field *wouldn't* be recalled until the users had problems; only Apricot's stock of drives was going back to the States for replacement. All this may or may not be connected with reported CHKDSK problems. Anyway, despite the excellence of our source, we're happy to report that Apricot spokesperson **Jes Dorrell** categorically denied the story.

We name the man

Apricot's new **Customer Relations Manager** is a pleasant enough apologist called **David Tennant** who can be reached on **021-511 2284.**

Preprint sold

Apricot's belt-tightening after last year's poor financial performance has now seen the departure of one of its longer-established diversifications: **ACT Preprint**, the group's computer forms manufacturer, has been sold to **Aspen Communications**. Aspen has fingers in various media and computerish pies, and it owns a printer called **Walker**. Preprint, which quite recently installed a lot of new equipment, has always done quite well in a modest way — in the first nine months of last year it made £253,000 **profit** on a £2.6 million turnover.

Buy now while stocks last

On the Good News side, Apricot agents have coincidentally picked up two nice blue-chip European **contracts**. The Turkish distributor **Halafatogh** is to supply 365 F1Es to Turkish secondary schools for £250,000 (thought the F1E had been dropped?); and Conservative members at the European Parliament are getting a clutch of 15 Xens and Xis from the Belgian distributor **System** (but then Apricot has always been very friendly with the Tories).

Read all about DOS 3

A source at Apricot tells us that the best technical reference manual for Apricot's MS-DOS 3.1 on the Xen doesn't come from Apricot, it comes from **IBM** — whose dealers sell it for £95.

Numbers game (Pt 1)

Apricot claims that last month it built its **100,000th** micro since the October 1983 launch. At the Apricot Show, **Brian Androlia** also said 1,200 Apricot Network systems had been sold — rather less than other Official Pronouncements would have us believe.

Numbers game (Pt 2)

The Paris-based market research organisation **Intelligent Electronics** says 1.2 million microcomputers were sold in Europe last year; and 1986 should see an overall rise of 38 per cent, with some markets exploding (Italian shipments should rise by over 70 per cent) and others growing more sedately — though our own view is less optimistic than the 27 per cent growth forecast for the UK. Overall, **IBM** is given 33 per cent of the market; the next largest supplier is **Olivetti** with 10.5 per cent. **Apricot** rates below **Commodore** and **Apple**, with just 4.5 per cent of shipments. All other vendors are lumped together at a mere 3.9 per cent. ◆

DELTA passwords

by P J Gaywood

As a keen subscriber to APRICOT FILE (and previously to 16 BIT COMPUTING), I have always been intrigued by the small but helpful hints sent in by other readers.

I have at last come across just such an item myself — a simple yet useful explanation of how to find and decode passwords used by Compsoft's DELTA file manager to protect its files. And very poorly protected they are too, as you will see ...

We set ourselves the task of finding these passwords when a colleague returned slightly merry from a celebratory lunch to spend the afternoon creating and passwording a number of mask files in DELTA. Unfortunately next day he hadn't a clue what any of the passwords actually were; and he was faced with the daunting prospect of recreating all the masks unless we could help him find the missing passwords.

To cut a long story short, we first found them the hard way via DEBUG — and then discovered that you can simply TYPE out the files to disclose passwords.

DELTA passwords come in three styles:

- 'DELTA ' ... which is equivalent to no password at all
- an encoded password, used in mask files
- a straight password as used in definition files

Figures 1 to 3 illustrate how these look when the file is viewed with DEBUG. These passwords can also be seen by printing the file (the screen display zips by too quickly). Use TYPE FILENAME.MSK > PRN or TYPE FILENAME.DEF > PRN; the password appears in the first few lines printed (as in Figure 4).

The encoded password is simple to interpret by taking the next character in the ASCII codes list. For example, the encoded version of the password '**Fred** ' would be '**Gsfe!!!!**'. Note that all passwords must be eight characters long but they can include spaces. ◆

Figure 1: DELTA-space

```
2E7D:0100  50 4D 20 20 20 20 20 20-50 4F 54
           45 4E 54 49 41  PM      POTENTIA
2E7D:0110  4C 20 4D 45 4D 42 45 52-53 2E 2E
           2E 2E 2E 2E 2E  L MEMBERS.......
2E7D:0120  2E 2E 2E 2E 2E 2E 55 0A-09 55 0A
           09 00 00 00 00  ......U..U......
2E7D:0130  25 01 45 00 20 00 00 00-00 61 01
           21 44 45 4C 54  %.E. ....a.ELT
2E7D:0140  41 20 20 20 00 00 00 00-22 04 4A
           00 44 00 00 00   A  ....''.J.D...
2E7D:0150  00 00 00 FF FF 54 49 41-4C 20 00
           00 00 00 20 00  .....TIAL .... .
2E7D:0160  00 00 00 00 00 FF FF 2E-2E 2E 2E
           2E 00 00 00 00  ................
2E7D:0170  20 00 00 00 00 00 00 FF-FF 01 45
           00 20 00 00 00   .........E. ...
```

Figure 2: Encoded password

```
2E7D:0100  38 30 2C 32 34 0D 0A 22-50 4D 4F
           4F 22 22 22 2C
           80,24..''PMOO''''',
2E7D:0110  22 64 70 6E 71 21 21 21-21 22 0D
           0A 32 35 0D 0A
           ''dpnq!!!!''..25..
2E7D:0120  32 39 0D 0A 32 35 35 0D-0A 32 35
           35 0D 0A 32 35  29..255..255..25
2E7D:0130  35 0D 0A 32 35 35 0D 0A-32 35 35
           0D 0A 32 35 35  5..255..255..255
2E7D:0140  0D 0A 32 35 35 0D 0A 32-35 35 0D
           0A 32 35 35 0D  ..255..255..255.
2E7D:0150  0A 32 35 35 0D 0A 32 35-35 0D 0A
           32 35 35 0D 0A  .255..255..255..
2E7D:0160  32 35 35 0D 0A 32 35 35-0D 0A 2D
           31 2C 2D 31 2C  255..255..-1,-1,
2E7D:0170  2D 31 2C 2D 31 2C 2D 31-2C 2D 31
           0D 0A 30 2C 30  -1,-1,-1,-1..0,0
```

Figure 3: Straight password

```
2E7D:0100  50 4D 4F 4F 20 20 20 20-50 6F 74
           65 6E 74 69 61  PMOO    Potentia
2E7D:0110  6C 20 4D 65 6D 62 65 72-73 20 6F
           6F 2E 2E 2E 2E  l Members oo....
2E7D:0120  2E 2E 2E 2E 2E 2E 55 0B-1D 55 0B
           1D 00 00 00 00  ......U..U......
2E7D:0130  25 01 45 00 20 00 00 00-00 61 01
           21 50 45 54 45  %.E. ....a.ETE
2E7D:0140  52 20 20 20 00 00 00 00-22 04 4A
           00 41 00 00 00  R       ''.J.A..
2E7D:0150  00 00 00 FF FF 74 69 61-6C 20 00
           00 00 00 20 00  .....tial .... .
2E7D:0160  00 00 00 00 00 FF FF 2E-2E 2E 2E
           2E 00 00 00 00  ................
2E7D:0170  20 00 00 00 00 00 00 FF-FF 01 45
           00 20 00 00 00   .........E. ...
```

Figure 4: Encoded password seen from a TYPE

```
80,24
"PMOO""","dpnq!!!!"'
25
29
255
255
255
0,0,0,""
```

Software Round-up

Brikat expands

Pegasus purveyor **Brikat** continues its expansion — it's just bought two dealers. It's paying £500,000 for **Globestyle** in Hull and Doncaster and around £140,000 for **South Coast Business Machines** in Bournemouth.

Read all about it

Getting printed or typed text easily into your computer easily (and especially without re-keying it) has long been one of the Grails of computing — which is why we were sad to see the demise of the £400 OmniReader. That was pretty limited, though; software house **Southdata**, better known for its SuperFile database package, has come up with a clever-looking OCR reader plus software under the name **Typereader**. It reads typesetting at roughly twice the speed of copy-typing.

At present it's expensive (£25,000) and needs a go-faster processor for itself with a lot of memory in the host machine. But Southdata reckons that some outside cash for development could drop the price to £6,000 and put the thing happily on to 16-bit micros.

Act now

The **Data Protection Act** is going well in generating a mini-industry of explanatory books, seminars, and associated punditry (including our own essays). At least this one's **free** ... provided you're a member of the **Institute of Chartered Accountants in England and Wales**. The ICAEW's Publications Department (399 Silbury Boulevard, Milton Keynes MK9 2HL) has a helpful 'guidance note' on registration. Members only, though.

D-I-Y Data Protection

You'll pay for this one — **ADM**, one of the country's biggest computer training organisations (it normally teaches in excess of 350 souls per annum) has launched a self-help kit aimed at small to medium companies keen to comply with the Act's requirements. ADM's marketing director **Ray Hunnibal** opined that "we have adopted an approach which should save companies considerable time and expense". The kit costs £65 (inc P&P) and is available from **0223 314545**.

Watch that Xen!

Burglars are getting more selective — **Missing Link Software** has an office full of different machines for implementations of its rather fine Personnel Assistant software, but thieves took only £20,000 worth of Xens and Apricot PCs. It's the second Apricot-only theft from Missing Link's North London premises in six months. So if you see a cheap secondhand Xen on offer, it's worth contacting Missing Link on 01-625 5111 ...

And if you're trying to run a personnel department in a smaller company, it's probably worth contacting Missing Link anyhow. Its latest product is **TPA 200**, a small version of Personnel Assistant that runs on the F2 up. It holds up to 200 employee records.

SPI scores

There's still City money for high-tech — **SPI**, purveyor of the integrated package **Open Access**, has pulled a whopping £4.6 million of new capital for development from a group of European and US banks collected by **Rothschilds** in Germany.

At the Which Computer Show last month, SPI was doing very little of note product-wise apart from pulling discrete sections out of Open Access (spreadsheet with goal-seeking and 3-D graphics, an applications-generator database) plus networking, a slimmed-down Open Access Junior, and Open Access for GEM. Apricot versions of all are available or promised; SPI is on 0735 74081.

What's in a name?

Ansible Information, the SuperWriter-oriented software house run on a part-time basis by **Chris Priest** and our regular columnist **David Langford**, has renamed its **P.A.S.D.** (Professional Author's SuperWriter Disk) as **ASD** (Ansible SuperWriter Disk). Nothing else appears to have changed ...

Another of Ansible's products has just been recommended to us (and not by David); it's **AnsiblePatcH**, a £15 utility for users of the Portable and F Series that greatly improves the clarity of the SuperWriter display. The price covers VAT and despatch, Ansible is on 0672 62576. ◆

So you don't understand the Statutory Sick Pay scheme? Or it's clear as a bell, but you're snowed under by the record-keeping it requires? Well, on 6 April it gets a bit worse ... new rules come into force then that mean the employer's liability for a 'period of incapacity for work' (four or more days off) is increased from eight to 28 weeks. Make sure your payroll package will cope with the change.

Workshop

UNDELETE

the True Story

by Tony Weyman

I picked up a copy of the DEALS FOR READERS disk no. AP008 at the Apricot Show last month, and I was horrified by the UNDELETE.DOC file thereon.

This procedure for using DEBUG to recover disk files that have accidentally been DELeted is positively dangerous. And it does *not* undelete files at all.

All it does is to replace their names in the directory. On first sight this might not appear to be particularly harmful: but the directory entry contains a pointer to the start of the file in the File Allocation Table. The FAT is then read to determine the sequence of file clusters across the disk. All the pointers in the FAT are restored to zero when a file is deleted to indicate to MS-DOS that these clusters are free; the next file added to the disk will overwrite these locations.

If you use the UNDELETE procedure only to find that the 'undeleted' file is not working, you may decide to erase the partially recovered file ... which could mean you've just deleted any new file that's been added. And when you figure that this too have been screwed up, deleting *that* will blow another new file.

The original UNDELETE.DOC was obviously written when MS-DOS Version 1 was in use. It works perfectly well if you are using (say) MS-DOS 1.25 on the Sirius ... but Apricots were never issued with MS-DOS version 1.

If you want to be able to undelete a file on an Apricot, a little understanding of the file handling is needed. In which case, read on ...

The locker room

To be able to undelete a file stored by MS-DOS it is essential to have a little understanding of how MS-DOS keeps track of the files on disk.

Imagine a collection of lockers in a club changing room, each locker being identical in every way except for the number on its door. Naturally the lockers are numbered consecutively; and any one is easily found if you know the number you require.

Only one rule applies: club members may store as much as they like in the locker room — provided that every time they require another locker, they take the lowest-numbered free locker.

The first users will find it quicker to claim adjacent lockers; in fact they have no choice if they obey the rule. But when there are only a few free lockers scattered around the room, members will have to store their belongings where they can.

A system is required to help members (a) find their belongings and (b) locate free lockers to use when they need more space.

It works something like this: on the club notice board is a directory that lists those members who have property in the room. Next to each name is some incidental information such as the date and time they last used the lockers and their status in the club; more important, there's also a note of the number of their first locker and the total number of lockers they have in use.

A second document pinned to the board is the Locker Allocation Table (old hands call it the 'LAT'). This is a list of numbers with an entry for every locker in the room. The entries are made in a rather unusual manner: there's no point in numbering the entries as the first one relates to the first locker, the second entry to the second locker, and so on to the last. Instead, each entry contains the number of the *next* locker that the owner is using. If the locker is free, the LAT entry is zero; if the owner has no more lockers then the entry is 'FFF' (hexadecimal for finished).

So to trace any particular set of belongings you first look up the name of the owner in the directory list and note the number of his or her first locker. Then you go to the Locker Allocation Table and follow the chain of locker numbers (opening each locker as you go) until you get to the final one (a locker marked 'FFF' on the LAT).

There are some rules for members who wish to leave the club:

- Members are not allowed to remove the contents from their lockers
- On leaving, they must mark their directory entry by erasing the first letter of their name and replacing it with a lower case 'e' (all names are entered in upper-case, so an 'e' is a clear indication that the member has indeed left)
- They must change all their old LAT entries to zero.

This way, a *new* member can look along the LAT for the first zero entry. They may then throw out the contents of the locker and fill it with their own goodies, write their name (block capitals please) and other details in the directory list in place of the first entry starting with an 'e', and finally fill out the LAT entries as they use up more empty lockers.

The story you have just read is true. Only the names have been changed to protect the innocent — for 'locker room' read *disk*, for 'locker' read *cluster*, for 'member' read *file*, and for 'LAT' read *FAT*.

How files are deleted

When a disk is formatted, MS-DOS sets aside two special areas which it uses to keep track of the files. These are the Directory and the File

Allocation Table (known as the FAT).

The directory contains one entry for each file on disk. Each entry is 32 bytes long, and is made up as follows:

BYTES	USAGE
0 – 7	file name
8 – 10	name extension
11	file attribute
12 – 21	[not used]
22 – 23	time
24 – 25	date
26 – 27	FAT and sector reference
28 – 31	file size

You can browse through the directory with DEBUG by loading the appropriate sectors — more about this later — and using the 'D' (for Display) command.

The 32-byte format is really convenient for the DEBUG user, as each entry is displayed neatly on two lines of text. The first line is the filename and extension followed by the attribute byte, the second contains the file data. It looks like this:

```
FILENAME EXTAxxxx
xxxxxxTT DDRRSSSS
```

... where 'A' is the attribute, 'T' is the time and 'D' the date of creation, 'R' is the FAT and cluster reference, and 'S' is the file size. Unused bytes are shown as 'x'.

Take note of the positions of the reference and size bytes — we'll be using these to undelete our files.

When a file is deleted, it's not actually erased physically from the disk. In the directory, the first character of its filename is changed to E5 hex or 'e'; no other information is changed in the directory entry (but there *are* changes in the FAT, so you can't simply change the filename back to recover the file). The 'e' is only a flag to indicate to MS-DOS that this entry is available for overwriting when a new file is added — ever wondered why new files pop up in the middle of a DIR list and not at the end?

In contrast to the directory entries, the FAT data is most difficult to decipher using DEBUG. There is one entry for each cluster on the disk; each entry is 12 bits long (one and a half bytes).

The reference bytes in the directory entry indicate the starting point in the FAT for the specified file. This in turn points to the next location in the FAT (and on the disk) ... and so on, until the end of the file is reached. The FAT entry for that contains the flag 'FFF'.

Any free clusters on the disk are indicated by a FAT entry of '000'. When a file is deleted, the chain of FAT entries for the progress of the file across the disk are reset to '000'; as with the 'e' directory flag, this indicates to MS-DOS that these clusters are free for overwriting.

Incidentally, this is why you must *not* write anything to disk if you wish to recover a lost file. The FAT entries that *point* to the clusters may be set to '000', but the clusters themselves still contain the bits of your deleted file. Undeleting it involves re-establishing the original FAT pointers; so if you've added something to the disk in the interim, some of the clusters you need for the original file may have been taken for the new one. The safest course of action to take is to immediately remove the disk from the drive and slide the protect slot open.

Cutting through the FAT

In theory it's possible to make a painstaking sequential search through the disk sectors and manually reorganise the FAT. In practice this would be far too laborious to be worthwhile unless you're trying to save a lifetime's work — particularly because most of the files on a disk will have been laid down on contiguous sectors, and these can be resurrected with ease.

The instructions which follow can be executed more quickly than they can be read.

For a start, load DEBUG (it's available on DEALS FOR READERS disk no. AP002). Now place the disk with lost files in drive B and put a formatted disk in drive A (it's ok to use a hard disk if you have an Xi or F10). Now load the first 80h sectors into memory (that's 80 in hex):

```
>L 100 1 0 80
```

... and search for the deleted file:

```
>S 0 FFFF E5'ILENAMEEXT'
```

DEBUG will return the offset memory location of the first byte of the directory entry you require. Here, of course, 'ILNAMEEXT' is the filename and extension that you wish to find with the first character omitted. Note that there's no full stop between the name and extension; and pad out the name with spaces up to the seventh character, pad out the extension to three places with spaces.

Or you could just type ...

```
>S 0 FFFF E5
```

That will tell you every occurrence of byte E5, a lower-case 'e', so you should get a list of possible deletion locations.

If the file is in a subdirectory an extra step is needed to find the subdirectory entry first. Do this with ...

```
>S 0 FFFF 'SUBDIRNAME'
```

... in the same way as you find a file name.

Once you've located the deleted file's name, display the entry with ...

```
>D offset
```

... where 'offset' is the value returned by DEBUG in the previous step.

Remember the two-line directory entry that DEBUG will display:

```
FILENAME EXTAxxxx
xxxxxxTT DDRRSSSS
```

That's what you should be seeing. Note the values of the last six bytes in the second line, the reference and size bytes RRSSSS.

DEBUG can load a maximum of 64K bytes into memory directly from disk. If the file size is less than 64K you're in with a fighting chance of saving the entire file. This is easy to check: the last two bytes in the directory entry will be 00 00.

Now to locate the first disk sector containing the file. Reading from left to right the reference bytes are R1 and R2; the formula to find the sector is ...

R2R1 * constant + offset

For a 720K disk the constant is '2', for single-sided disks it is '1'. The 10MB hard disk constant is '8'.

The added offset can vary. Try 'E' first, but I have found a value of '5' on some double-sided and 'B' on single-sided disks — it depends on your version of DISK.COM.

Don't forget that these values are all in hexadecimal, but DEBUG can do the arithmetic work for you. Start with ...

```
>H R2R1 R2R1
```

Two values are returned by this; one is the result of addition and the other of subtraction of the two numbers you've typed in. Obviously in this case the second answer is 00. Now add the constant:

```
>H result constant
```

The first value returned by this is the sector number.

Now you can load the actual file with ...

```
>L 100 1 sector 80
```

Browse through the memory with the D display command — 'D100' to start, and 'D' alone to continue displaying consecutive areas of memory. If the file is in ASCII the dump on the righthand side of the display will be easily recognised.

If all is well you can set about saving the memory image as a file on the *other* drive. Create a name for it by ...

```
>NA:FILENAME
```

... and enter the size of the file into the CX register by typing ...

```
>RCX <CR> S2S1
```

'S1' and 'S2' are the leftmost size bytes that you noted earlier.

Write it away to disk by typing 'W' — the command requires no arguments this time because the filename and size have been entered into the appropriate places earlier. 'Q' quits from DEBUG, and you can check the directory on drive A for the newly saved file.

If the file you need is in a subdirectory, two extra steps are required. Firstly find the subdirectory entry. Do this with ...

```
>S 0 FFFF 'SUBDIRNAME'
```

Then follow the steps above to locate and load into memory the subdirectory itself. A subdirectory is simply a file situated in the data area of the disk and can be handled in the same manner.

It really is easier than all this explanation appears. A sample session should make things clear: for this example I have accidentally deleted a file called FUNKY.DOC from a double-sided floppy in drive B.

Sample run

```
A:
A> MD UNDEL
A> CD\UNDEL
A> DEBUG

DEBUG-86  version 1.07 >L 100 1 7 10
>S 0 FFFF E5'UNKY   DOC' 20E0:0160
>D 160
20E0:0160  E5 55 4E 4B 59 20 20 20-
           44 4F 43 20 00 00 00 00
           eUNKY   DOC ....
20E0:0170  00 00 00 00 00 00 02 AC-
           44 0C 0B 00 80 11 00 00
           ........D.......
>H B B
0016 0000
>H 16 E
0024 0008
>L 100 1 24 80
>NA:FUNKY.DOC
>RCX
CX 0000
:1180
>W
Writing 1180 bytes
>Q
A> DIR
 Volume in drive A is Apricot
 Directory of A:\UNDEL

.            <DIR>        86-02-12   14.03
..           <DIR>        86-02-12   14.03
FUNKY   DOC       4480    86-02-12   14.07
        3 File(s)       4050944 bytes free
```

Notes and afterthoughts

It will be clear to the imaginative reader that there are some limitations and some certainties in the undelete procedure — among them:

- a file that is small enough to fit into one cluster will *always* be fully recovered
- files that are not contiguous *cannot* be recovered without further investigation
- files over 64K can be recovered in sections and concatenated with the COPY or by your word processor.

Don't forget that all values are in hexadecimal. If you have loaded 80 sectors from sector 80 onwards you have reached sector 100, not sector 160! Incidentally, 80 sectors is 64K bytes; that is is the maximum value which can be loaded or written by DEBUG.

The syntax for the Load and Write commands of DEBUG is as follows:

command offset drive sector extent

... where *command* will be 'L' or 'W'; *offset* is the memory offset (usually 100); *drive* is '0' for drive A or '1' for B; *sector* is the starting sector on disk; and *extent* is the number of sectors to load.

A cluster is the smallest block of information that MS-DOS can write to a disk. The size varies according to the medium:

```
Single-sided 315KB floppy:
1 cluster = 1 sector = 512 bytes (h200)

Double-sided 720KB floppy:
1 cluster = 2 sectors = 1024 bytes (h400)

10MB hard disk:
1 cluster = 8 sectors = 4096 bytes (h800)
```

The directory on a 720K floppy starts at sector 7, on a 315K disk it's at sector 5.

If you have a hard-disk machine, and you're undeleting files on the hard disk, all new files must be written to drive B. Otherwise it's easiest to create a temporary subdirectory and use 'CD\TEMP' before you start; that will keep your new files separate. ◆

As soon as we received Tony's article, we replaced the original UNDELETE.DOC on Disk No AP008 with a slightly amended version of this piece.

Technote

A line-drawing .KB file

by Alastair McIntyre

I use David Langford's BORDER.XQT SuperWriter file to create good-looking start-up menus; and for that I have created a special .KB file which produces the lines by using the keypad keys in Shift mode.

I used the keyboard editor of ACTIVITY to change the numeric keypad keys. With this you first clear out the CONTROL entries for numbers 1, 2, 3, 4, 6, 7, 8 and 9. Point the arrow on to the key you wish to clear and press Return; this will show the values for that key. You then point the arrow at the eraser symbol and press Return, which highlights the symbol. Your final step is to point the arrow at the entry in the CONTROL box and hit Return. You'll then see the Null sign appear.

Now find the graphic you wish to enter. In the case of the '1' key, you're looking for a line that will take you around the corner: this you'll find in the bottom right box — send your cursor to the big arrow symbol and press Return once. Then place the arrow on the corner shape you require and press Return to load the value for that line. Take the arrow back to the CONTROL box and press Return to replace that Null with the new value.

Repeat these steps for the relevant keys until you have a box shape on your keyboard — I have keys 2 and 8 as horizontal lines, 4 and 6 as verticals; 1, 3, 7 and 9 are the corners.

When you have finished this, place your arrow on the top right symbol which puts you back to the opening menu. Type in the name of your new .KB file and load it into the box with the other .KB names — hit Return, place the arrow in the names box, press Return again. This writes the new .KB file to disk.

Your final stage is to configure the system to this .KB file. Return to the name box (where you just typed the name for the new .KB file), hit the Return key, and drag the box to the disk symbol. Pressing Return again will write it to disk. You can also repeat this process for the computer symbol instead of disk; this loads the .KB file as the default table.

And now to use it. I employ EDLIN to create a box — bring up EDLIN, press the Tab key twice, then hit Shift-7 to produce a corner; Shift-8 draws the horizontal line; and Shift-9 gives a closing corner. This way you can generate a box very quickly and easily. ◆

Deals For Readers

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

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

How to order

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

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

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

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

Overseas orders

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

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

Contribute!

We're always looking for contributions to further £15 disks. So if you've written something — a game, a utility, a helpful batch file — why not share it with other readers? And if the ideas and techniques therein would make a useful mini-article for the newsletter, we'd certainly be interested in printing it.

And that also applies to bugs. No-one can expect this stuff to have the same rigorous checking-out that you should expect from commercial packages. We do check the material on the public domain disks, but we can't do that exhaustively — so let us know if you come across any bugs. And let us have a copy of any fixes you can make ...

Public Domain Disks

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

Apricot Disk No. 1

Order no:	**AP001**
Price:	**£15**

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

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

Apricot Disk No. 2

Order no:	**AP002**
Price:	**£15**

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

ALTER and **CHMOD** change file attributes
BASICKEYS single-key entry of Basic commands
CRF Basic cross-referencer
DEBUG the complete Apricot MS-DOS 2.11 version
EPSON sets alternative print modes on Epsons and compatibles
INDEX-84 complete index to 16 BIT COMPUTING 1984
KNEW backup utility that copies only new files

LOWER and **UPPER** change case of text files
MENU demo of simple batch-file menu system
MOVE moves files from one subdirectory to another
PLOT demo of plotting and screen resolution
RENDIR rename subdirectories
SORTDISK sorts directories in a variety of ways
TREEDIR displays a DIRectory of all files on a disk — in their subdirectories
VCOPY selective COPY
WHEREIS searches disk subdirectories for named file(s)

Apricot Disk No. 3

Order no:	**AP003**
Price:	**£15**

A bumper bundle of goodies. Highlights:
SWEEP excellent all-purpose file housekeeper
SCAVEN finds and locks out bad sector on microfloppies
MEMDUMP browse through memory
TURBOKEY key redefinition for aficionados of Turbo Pascal
SUPERWRITER XQT FILES a useful clutch of macros
HEXFILE create hex-coded ESCape sequences and the like — comes with some fun MicroScreen-driving demos

Games

Order no:	**AP004**
Price:	**£15**

Games — and nothing but games. Some in Basic, some in EXE form. 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

GSX User's Disk

Order no:	**AS005**
Price:	**£15**

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

Games No. 2

Order no:	**AP006**
Price:	**£15**

Another diskfull of Apricot entertainment. 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**

Games No. 3 — Great Britain Ltd

Order no:	**AP007**
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 too if you can. And **INHERITANCE** is an interesting investment/gambling game; you could get hooked on both!

Apricot Disk No. 4

Order no:	**AP008**
Price:	**£15**

More indispensible tools and utilities. Highlights:
ASMGEN dissassembler
132-column screen configurator
memory speed test benchmarks
improved version of **SETUP**
DIARY year-planner
Microscreen tips
undelete files using DEBUG
LABEL address label printer
BANNER prints fancy letterheads with your daisywheel printer

Demo Disks

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

BusiPost

Order no:	**AD002**
Price:	**£15**

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

ChartMaker

Order no:	**AD003**
Price:	**£15**

Scientific/technically-oriented graphics. Lots of graph-drawing functions built in.

Delta 4

Order no:	**AD004**
Price:	**£15**

The demo for the latest version of Compsoft's best-selling all-British database manager.

Letterer-Set

Order no:	**AD005**
Price:	**£15**

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

Lotus 1-2-3 Tutorial

Order no:	**AD006**
Price:	**£15**

If you're thinking of putting 1-2-3 on your Apricot, here's a chance to see just what it can do first.

MENUGEN

Order no: AD007
Price: £15

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

MicroSight

Order no: AD008
Price: £15

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

PC Programmer

Order no: AD009
Price: £15

Program generator that lets you produce complex applications software with minimal effort and no need to learn a programming language.

Praxis MicroChart Plus

Order no: AD010
Price: £15

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

Cardbox Plus

Order no: AD011
Price: £15

Particularly good demo for one of the best 'flat-file' records-keeping packages. [Not available for the Apricot F1]

Special software

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

FAST Apricot Screen Accelerator

Order no: AS001
Price: £25

Have you slyly suspected that your Apricot screen was a little, er, slow? Well, you're right. The Apricot screen *is* slo-o-w-w ... especially when it comes to scrolling about a screenful of text.

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

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

BPS Disk no. 2 — RAMdisk with WordScan

Order no: AS002
Price: £40

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

POLYPRINT

Order no: AS003
Price: £30

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

The Apricot Catalogue

Order no: AS004 [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. Based on the lists originally printed in 16 BIT COMPUTING, but regularly updated by us; currently well over a megabyte of information on six disks. 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!).

ASD — the professional's SuperWriter disk

Order no: AS006
Price: £46

Formerly known as P.A.S.D.: this is the latest version of the first (and only?) independently-produced package of upgrades to and tools for SuperWriter — produced by a couple of professional writers (one of them being our columnist David Langford) for people who want to make the most of the word processor. A packed disk, and a good piece of work. Highlights:

- **two-column** formatting and print
- expansion, improvements, correction 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**

IANKEY Crash Course

Order no: AS007
Price: £48

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.

IANKEY Two-Fingers to Touch-Typing

Order no: AS008
Price: £48

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 those enthusiastic review conclusions, and again available from us at a price £10 below RRP.

FORTRAN Utilities

Order no: AS009
Price: £50

A collection of assembler utilities for MS-FORTRAN applications, complete with test programs and link instructions. Includes:

- **ISETDT** set date
- **ISETTIM** set time
- **DATER** current date
- **TIMER** current time
- **IDRN** default drive ID
- **ISETDR** change default drive
- **IDFS** free space and total space on disk
- **ISFILE** see if a file is on disk
- **DLFILE** delete named file from disk
- **IRENAM** rename file
- **ISCHAR** reads character from keyboard buffer
- **INK** get character from keyboard
- **JOINST** concatenate two strings
- **PRCHRS** prints characters on printer

■■■ NEW ■■■

Do-To-All

Order no: AS010
Price: £15

Essential 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.

■■■ NEW ■■■

The FTU

Order no: AS011
Price: £125

The very best file transfer package we know of; we use it all the time, and that's not just because it's written by Friedman. A complete kit that enables you to transfer files between the three most popular machines — IBM PC, Apricot, and Victor/Sirius. Includes software for all three, and an 'intelligent' RS232 cable that will configure itself correctly for connection between any two of the computers. Our price is over £50 below RRP!

■■■ NEW ■■■

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 ...

The Lotus Pack

Order no: AS013
Price: £65

Useful utilities from **S&S Enterprises** for 1-2-3 and Symphony 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

BACK2123

Order no:	**AS014**
Price:	**£30**

Translates Symphony .WRK files into 1-2-3 .WKS format. Converts data, labels, formulae.

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 Word Processing Pack

Order no:	**AS016**
Price:	**£50**

Three more handy S&S utilities:

- **WSASCII** converts WordStar files to ASCII format so that they can be read by other word processor
- **WORDNO** word counter
- **READAGE** calculates the reading age (ie the comprehensibility) of text
- **SEARCH** searches text for 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 File Protection Pack

Order no:	**AS018**
Price:	**£60**

SECRET makes files invisible (or visible again) to DIR 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

The Games Box

Order no:	**AS019**
Price:	**£25**

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

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. *Not for hard disk systems, though.*

■■■ NEW ■■■

FAST

Order no:	**AS021**
Price:	**£25**

Speed up screen operations Rn your Sirius! This utility by-passes some of the in-built MS-DOS interrupts to accelerate screen-handling. Works best at the system level (eg on DIR, TYPE, etc) — so don't expect too much. But for the money, it's a cheap way to get *any* improvement in performance.

Hardware extras

Expansion Memory Boards

128KB	**AH001 £85**
256KB	**AH002 £140**
512KB	**AH003 £195**

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

Books

A book by the editor ...

The Good Computing Book

Order no:	**AB001**
Price:	**£3.00**

A tubby and packed paperback for beginners. Mostly a reference dictionary of computing terms, but also lots of introductory info. Claims to offer "everything you need to know and nothing you don't"; well, that may be blurb-writer's licence but Dennis is still pretty proud of the book.

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

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

Queries: 01-609 3800

Cheques should be made payable to TP GROUP.

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

Legal corner

The latest news from **Computer Hardware Maintenance** is that amazingly the cut-price maintenance organisation is still in business — despite the increasing number of Court Orders against it ("with over 3,000 customers there are bound to be a few problems"). The egregious **Chris Sullivan** appeared briefly on BBC TV's otherwise rather wishy-washy *Micro Live*, and when a couple of specific complaints were put to him he stalked off the set with the immortal lines "This could be a matter for fraud ... I thought this was going to be a serious discussion".

OEM out

The receiver has finally been called into the Hammersmith dealer **Office Efficiency Machines**. The company stopped trading in December, blaming its troubles on the theft of £100,000 worth of equipment a couple of months earlier; seems Sun Alliance declined to pay up because it didn't like OEM's burglar alarm. Now let that be a lesson to you ...

Here's the Action

We're told that **Action Computer Supplies**, the fast-service modest-prices consumables company we mentioned last time, as a free 0800 number — call **0800-333333** for a catalogue or to place an order.

Less is more by Apstor

Another 'Bernoulli box' from **Apstor**: it's a new half-height version of its removable cartridge disk add-on. You get two 10MB disks for your £3,450; Apstor is on 0273 422512.

Microsoft loses heads

Resignations at **Microsoft** — retail sales manager **Alastair Boyle** and his three salespeople have gone, saying they didn't fit in with Microsoft's current sales strategy of direct mail-order selling for Windows ...

Lotus marks time

Reorganisation and acquisitions were blamed for the standstill in profits at **Lotus Development Corporation**. Worldwide sales rose 44 per cent to $226 million, but profits were about the same at $66 million. The past few months have seen new plants in Dublin and Japan, new business divisions, and the acquisition both of new companies (Dataspeed, Software Arts) and new products from external suppliers.

F Users Unite!

GATEWAY USERS is a new users' group for users of Apricot's 'Gateway' machines, the F1 and F1E. It's much more of a hobbyist's group than say Apri-Soft-Exchange; **Tony Hill**, who is setting it up, says it's definitely for home users. "I'm not an expert and I'm not a programmer" he told us (he's a schoolteacher, actually) and "it's absolute agony dealing with Apricot". So he's set up GATEWAY USERS as a non-profit self-help organisation based on an chatty little newsletter. More information from him at **85 Edna Road, London SW20 8BS.**

Boom bust?

Looks as though the small-company computer boom is over. According to the latest QUARTERLY SURVEY OF SMALL BUSINESSES IN BRITAIN from the **Small Business Research Trust**, a third of the 1,000 firms polled *had* a computer — but fewer than ten per cent of the balance expected to acquire one in the future.

Roll up

Jack Palmer, proprietor of **Spaceheights Ltd**, has not moved to Brazil. He can still be found in sunny Weymouth: his TIMES Portfolio game scheme has yet to produce a winner. You may recall that he devised a program called **Jackport** which saves you the need to buy the TIMES and check the share prices against those on card. Instead you send Jack your card and he will enter your details in the Jackport database; he inputs the relevant daily data and the results are spewed out at the touch of a button.

Jack gets 15 per cent of the divvy if (when?) you win. And so far no one has: ergo Jack remains impoverished. But we all live in hope. Quoth Jack, "we came very close last week". Jack has never had the exposure he deserves — the scheme *was* mentioned in The TIMES but gave no details of how to contact him ... To get a Portfolio card send an SAE to TIMES Portfolio, PO Box 40, Blackburn, BB1 6AJ. Then, to get rich quick, speak to Jack on **0305-771974**.

Card carrying

If you live anywhere near a **WH Smith Business Centre**, you may be one of 40,000 lucky recipients of a promotional 'Executive Card'. This is a sales come-on that entitles the card-holder to register for a lucky-dip draw — three prizes of Xen, printer, "£3,000 worth of software", and a one-day course. ◆

Apologies for the late arrival of the newsletter this month. It's all part of the contemporary over-reliance on technology — we lost the use of the font cartridge that we need for producing the File on our laser printer, and we had a hell of a job finding a replacement cartridge. *And* they cost about £240 at minimum ... Still, we're buying a complete duplicate system for backup, so we should be able to get back to mid-month publication soon.