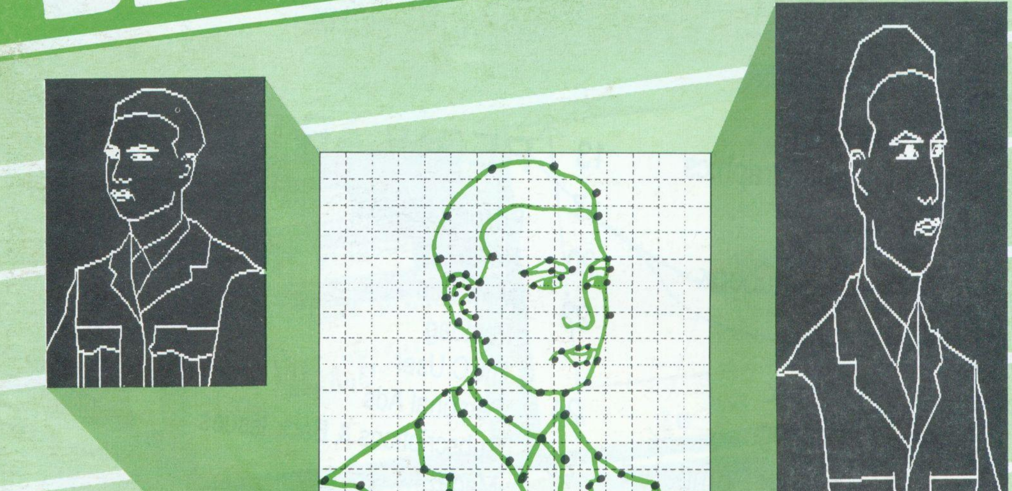
Vol.10 No.4 August/September 1991

BEEBUG

FOR THE BBC MICRO & MASTER SERIES



Practical Ways with Graphics

- TEXBASE
- MULTIPLE WINDOWS
- RANDOM SAMPLING
- USING PC DISCS

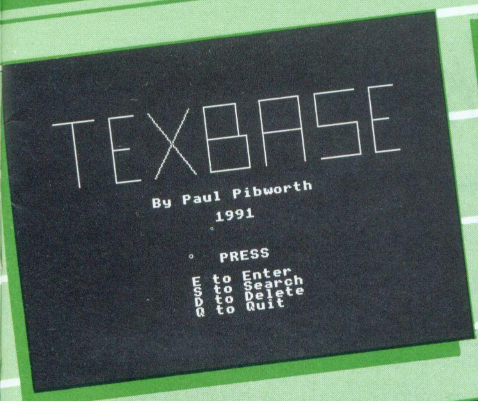
BEEBUG

Vol.10 No.4 August/September 1991

PROGRAM INFORMATION

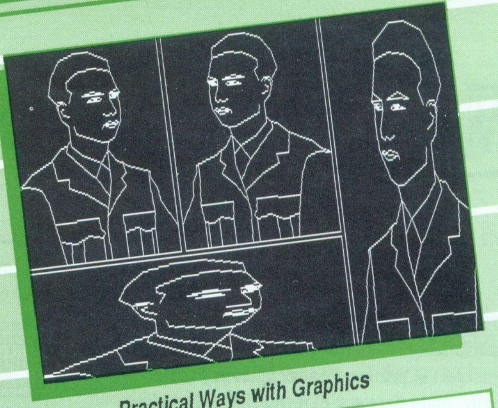
All listings published in BEEBUG magazine are produced directly from working programs. They are formatted using LISTO 1 and WIDTH 40. The space following the line number is to aid readability only, and may be omitted when the program is typed in. However, the rest of each line should be entered exactly as printed, and checked carefully. When entering a listing, pay special attention to the difference between the digit one and a lower case l (L). Also note that the vertical bar character (Shift \) is reproduced in listings as |.

All programs in BEEBUG magazine will run on any BBC micro with Basic II or later, unless otherwise indicated. Members with Basic I are referred to the article on page 44 of BEEBUG Vol.7 No.2 (reprints

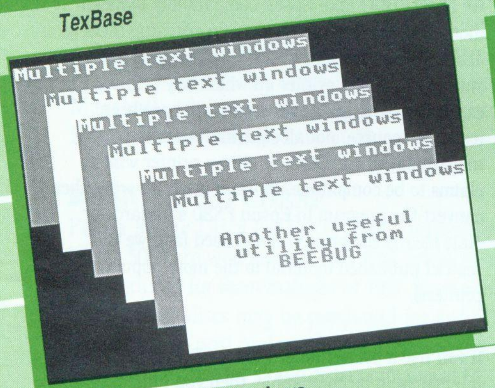


```
TEXBASE

By Paul Pibworth
1991

PRESS

E to Enter
S to Search
D to Delete
Q to Quit
```

TexBase

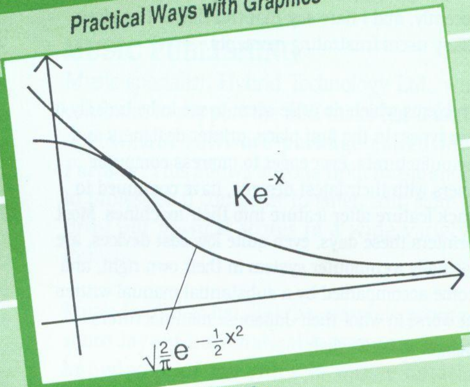


Practical Ways with Graphics

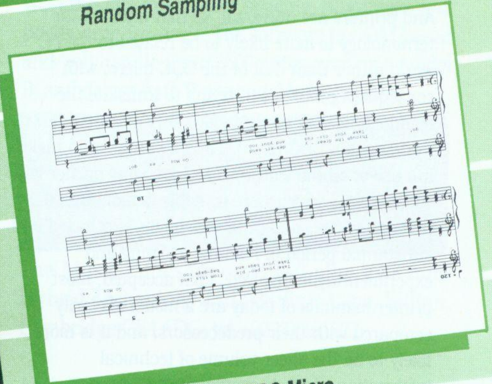


```
Multiple text windows
Multiple text windows
Multiple text windows
Multiple text windows
Multiple text windows
Multiple text windows
Multiple text windows

Another useful
utility from
BEEBUG
```

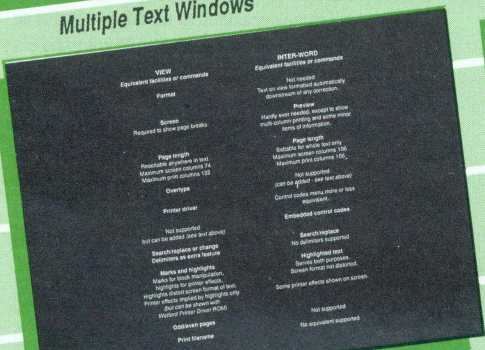
Multiple Text Windows



Random Sampling



News for the BBC Micro



VIEW *Equivalent facilities or commands*	INTER-WORD *Equivalent facilities or commands*
Format	Not needed Text on view formatted automatically downstream of any correction.
Screen Required to show page breaks	**Preview** Hardly ever needed, except to show multi-column printing and some minor items of information.
Page length Resettable anywhere in text Maximum screen columns 74 Maximum print columns 132	**Page length** Settable for whole text only Maximum screen columns 106 Maximum print columns 106
Overtype	Not supported *(can be added - see text above)*
Printer driver	Control codes menu more or less equivalent.
Not supported but can be added *(see text above)*	**Embedded control codes**
Search/replace or change Delimiters as extra feature	**Search/replace** No delimiters supported
Marks and highlights Marks for block manipulation, highlights for printer effects. Highlights distort screen format of text. Printer effects implied by highlights only *(but can be shown with Watford Printer Driver ROM)*	**Highlighted text** Serves both purposes. Screen format not distorted. Some printer effects shown on screen.
Odd/even pages	Not supported
Print filename	No equivalent supported

View and Inter-Word Compared

available on receipt of an A5 SAE), and are strongly advised to upgrade to Basic II. Any second processor fitted to the computer should be turned off before the programs are run.

Where a program requires a certain configuration, this is indicated by symbols at the beginning of the article (as shown opposite). Any other requirements are referred to explicitly in the text of the article.

Program needs at least one bank of sideways RAM.

Program is for Master 128 and Compact only.

Editor's Jottings

PRINTERS

The subject of printers has occupied my mind recently, and I dare say that this device has given many users frustrating moments.

Problems which do arise seem to me to be broadly of two types. In the first place, printer designers and manufacturers, ever eager to impress computer users with their latest designs, have continued to pack feature after feature into their machines. Most printers these days, even quite low cost devices, are virtually a computer system in their own right, and come accompanied by a substantial manual written at worst in what their Japanese manufacturers believe to be English.

And printers are universal, so that much of the terminology is more likely to be related to the PC world rather than that of the BBC micro, with consequent problems in trying to translate the technical information provided into a format that works with a BBC micro. Thus if you want to make full use of all the features of which your printer is capable, and which were so glibly demonstrated perhaps in the shop from which you purchased it, a substantial period of reading, learning and experimenting is involved. That accepted, most printer manuals of today are a model of clarity compared with their predecessors, and it is more likely to be the sheer volume of technical information which is overwhelming.

The other problem which bedevils many printer users is the question of compatibility. Epson established an early lead with the successful marketing of their printers to a point where other manufacturers started claiming 'Epson compatibility' in order to promote sales. Unfortunately, there is no agreement on what the term 'Epson compatibility' actually means, and most printers which claim this standard nearly always include a range of additional features designed to make that model more attractive to the potential user.

We at BEEBUG take Epson compatibility to mean those features and the means of controlling them as described for the Epson FX80. Unless otherwise stated, that is the printer on which any printing capability in magazine programs will have been tested. Of course, not all contributors to the magazine have an FX80, or even a printer which claims to be compatible. In those cases we will either convert the program to Epson FX80 standards or state clearly the printer model used (and we will restrict published material to the more popular printers).

If you use a printer other than an Epson FX80, even if it claims to be compatible, then you must be prepared on some occasions to modify programs so that they will run with your printer. There is no easy way of doing this other than identifying the relevant code sequences in the program (usually VDU sequences) and checking your own printer's manual to see if it uses the same codes for whatever feature is being used. We will always try to include enough information with any program which uses more specialised printing capabilities so that you can modify the program if necessary for your own printer, but you must be prepared to read and use your printer manual to do this.

Of course, all this applies only to a very small percentage of all the programs which we publish, where the more advanced printing capabilities are used. The majority of programs which we print either do not use a printer, or do so only in the simplest style with which all printers can usually cope.

M.W.

BEEBUG News BEEBUG News BEEBUG News BEEBUG News

SOLINET SUPPORT FOR BEEBS

We have been asked to point out that the information "Two into One on a B" published in *Hints & Tips* (BEEBUG Vol.10 No.2) originated from *Solinet*, a support group for BBC Micro owners. Solinet grew out of a user group for Solidisk products, and publishes a bi-monthly magazine on disc which is of interest for all Beeb users, though its emphasis is on Solidisk users. These days any support for the BBC micro is worth publicising. The contact for Solinet is Ron Marshall, Solinet, 41 Westbrook Drive, Rainworth, Mansfield, Notts NG21.

MUSIC GOING CHEAP

Panda Music Discs for the Hybrid Technology Music 5000 are well known, and now prices have been reduced for the remainder of 1991. From now on any three discs may be purchased for just £12 including UK post and packing, additional discs £4 each. The price for single orders remains at £6.

Available discs include 16 volumes for the Music 5000, 3 volumes for the earlier Music 500, and 3 volumes for the System Music System (formerly known as the Island Logic Music System).

For a list of all tracks available send an SAE to Panda Discs, Four Seasons, Tinkers Lane, Brewood, Stafford ST19 9DE.

EDUCATIONAL SOFTWARE

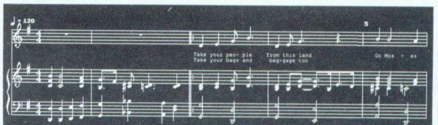
A source of educational software for the BBC and Master series is G.A.Herdman Educational. The latest catalogue just received shows applications covering the Periodic Table, Organic Chemistry, Inorganic Chemistry, Physical Chemistry, Atomic Theory, Electronics and Household Electricity. There is also a range of digital recording thermometers using a temperature sensitive probe and supporting software. Many applications are also available in Archimedes versions, for which system there is additional software.

For more information contact G.A.Herdman Educational, 43 Saint Johns Drive, Clarborough, Retford, Notts DN22 9NN, tel. (0777) 700918.

MUSIC PUBLISHING

Music specialist, Hybrid Technology Ltd., which continues to support the BBC micro has launched a powerful software package called *Music Publisher*. This allows anyone from music student to professional composer to enter, edit, store and print out musical scores in a wide variety of formats.

Music Publisher handles the complete process of score layout automatically using its in-built knowledge of the rules of music notation, working out the correct spacing, alignment and positioning for itself. The software is ROM-based and will handle up to 3500 notes even on a model B, and drives both 9-pin and 24-pin printers.



The Music Publisher costs £60 plus VAT from Hybrid Technology Ltd., 88 Butt Lane, Milton, Cambridge CB4 6DG, tel. (0223) 861522.

ALL FORMATS

Latest dates and venues for *All Formats Computer Fairs* are:

1st Sept	University of Leeds Sports Centre, Calverly Street, Leeds.
7th Sept	Royal Horticultural Hall, Westminster.
14th Sept	National Motorcycle Museum, Solihull.
22nd Sept	City Hall, Candleriggs, Glasgow.
6th Oct	The Brunel Centre, Bristol Old Station (next Temple Meads).

BBC ACORN USER SHOW '91

WEMBLEY CONFERENCE CENTRE - LONDON - 11th to 13th OCTOBER 1991

THE SHOW FOR ALL ACORN USERS

ARCHIMEDES - A3000 - BBC - MASTER - ELECTRON

* *DON`T MISS* *

* *OVER 60 LEADING EXHIBITORS* *
* *LATEST SOFTWARE & HARDWARE*
* *FABULOUS GAMES ARCADE*
* *ACORN COMPUTERS FEATURE STAND*
* *CELEBRITY VISITS*
* *COMPUTER PROBLEM `CLINICS`*
* *SCHOOLS PROJECTS*
* *INFORMATIVE & ENTERTAINING FEATURES*
* *BBC ACORN USER MAGAZINE EDITOR`S OFFICE*
* *MUCH - MUCH - MORE* *

ADVANCE DISCOUNT TICKET APPLICATION ***BBC ACORN USER SHOW `91***

TICKETS AT DOOR - ADULTS £6.00 - UNDER 16`s £4.00 - FAMILY TICKETS £16.00 - UNDER 5`s FREE

ADULTS	£5.00	NAME ....
UNDER 16`s	£3.00	ADDRESS ....
FAMILY TICKETS	£13.00	....
(2 Adults & 2 Children)		....
		.... POSTCODE ....

PLEASE SEND ME .... ADULT .... UNDER 16`s .... FAMILY TICKETS

I ENCLOSE A CHEQUE/POSTAL ORDER FOR £.................. (MADE OUT TO SAFESELL EXHIBITIONS LTD)

SEND THIS FORM TO ; SAFESELL EXHIBITIONS LTD, MARKET HOUSE, CROSS ROAD, TADWORTH, SURREY, KT20 5SR. SHOW INFORMATION WILL BE SENT WITH TICKETS.

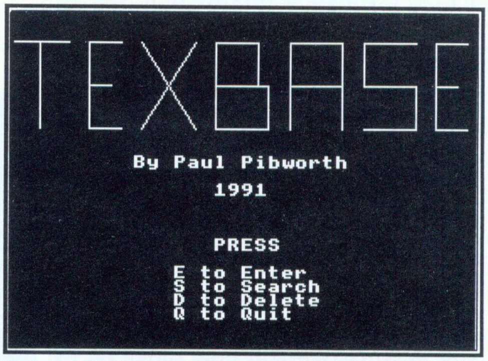
TexBase: A Free Format Information System

by Paul Pibworth

How do you file those handy Beebug Hints and Tips? How do you remember the gist of a particular review? Having typed in that new recipe that uses spinach, turkey, olives, and custard, can you locate and retrieve it from up to 250 other exotic recipes? This program, or rather suite of programs assumes that you already have some familiarity with both databases and word processors. However, can your information system (word processor) store and retrieve plain text such as quotations, reviews, recipes or examination questions, similar in style to a data base; or put another way, can your database handle free format pieces of text?

If these are your requirements then *TexBase* is a suite of programs that might just fit your needs. The system has been written for a BBC model B, and makes use of a WYSIWYG screen in mode 3. This means that the memory space is limited, making it necessary to work within certain constraints. The main limitation is in the use of separate programs to handle different processes. In this, the first part, we will create a boot file, a menu system (program *TMenu*, listing 1), and the first of the main programs which can be called from the menu. This is the program for entering the text, and is called *Enter* (listing 2). Separate articles will then deal with searching and deleting.

The basis of the system is one large random access file in which the entries are stored. Each entry is called a page, and can be up to 50 lines of text. I will describe the format of this data file in the next article. Before we start, however, we need to create a !BOOT file, and also a temporary data file for testing purposes. Then you can enter listings 1 and 2.



```
TEXBASE

By Paul Pibworth
1991

PRESS

E to Enter
S to Search
D to Delete
Q to Quit
```

Initial menu display (program TMenu)

For the !BOOT file, take a clean disc, and type:

```
*BUILD !BOOT <Return>
```

Against the 1 type   :

```
CHAIN"TMENU"  <Return>
```

and then press Escape. Now type:

```
*OPT4,3 <Return>
```

For the data file enter the following one-liner at the keyboard. Do not press Return until the very end.

```
   Z%=OPENOUT"text":FOR I%=1 TO 8212:
BPUT#Z%,0:NEXT:CLOSE#Z%
```

LISTING 1: THE MENU

The purpose of TMenu is to set some of the conditions. One of the conditions is that (on a model B) the programs must run with PAGE set to &1400 in order to recover some memory space. When run, TMenu provides a simple title screen, and gives four choices, *Enter*, *Search*, *Delete*, and *Quit*.

Lines 1030-1070 modify the use of the function keys, and cursor keys. For example, *FX225,239 sets the function keys to return ASCII values of 239 and upwards when pressed alone, and *FX226,140 sets them to return values of 140 upwards when used in conjunction with Shift. You are referred to the User Guide for further information (see also this month's article in the *First Course* series). Lines 1080-1130 are used to reprogram some of the displayed characters, (240-245) which are used to give an on-screen indication of embedded printer codes.

The program also resets some of the *FX modifications if Quit is chosen. At this stage, if you run TMenu, do not select any option other than 'Quit' as the programs for the various options are missing. We will now move on to the first of these for entering text.

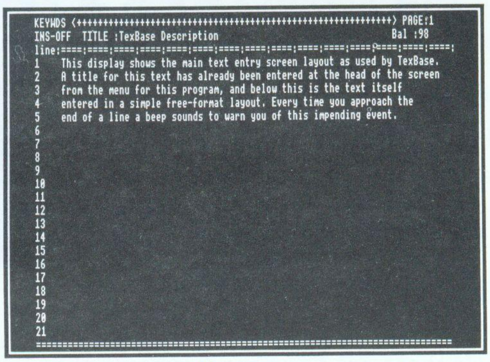
LISTING 2: ENTER

This is the program for entering text. It is well structured, and quite easy to follow, apart from the multi-statement lines (to save on memory). Note that in some cases, lines must be split if there is a condition introduced into the statement ("IF..."). The main body of the program is very short (lines 140-190), and consists of a REPEAT-UNTIL loop. Key presses are detected, and action is taken from within this loop.

ENTERING TEXT

In essence, the Enter program is a simple word processor, driven from a very brief menu. The brevity is due to the memory limitation. However, this does not make it hard to follow. It gives five choices:

1. INPUT
2. TITLE
3. SAVE
4. RESET
5. EXIT



```
KEYWDS <+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++> PAGE:1
INS-OFF  TITLE :TexBase Description                              Bal :98
line:====:====:====:====:====:====:====:====:====:====:====:====:====:====:====:
1    This display shows the main text entry screen layout as used by TexBase.
2    A title for this text has already been entered at the head of the screen
3    from the menu for this program, and below this is the text itself
4    entered in a simple free-format layout. Every time you approach the
5    end of a line a beep sounds to warn you of this impending event.
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
================================================================================
```

Text entry (from Enter program)

INPUT takes you to the screen, for you to enter text. Pressing Escape at any time will return you to this menu. The input screen is WYSIWYG, and is function key driven. At this stage you may find it more convenient to refer to the key strip (see figure 1). The following facilities are included:

(a) Insert on/off
(b) Printer codes for italic, underline, sub-script, super-script, and bold.
(c) Some rearranging of the text.
(d) Loading an ASCII text file.
(e) Marking keywords.

	f0	f1	f2	f3	f4	f5	f6	f7	f8	f9
CTRL	COPY 1 LINE	DISPLAY 1 LINE	REMOVE 1 LINE	INSERT 1 LINE	LOAD ASC FILE	DELETE 1 LINE	CENTRE TEXT	EDIT	MOVE A BLOCK	DELETE A BLOCK
SHFT						DEL TO RIGHT			KEYWD	RESET KEYWDS
	INS/ OVER	ITAL	UNDER LINE	SUB/ SCRIPT	SUPER/ SCRIPT	BOLD	CANCEL PRNTER CODES	CHANGE CASE		HOME

Figure 1

Most of these will be obvious from the key strip, but a few may need further comments. Sub/super-script act on one character only, i.e. they self cancel, the others require a cancel code, which is then shown on the screen. The block-text commands obviously need parameters. On issuing these commands, questions appear on the ruler, such as "from what line?". The cursor keys are modified using Ctrl and Shift. You can set up to three tab positions using Shift/Tab, and cancel them with Ctrl/Tab.

The header information shows five states. The "Page=", on the right, is the current page, or entry, on which you are working. The "Bal" is the number of available lines of text, still unused in the data file. This is a guide, as will be explained later, but with say Bal=200, you could expect to input another 20 short entries. "Title" is an optional entry which is accessed via the menu, and then displayed in place of the row of hash characters.

Finally, there is the heading "KEYWDS". This is the basis of the whole system. Text cannot be saved until you have chosen at least one keyword. Keywords are words in the text that you will use as the basis of subsequent retrieval. They are chosen and stored separately before the bulk of the text. To chose your keywords, place the cursor under the first letter of each word in the text that you wish to use and press Shift/f8. Only alphabetic characters are recognised, and they are converted to upper case. You can enter as many keywords as you like, space permitting. If the space runs out, the first one in is lost! Try it and see. To cancel all the keywords, just press Shift/f9.

The facilities NOT included are word wrap, right justify and reformat. The use of printer codes also takes up actual space, which is then ignored on printout (see later).

One final facility that must be mentioned is EDIT (Ctrl/f7). This allows a previously saved entry to be modified. The page number must be known, and that page is then displayed. The program will then resave that page from the menu, overwriting the original. It is not possible to extend or enlarge the entry, as the number of lines saved will be the same as that loaded. It is anticipated that changes are cosmetic, such as spellings. In the case of a major review, the page can be saved separately, and then reloaded. This will be become clearer when you type in the Search program, next month.

Returning to the menu, apart from *Save* and *Exit*, there is one other choice, *Reset*. This is an important one. Each time the program runs, it calculates the next available space within the data file. This is not updated following a Save action, unless Reset is pressed. This is not an oversight, it allows you to modify and even extend what has just been saved. On pressing Reset, the program examines the disc, and resets the variables for the next entry.

The article so far has described how to set up and write text into TexBase. I would suggest that until the program has been thoroughly checked, you change the appropriate line in Enter that chains TMenu (line 2620), by deleting this action. Replace it with:

```
 2620 DEFPROCend:OSCLI"FX4,0":OSCLI"FX229
, 0":END
```

At the same time, add:

```
   40 *FX4,1
   50 *FX229,1
```

Press the spacebar in place of Return if an error is signalled, and then LIST if necessary. You can then re-run the program by typing RUN. Pressing 5 (EXIT) from the menu will also leave you with the listing in the computer. You could also omit line 1230 in listing 1 (TMenu), that resets PAGE to &1900 until you are satisfied that all is well.

If you have a Master, omit all references to setting page at &1400 and &1900 (lines 1140 & 1230 in TMenu). The next article will show how the data can be searched. I will also tell you about the structure of the data file ("text"), and explain how to create a working example.

Listing 1

```
   10 REM Program TMenu
   20 REM Version B1.0
   30 REM Author  Paul Pibworth
   40 REM BEEBUG  Aug/Sept 1991
   50 REM Program subject to copyright
   60 :
  100 MODE1:COLOUR 129:CLS:PROCtitle
  110 COLOUR0:VDU28,13,25,32,18
  120 PRINT"     PRESS"'
  130 PRINT" E to Enter"
  140 PRINT" S to Search"
  150 PRINT" D to Delete"
  160 PRINT" Q to Quit"
  170 VDU23,1,0;0;0;0;
  180 REPEAT:R$=CHR$(GET AND 95):UNTIL I
NSTR("ESDQ",R$)>0
  190 IF R$="E" THEN PROCchain("ENTER")
  200 IF R$="D" THEN PROCchain("DELETE")
  210 IF R$="S" THEN PROCchain("SEARCH")
  220 PROCquit
  230 END
  240 :
 1000 DEF PROCchain(A$)
 1010 Z%=OPENUP"text":CLOSE#Z%
 1020 IF Z%=0 PRINT"There is no text fil
e."'"You need to create one first.":ENDP
ROC
 1030 *FX229,1
 1040 *FX225,239
 1050 *FX226,140
 1060 *FX227,14
 1070 *FX4,1
 1080 VDU23,240,8,16,32,65,130,4,8,16
 1090 VDU23,241,0,0,0,0,0,0,255,255
 1100 VDU23,242,0,0,0,0,65,34,20,8
 1110 VDU23,243,8,20,34,65,0,0,0,0
 1120 VDU23,244,60,60,60,60,60,60,60,60
 1130 VDU23,245,0,0,255,255,255,0,0,0
 1140 VDU22,3:PAGE=&1400:CHAIN(A$)
 1150 ENDPROC
 1160 :
 1170 DEF PROCquit
 1180 *FX229,0
 1190 *FX225,1
 1200 *FX226,1
 1210 *FX227,1
 1220 *FX4,0
 1230 VDU22,7:PAGE=&1900
 1240 ENDPROC
 1250 :
 1260 DEF PROCtitle
 1270 MOVE75,900:DRAW200,900
 1280 MOVE135,900:DRAW135,700
 1290 MOVE325,900:DRAW250,900
 1300 DRAW250,700:DRAW325,700
 1310 MOVE300,800:DRAW250,800
 1320 MOVE375,900:DRAW500,700
 1330 MOVE375,700:DRAW500,900
 1340 MOVE550,800:DRAW675,800
 1350 DRAW675,700:DRAW550,700
 1360 DRAW550,900:DRAW650,900:DRAW650,80
0
 1370 MOVE725,700:DRAW725,900:DRAW850,90
0
 1380 DRAW850,700:MOVE725,800:DRAW850,80
0
 1390 MOVE900,700:DRAW1025,700:DRAW1025,
800
 1400 :DRAW900,800:DRAW900,900:DRAW1025,
900
 1410 MOVE1150,900:DRAW1075,900
 1420 DRAW1075,700:DRAW1150,700
 1430 MOVE1075,800:DRAW1125,800
 1440 PRINTTAB(11,12)"By Paul Pibworth"
 1450 PRINTTAB(17,14)"1991"
 1460 ENDPROC
```

Continued on page 54

Chess for the BBC Micro

Bernard Hill takes a look at a new chess package for the BBC micro.

In these times which see most packages being written for IBM compatibles, it is good to see appearing a BBC version of the PC-based product known as "ChessBase".

Product	BBChessBase
Supplier	Peter Tate 99 Woodlands Rd, Woodlands, Doncaster DN6 7JY.
Price	£39.95 inc.

BBChessBase is not quite the same as ChessBase, but for the chess-playing BBC owner it offers very similar features. It is essentially a chess game storage and archiving system which offers the ability to store not only games but also positions, analysis and comments.

When you have created a library disc (an empty 40 track or 80 track disc should be used - and put the kettle on as the process takes a long time!), you are ready to store your games or positions, with room for several hundred on each side of an 80-track disc. In addition to the storage of a complete game, some ancillary information can optionally be kept:

White's name: 13 characters, spaces upper-case letters and digits only (no hyphenated names).

Black's name: ditto

Comment on the game: ditto but now 16 characters. This would give room for a tournament or competition name, e.g. "MILANO 1990".

The result: White win, Black win, Draw.

This information is specific to the game, but in addition it is possible to add to each individual move either an opening name, or a comment, or both. Up to four opening names may be used in one game, and each is up to 16 characters (lower case allowed this time).

The opening names are not shown at any stage on the game screen, but are used solely for indexing purposes (see later).

Comments (lower case allowed here) may be added to each move. A comment consists of up to 4 lines of 38 characters, and in addition any of the standard chess move indicators - !!, !?, ? etc. - may be added to any move.

As you enter the actual moves of the game legal-move checking algorithms become active to prevent silly errors.

I have specified the data held with each game in full in order to emphasise the most impressive aspect of the package: the wide range of search options available on a complete database of games. These include:

1. Searching by player. You can do a global search by part of a name, e.g. "KA" would find all the games of Kasparov and Karpov. Or you can restrict the games to those of a particular colour or opening by the player(s) of your choice.

2. Search by a game comment, e.g. all those from "MILANO 1990".

3. Search by opening name.

4. Search by game result, e.g. all white victories.

All these may be combined, so that a search can be made of the library file of, say, Karpov's Sicilian games from Milano in which black won.

STORAGE

The storage of all the games is in tree-fashion. Thus two 40-move games which have the first 20 moves identical would only occupy 60 moves in the data file. This not only saves on disc space but enables the user to browse through the games in what the program calls a 'Tree Search'. Once selected, the initial chess board is shown on screen, and pressing the cursor right key gives a choice of first moves from the database. Change your selected move by pressing space to step through the options and then use the cursor right key again to select. This process of choice selection will continue until you are in a unique line, when the arrows now simply move forwards or backwards through the game, and at any point you can view a note attached to the current move. In addition you can print the game (complete with notes) in horizontal form (1 e2-e4 g8-f6 2 d2-d4 d7-d5 etc.), or print a diagram on an Epson-compatible printer.

The product is fairly comprehensive and is a example of what is possible on an unexpanded BBC micro with disc drives. Inevitably I have some criticisms, but these are minor, and really only encompass ergonomic points. For instance, I am used to pressing Escape to abort or return up one level of menu, and up/down cursor keys with Return to select from a set of vertically presented options. Yet there is usually on-screen help available once you've discovered roughly how the product works. But putting aside these minor criticisms, the chief drawback must be the price. For the rather high price of £39.95 you get a program disc and two sample data discs together with a manual which is reasonably comprehensive (but not very well organised for getting started).

Yet perhaps this level of pricing is inevitable in view of the minority interest which the product serves: the reduced sales inevitably mean a greater unit price to recompense the programmer's time. After all its big brother for the PC costs nearly £200!

In conclusion if you have a need for a sophisticated storage and retrieval system for your chess games then BBChessBase should fill your requirement admirably. B

Wish something new was happening for your BBC Micro, Master or Electron? Something is!

BBC Public Domain software library has over 20Mb of software available!
Send £1.50 for catalogue and sampler disc to:

BBC PD, 18 Carlton Close, Blackrod, Bolton, BL6 5DL.

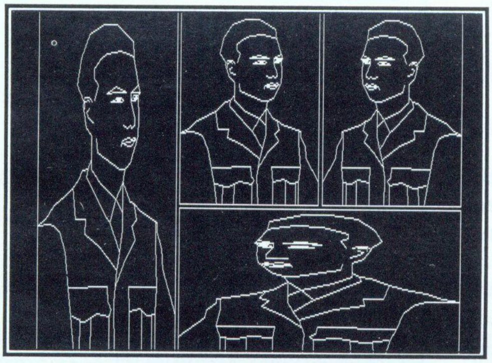
Make cheques payable to;
'A Blundell' or send an A5 s.a.e for more details.
(Please state disc size and format)

Practical Ways With Graphics

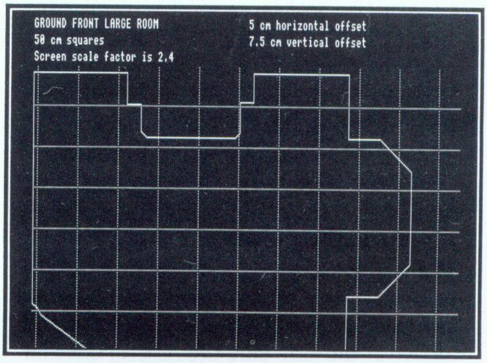
Cliff Blake discusses ways of transferring graphics designs to computer screen based on his own experiences.

USING SCREEN UNITS

Some while ago I photocopied the graphics planning sheet from the early User Guide, but it was a bit too dark to sketch on. I found it better to draw the original on tracing paper, and place it over a grid to work out the co-ordinates. As an example, the program Photo (see listing 1) was obtained by first tracing a photograph and obtaining the x,y values from a grid in this way. The points were enlarged on the tracing in pencil, and the lines gone over with a felt tip pen as they were added to the listing. The blue grid provided with the May issue of the BEEBUG magazine (Vol.10 No.1) is very useful for this purpose.


Program Photo

Using View as a Basic editor, I enter just a pair of co-ordinates on each line with their separating comma. When a position move occurs, a blank line is left. It is then easy to use the editing functions to copy in MOVE at the start of each block, and DRAW or PLOTnn for the following lines. Factors for scaling may also be inserted in this way, and lines joined with colons for multi-statement lines.



```
GROUND FRONT LARGE ROOM                5 cm horizontal offset
50 cm squares                          7.5 cm vertical offset
Screen scale factor is 2.4
```

Program GFL

A BESPOKE GRID

To copy a small sketch to a canvas for painting, a coarse grid is usually employed, and the same technique can be used when copying for the screen. The program Grid (see listing 2) will print a grid on plain paper, or onto tracing paper which may be placed over the original. If copying something like a blazer badge, tracing paper may not be sufficiently transparent, and it will be necessary to first trace the grid onto something like a polythene bag using a ballpoint pen. A printer dump may not preserve the aspect ratio, so the program includes a calibration procedure which will adjust for a particular combination of dump routine and printer.

BY-PASSING SCREEN UNITS

For some applications it may not be necessary to work out screen x,y values directly. In the program GFL (see listing 3), which plots a floor plan, the room is measured in centimetres and these are used as the DATA co-ordinates. A scale

factor is then found by trial and error to make the plan just about fill the screen. No correction is included for aspect ratio as it hasn't been found to be a practical necessity in this case. The program allows the trial fitting of a grid over the plan so that one can determine the required number of carpet tiles of given size. After plotting the plan press:
R to RUN again, P to dump to Printer,
I to CHAIN your own Index, Q to Quit.

A practical point: work to MCC, Maximum Carpet Condition. If a hearth projects into the room take the nearest cm that makes it smaller, and for the alcove at the side take the nearest cm that makes it bigger. If the hearth has rounded corners plot the diagonals across them, and if the alcove has rounded corners plot them square.

You could of course plot bathroom wall dimensions to plan wall tiling, but the principle is the same.

FIND YOUR OWN METHOD

Maybe by now you have considered that perhaps you would photocopy the blazer badge, measure the copy in mm, and apply a scaling factor to get it on the screen. However it is personal preference, subject to trial, as to which method is best employed at any particular time.

So for homework: draft a design with the slogan I BELONG TO BEEBUG in reverse from right to left, use a heat transfer ribbon to print it on tracing paper, and decorate your own tee shirt.

Yes, Jones, you may draw it the right way round, and turn the tracing paper over to work out the co-ordinates. But have you considered negative x values?

A good idea, Brown. Use different scaling factors for BEE, BUG, surround and slogan to adjust different size tracings on screen. You can also move the origin to position each one.

Yes, Smith, you may use drawing software if you are artistic, but then you can't directly trace a BEE and a BUG from a book. However you could try a small tracing paper grid on the book, and a larger polythene grid over the screen.

Listing 1

```
   10 REM Program Photo
   20 REM Version B4.1
   30 REM Author  Cliff Blake
   40 REM BEEBUG  Aug/Sept 1991
   50 REM Program subject to copyright
   60 :
  100 *FX 144,0,1
  110 MODE 0
  120 VDU19,0,7,0,0,0
  130 VDU19,1,0,0,0,0
  140 VDU29,10;10;:PROCplot(.55,.98)
  150 VDU29,1269;10;:PROCplot(-1.115,.42
)
  160 VDU29,433;450;:PROCplot(.55,.55)
  170 VDU29,1269;450;:PROCplot(-.55,.55)
  180 END
  190 :
 1000 DEF PROCplot(sx,sy)
 1010 MOVE0,0:DRAWsx*750,0:DRAWsx*750,sy
*1023
 1020 DRAW0,sy*1023:DRAW0,0
 1030 MOVEsx*200,0:DRAWsx*185,sy*110:DRA
Wsx*170,sy*190
 1040 DRAWsx*150,sy*300:DRAWsx*120,sy*36
0:DRAWsx*90,sy*380
 1050 DRAWsx*60,sy*385:DRAW0,sy*380:DRAW
sx*45,sy*400
 1060 DRAWsx*90,sy*440:DRAWsx*150,sy*470
:DRAWsx*190,sy*495
 1070 DRAWsx*205,sy*400:DRAWsx*260,sy*40
0:DRAWsx*255,sy*355
 1080 DRAWsx*335,sy*320:DRAWsx*430,sy*25
0:MOVEsx*750,sy*120
 1090 DRAWsx*680,sy*255:DRAWsx*675,sy*31
```

```
5:DRAWsx*650,sy*365
 1100 DRAWsx*650,sy*390:DRAWsx*550,sy*43
0:DRAWsx*520,sy*450
 1110 DRAWsx*535,sy*400:DRAWsx*510,sy*38
0:DRAWsx*530,sy*340
 1120 DRAWsx*470,sy*275:DRAWsx*425,sy*20
0:MOVEsx*420,0
 1130 DRAWsx*415,sy*150:DRAWsx*430,sy*25
0:DRAWsx*445,sy*300
 1140 DRAWsx*390,sy*390:DRAWsx*370,sy*41
0:DRAWsx*265,sy*525
 1150 DRAWsx*270,sy*550:DRAWsx*240,sy*54
5:DRAWsx*220,sy*520
 1160 DRAWsx*190,sy*495:MOVEsx*520,sy*45
0:DRAWsx*500,sy*455
 1170 DRAWsx*460,sy*515:DRAWsx*460,sy*42
0:DRAWsx*460,sy*370
 1180 DRAWsx*445,sy*300:MOVEsx*170,sy*19
5:DRAWsx*380,sy*200
 1190 DRAWsx*385,sy*155:DRAWsx*400,sy*13
0:DRAWsx*390,sy*110
 1200 DRAWsx*325,sy*120:DRAWsx*260,sy*95
:DRAWsx*240,sy*95
 1210 DRAWsx*225,sy*103:DRAWsx*190,sy*11
0:MOVEsx*230,0
 1220 DRAWsx*233,sy*100:MOVEsx*290,0:DRA
Wsx*293,sy*103
 1230 MOVEsx*390,0:DRAWsx*390,sy*115:MOV
Esx*640,sy*390
 1240 DRAWsx*640,sy*325:DRAWsx*650,sy*20
0:DRAWsx*640,sy*120
 1250 DRAWsx*640,0:MOVEsx*650,sy*200:DRA
Wsx*495,sy*200
 1260 DRAWsx*500,sy*125:DRAWsx*560,sy*13
5:DRAWsx*595,sy*115
 1270 DRAWsx*605,sy*115:DRAWsx*640,sy*12
0:MOVEsx*500,0
 1280 DRAWsx*500,sy*125:MOVEsx*570,0:DRA
Wsx*570,sy*125
 1290 MOVEsx*625,0:DRAWsx*625,sy*120:MOV
Esx*270,sy*550
 1300 DRAWsx*290,sy*580:DRAWsx*290,sy*55
0:DRAWsx*340,sy*505
 1310 DRAWsx*425,sy*460:DRAWsx*390,sy*39
0:MOVEsx*460,sy*510
 1320 DRAWsx*425,sy*460:DRAWsx*460,sy*42
0:MOVEsx*290,sy*580
 1330 DRAWsx*290;sy*650:DRAWsx*260,sy*67
5:DRAWsx*240,sy*740
 1340 DRAWsx*250,sy*765:DRAWsx*275,sy*76
5:DRAWsx*290,sy*750
 1350 MOVEsx*250,sy*730:DRAWsx*250,sy*75
0:DRAWsx*270,sy*755
 1360 DRAWsx*275,sy*750:MOVEsx*275,sy*73
0:DRAWsx*260,sy*720
 1370 DRAWsx*270,sy*700:DRAWsx*295,sy*69
0:MOVEsx*240,sy*755
 1380 DRAWsx*210,sy*800:DRAWsx*230,sy*88
0:DRAWsx*275,sy*950
 1390 DRAWsx*350,sy*980:DRAWsx*420,sy*98
0:DRAWsx*510,sy*930
 1400 DRAWsx*515,sy*880:DRAWsx*500,sy*89
0:DRAWsx*440,sy*900
 1410 DRAWsx*370,sy*895:DRAWsx*340,sy*88
0:DRAWsx*300,sy*850
 1420 DRAWsx*300,sy*820:DRAWsx*320,sy*80
0:DRAWsx*305,sy*755
 1430 MOVEsx*515,sy*880:DRAWsx*540,sy*80
0:DRAWsx*550,sy*780
 1440 DRAWsx*540,sy*770:DRAWsx*545,sy*70
0:DRAWsx*540,sy*650
 1450 DRAWsx*500,sy*570:DRAWsx*470,sy*55
0:DRAWsx*440,sy*550
 1460 DRAWsx*320,sy*600:DRAWsx*315,sy*64
0:MOVEsx*470,sy*550
 1470 DRAWsx*460,sy*510:MOVEsx*540,sy*80
0:DRAWsx*500,sy*780
 1480 DRAWsx*550,sy*780:MOVEsx*500,sy*78
0:DRAWsx*500,sy*750
 1490 DRAWsx*520,sy*700:DRAWsx*520,sy*68
0:DRAWsx*505,sy*675
 1500 DRAWsx*500,sy*680:DRAWsx*480,sy*67
5:DRAWsx*460,sy*680
 1510 MOVEsx*475,sy*780:DRAWsx*420,sy*79
5:DRAWsx*370,sy*775
 1520 DRAWsx*425,sy*778:DRAWsx*475,sy*78
0:MOVEsx*390,sy*750
 1530 DRAWsx*420,sy*760:DRAWsx*450,sy*76
5:DRAWsx*460,sy*755
 1540 DRAWsx*450,sy*750:DRAWsx*390,sy*75
0:MOVEsx*420,sy*765
 1550 MOVEsx*450,sy*765:PLOT85,sx*435,sy
*753
 1560 MOVEsx*425,sy*753:PLOT85,sx*420,sy
*765
 1570 MOVEsx*500,sy*755:DRAWsx*515,sy*77
0:DRAWsx*535,sy*770
 1580 DRAWsx*540,sy*765:DRAWsx*520,sy*75
```

```
5:DRAWsx*500,sy*755
 1590 MOVEsx*520,sy*770
 1600 MOVEsx*535,sy*770:PLOT85,sx*530,sy
*760
 1610 MOVEsx*520,sy*755:PLOT85,sx*515,sy
*765
 1620 MOVEsx*450,sy*620:DRAWsx*455,sy*63
5:DRAWsx*490,sy*645
 1630 DRAWsx*505,sy*640:DRAWsx*520,sy*64
5:DRAWsx*530,sy*640
 1640 DRAWsx*505,sy*620:DRAWsx*490,sy*62
0:DRAWsx*455,sy*630
 1650 DRAWsx*490,sy*633:DRAWsx*510,sy*63
5:DRAWsx*530,sy*640
 1660 ENDPROC
```

Listing 2

```
   10 REM Program Grid
   20 REM Draws grid for graphics
   30 REM Version B2.0
   40 REM Author  Cliff Blake
   50 REM BEEBUG  Aug/Sept 1991
   60 REM Program subject to copyright
   70 :
  100 MODE 3:PROCinitial
  110 MODE 0:PROCplot
  120 REPEAT
  130 REPEAT:g$=GET$
  140 UNTIL INSTR("PpQq",g$)
  150 IF g$="P" OR g$="p" THEN PROCdump
  160 IF g$="Q" OR g$="q" THEN CLS:END
  170 VDU 7
  180 UNTIL FALSE
  190 END
  200 :
 1000 DEF PROCplot
 1010 VDU29,xo%;yo%;:GCOL 0,1
 1020 PROCaspect:hr=1:vr=1
 1030 IF vert<hori THEN hr=vert/hori
 1040 IF hori<vert THEN vr=hori/vert
 1050 MOVE -xo%*hr,-yo%*vr:DRAW (1279-xo
%)*hr,-yo%*vr
 1060 DRAW (1279-xo%)*hr,(1023-yo%)*vr
 1070 DRAW -xo%*hr,(1023-yo%)*vr:DRAW -x
o%*hr,-yo%*vr
 1080 MOVE -xo%*hr,0:DRAW(1279-xo%)*hr,0
 1090 MOVE 0,-yo%*vr:DRAW0,(1023-yo%)*vr
 1100 FOR x=0 TO (1279-xo%)*hr STEP st%*
hr
 1110 MOVE x,-yo%*vr:PLOT 21,x,(1023-yo%
)*vr
 1120 NEXT x
 1130 FOR x=0 TO -xo%*hr STEP -st%*hr
 1140 MOVE x,-yo%*vr:PLOT 21,x,(1023-yo%
)*vr
 1150 NEXT x
 1160 FOR y=0 TO (1023-yo%)*vr STEP st%*
vr
 1170 MOVE -xo%*hr,y:PLOT 21,(1279-xo%)*
hr,y
 1180 NEXT y
 1190 FOR y=0 TO -yo%*vr STEP -st%*vr
 1200 MOVE -xo%*hr,y:PLOT 21,(1279-xo%)*
hr,y
 1210 NEXT y
 1220 IF xo%>32 AND xo%<1216 THEN PROCxl
abel
 1230 IF yo%>32 AND yo%<960 THEN PROCyla
bel
 1240 VDU29,0;0;
 1250 ENDPROC
 1260 :
 1270 DEF PROCxlabel
 1280 VDU 5
 1290 MOVE -24,(1023-yo%-4)*vr:PRINT;xo%
 1300 VDU 4
 1310 ENDPROC
 1320 :
 1330 DEF PROCylabel
 1340 VDU 5
 1350 MOVE (1279-xo%-56)*hr,12:PRINT;yo%
 1360 VDU 4
 1370 ENDPROC
 1380 :
 1390 DEF PROCinitial
 1400 PRINT TAB(26,0)"GRID FOR GRAPHICS
PLANNING"
 1410 PRINT TAB(10,2)"1  Initial Correct
ion for Aspect Ratio:"
 1420 PRINT TAB(13,4)"Put your own print
er dump routine in PROCdump."
 1430 PRINT TAB(13,5)"Give the value 170
 to hori & vert in PROCaspect."
 1440 PRINT TAB(13,6)"RUN the program."
 1450 PRINT TAB(13,7)"Select: Grid inter
val=64, X value=1023, Y value=0."
 1460 PRINT TAB(13,8)"Dump the grid to p
```

Continued on page 52

Multiple Windows

We present a utility by Stephen Todd, first published in Vol.4 No.2, which allows up to ten different windows to be created and accessed on the screen with ease.

A useful feature of the BBC micro is its ability to split the screen into windows in any mode. The routine listed here allows you to select, by number, up to ten predefined text windows independently of each other, without redefining each time you switch from one to another. Each window can have any size and position, separated from one another or overlapping. The windows can be accessed from within programs or directly from the keyboard. On selection, a window can be either cleared or left uncleared. Lastly, the foreground and background colours of each window can be selected independently. The colours chosen will be used whenever that window is selected.

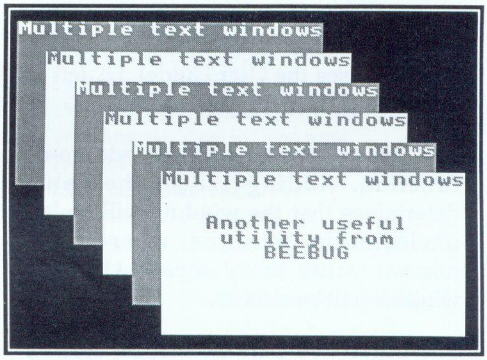
Type in listing 1 and save it with a name other than *Wind* before running it. When run, it will assemble and save a machine code program called *Wind*, which uses pages &900 and &A00. The routine can now be loaded and run with:

```
*Load Wind
C%=1:CALL &AC9
```

Once the routine has been run, the windows will be available, and pressing Break or Ctrl-Break will not affect the window definitions. If it becomes necessary to remove the window facility this can be done by typing *FX247 and pressing Break once.

The routine incorporates default definitions for all ten windows, as follows (windows 1-5 for 40 column modes and windows 6-10 for 80 column):

windows 1 & 6	top left quarter
windows 2 & 7	top right quarter
windows 3 & 8	bottom left quarter
windows 4 & 9	bottom right quarter
windows 5 & 10	centre of screen



YOUR OWN WINDOWS

The selection and definition of windows is done by way of a seldom used VDU code. Code 6 is normally used to enable the VDU drivers. It is therefore redundant in most programs and can be used to call up the window routine. Programs which do use it will still work since the code is still sent to the VDU drivers when selecting and defining windows.

Once the routine has been called, a window can be selected by issuing VDU6,n where n is the window number required. An illegal number will give a "Bad window" error. Since both numbers are in the control code range (ASCII 0-31), a window can be called from the keyboard with just two keypresses: Ctrl-F followed by one of Ctrl-A to Ctrl-J.

Redefining windows is a little more involved. The command for this uses VDU6 again, but this time expects more than just the window number. The full command is VDU6,n+10,a,b,c,d,bc,fc. As you can see, 10 is added to the window number to distinguish between calling and defining commands. The parameters a,b,c,d define the window size and position, and are exactly the same as the parameters used by VDU28 to define text windows (see the User Guide), i.e. left x, bottom y, right x, top y co-ordinates.

The first parameter 'a' has an additional function. Adding 128 to the value determines that the window will be left uncleared on selection, whereas the normal value of 'a' means that the window will be cleared.

The other two parameters, bc and fc, define the foreground and background colours, and are exactly the same as for the COLOUR command. The window will only be completely filled with the background colour if it is set to be cleared on selection.

After the windows have been defined, you may wish to recall the previous definitions. This can be done with VDU6,0 which will print out the definitions one line at a time. This enables you to use cursor/copy editing to restore the definitions exactly as if you were typing them in afresh.

```
   10 REM        >Window
   20 REM Program Multiple windows
   30 REM Version B 0.3
   40 REM Author  Stephen Todd
   50 REM BEEBUG  Aug/Sep 1991
   60 REM Program subject to copyright
   70 :
  100 IF ?&287=&4C ?&287=0:CLS:PRINT''"P
ress break and re-run":END
  110 PROCassem
  120 *Save Wind 900 B00
  130 MODE 4
  140 VDU6,1:FOR I%=0 TO 21
  150 PRINT" WINDOW 1  ";:NEXT
  160 VDU6,2:FOR I%=0 TO 23
  170 PRINT"WINDOW 2  ";:NEXT
  180 VDU6,3:FOR I%=0 TO 23
  190 PRINT"WINDOW 3  ";:NEXT
  200 VDU6,4:FOR I%=0 TO 21
  210 PRINT" WINDOW 4  ";:NEXT
  220 REPEAT UNTIL GET=32
  230 VDU6,15,5,20,15,10,131,4
  240 MODE2
  250 VDU6,5:FOR I%=0 TO 13:PRINT"BEEBUG
 ";:NEXT:PRINT"windows";
  260 REPEAT UNTILGET=32:MODE 7
  270 END
  280 :
 1000 DEFPROCassem
 1010 FOR L%=0 TO 3 STEP 3
 1020 P%=&900
 1030 [OPT L%
 1040 .pchar EQUB &20
 1050 .oldvec  EQUW 0:EQUB &60
 1060 .flag EQUB 0
 1070 .screen EQUB 0
 1080 EQUD &00130C00:EQUW &180
 1090 EQUD &00270C14:EQUW &180
 1100 EQUD &0D131800:EQUW &180
 1110 EQUD &0D271814:EQUW &180
 1120 EQUD &06211206:EQUW &180
 1130 EQUD &00270C00:EQUW &180
 1140 EQUD &004F0C28:EQUW &180
 1150 EQUD &0D271800:EQUW &180
 1160 EQUD &0D4F1828:EQUW &180
 1170 EQUD &0643140C:EQUW &180
 1180 .temp EQUW 0
 1190 .temp1 EQUW 0:EQUB 0
 1200 :
 1210 .printvalues
 1220 pha:txa:pha:tya:pha:lda #0
 1230 sta flag:lda #&1F:sta temp
 1240 lda #&99:sta temp+1
1250 ldx #0:ldy #1:sty temp1+1
1260 .loop1
1270 stx temp1:txa:clc:adc #1
1280 cmp #10:beq notab
1290 pha:lda #32:jsr pchar:pla
1300 .notab
1310 jsr printnum:lda #32:jsr pchar
1320 jsr pchar:jsr pchar:lda #ASC"6"
1330 jsr pchar:lda #ASC",":jsr pchar
1340 lda #6:sta temp1+2
1350 lda #32:jsr pchar:lda temp1
1360 clc:adc #11:jsr printnum
1370 .loop2
1380 lda #ASC",":jsr pchar:lda #32
1390 jsr pchar:ldy temp1+1
1400 lda screen,Y:cmp #11
1410 bcs notgreater:pha:lda #ASC"0"
1420 jsr pchar:pla
1430 .notgreater
1440 jsr printnum:inc temp1+1
1450 dec temp1+2:bne loop2:jsr &FFE7
1460 ldx temp1:inx:cpx #10:bne loop1
1470 pla:tay:pla:tax:pla:rts
1480 :
1490 .vec
1500 pha:lda flag:cmp #1:beq number
1510 cmp #0:bne definingwindow
1520 pla:cmp #6:beq checkdefine
1530 jmp (oldvec)
1540 :
1550 .number
1560 pla:bne notinfo:jmp printvalues
1570 .notinfo
1580 .checkcode
1590 cmp #11:bcc drawwindow
1600 cmp #21:bcs badwindow
1610 pha:sec:sbc #10
1620 jsr calcoffset:inc flag:pla:rts
1630 :
1640 .badwindow
1650 lda #0:sta flag:brk
1660 EQUB &FF:EQUS "Bad window":brk
1670 :
1680 .definingwindow
1690 pla:pha:sta temp:txa:pha
1700 ldx screen:lda temp:sta screen,X
1710 inc screen
1720 pla:tax:inc flag:lda flag
1730 cmp #8:beq enddefine:pla:rts
1740 .enddefine
1750 lda #0:sta flag:pla:rts
1760 :
1770 .checkdefine
1780 pha:txa:pha:tya:pha
1790 lda #&DA:ldy #&FF:ldx #0:jsr &FFF4
1800 txa:bne ctrlcode:inc flag
1810 .ctrlcode
1820 pla:tay:pla:tax:pla:jmp (oldvec)
1830 :
1840 .drawwindow
1850 pha:lda #0:sta flag:pla:pha
1860 jsr calcoffset
1870 txa:pha:tya:pha:ldx screen
1880 lda #28:jsr pchar:ldy #4
1890 lda screen,X:php
1900 .writewindow
1910 lda screen,X:and #&7F:jsr pchar
1920 inx:dey:bne writewindow
1930 lda #17:jsr pchar:lda screen,X
1940 jsr pchar:inx
1950 lda #17:jsr pchar:lda screen,X
1960 jsr pchar:plp:bmi noCLS
1970 lda #12:jsr pchar
1980 .noCLS pla:tay:pla:tax:pla:rts
1990 :
2000 .calcoffset
2010 sta temp:txa:pha:ldx temp:lda #1
2020 .count
2030 dex:beq endcount:clc
2040 adc #6:bne count
2050 .endcount
2060 sta screen:pla:tax:rts
2070 :
2080 .printnum
2090 sta &2A:lda #0:sta &2B:jmp (temp)
2100 :
2110 .init
2120 bcc over:lda &20E:sta oldvec
2130 lda &20F:sta oldvec+1
2140 lda #vec MOD 256:sta &20E
2150 lda #vec DIV 256:sta &20F
2160 .over
2170 rts
2180 .e
2190 ]:NEXT
2200 P%=&287:[OPT 0:jmp init:]
2210 C%=1:CALLinit
2220 ENDPROC
```

Wordwise User's Notebook

This month Chris Robbins describes some useful segment programs for users of Wordwise Plus, and a few alternative hints for Wordwise.

Wordwise (WW) and Wordwise Plus (WW+) are perhaps the most widely used of all word processing packages for the BBC model B. What might be termed the Mark II version of Wordwise, Wordwise Plus, although compatible with the original, contains many enhancements, not the least of which is a feature that, in my experience, many of its users are completely unaware of - a built in programming language.

It's rather like Basic and just as easy to use. Oriented towards the display and manipulation of text, it provides a means of tailoring WW+ to individual user needs and preferences, and generating an infinite variety of special purpose applications and text handling programs.

PREVIEWING TEXT

Take for instance the previewing of text to assess its final printed appearance. In both WW and WW+ this is achieved by selecting option 7 from the main menu. But this 'regular' way of doing things does have its limitations. For one thing, halting the text as it scrolls up the screen means holding down the Ctrl and Shift keys simultaneously, although finger fatigue can be reduced by means of the space bar, which acts as a scroll 'toggle'. However, I think the following approach gives a slightly better preview capability:

In one of the WW+ segments the following short code sequence should be placed:

```
VDU14
VDU19,0,7;0;
VDU19,1,0;0;
```

and at the very top of the main text area the command f1SEGnf2. Where f1 and f2 signify pressing the f1 and f2 red function keys in order to insert the embedded command delimiters |G and |W respectively, and n is replaced by the segment number in which the VDU code sequence has been placed.

VDU14 initiates paged scrolling, ensuring that scrolling halts automatically once the screen has been filled, whilst the other two VDU commands produce a reverse video display. Selecting option 7 now produces a reverse video preview (in this case black 'ink' and white 'paper'), that is automatically held until the Shift key is pressed. The result is text that looks almost like the real thing and no more finger fatigue!

The idea can be extended to give a more general approach by using variables to assign the ink and paper colours. For instance:

```
P%=7
I%=0
VDU14
VDU19,0,P%;0;
VDU19,1,I%;0;
```

gives the same effect, with the variables I% and P% controlling the ink and paper colours respectively. Rather than having to edit the routine to change colours, they can be set for convenience from menu mode by typing commands such as:

```
:P%=4
:I%=1
```

which in this example produces blue paper and red ink; an interesting effect, though perhaps a little hard on the eyes.

The idea can be made more user friendly by means of prompts to allow even easier selection of paper and ink at preview time. For example:

```
CLS
PRINT "Select paper colour 0-7 ";
P%=VAL(GCK$)
```

```
PRINT
PRINT
PRINT "Select ink colour 0-7";
I%=VAL(GCK$)
CLS
VDU14
VDU19,0,P%;0;
VDU19,1,I%;0;
```

The first CLS clears the screen; GCK$ is a WW+ function that *Gets a Character from the Keyboard*, and the WW+ function VAL (like its Basic equivalent) converts this to the appropriate value.

To keep things simple, no attempt has been made to validate the characters entered by the user. Obviously paper and ink should not have the same colour - invisible writing is extremely difficult to preview. Checks could also be made to ensure that only colours in the range 0-7 are input; an ink value of 9, for instance, would produce a most disconcerting flashing red/cyan. One way to prevent such nuisances, and at the same time give an even friendlier interface, would be to provide menus from which only the 'correct' colours could be selected. Further enhancements could include a pre-select facility to avoid having to make a selection each time text is previewed, a reset function, and a perhaps a representative colour display.

DELETE LINE FUNCTION

Both WW and WW+ provide a mechanism for deleting whole words and blocks of text, but despite its usefulness, there's no simple key combination for removing whole lines at a time, although this feature was later included in Interword by means of Ctrl-L. Line-at-a-time deletion is possible, but it involves several separate keystrokes. There are better ways of achieving the same end, one of which is the following routine:

```
CURSOR AT 39
FKEY3
CURSOR UP
CURSOR AT 39
FKEY3
DELETE MARKED
CURSOR DOWN
CURSOR AT 0
DISPLAY
```

This can be placed in any convenient WW+ segment, and invoked whenever required simply by pressing Shift and the function key associated with that segment, e.g. Shift-f1 if the routine is in segment 1. The routine will then delete the whole of the current line in the currently selected text area. What's more, unlike the WW/WW+ delete block function the cursor is not left 'hanging'; it's parked at the start of the new current line after the previous one is deleted.

Another way of achieving the same end, that can also be used in WW, is to define a suitable function key as follows (see also this month's First Course article):

```
*KEYn|!|M|!#|!/|!|M|!#|!'|!|L
```

where n is the key number. Positioning the cursor anywhere under a line, then pressing the Ctrl/Shift/fn combination will delete the whole line. The key definition may look a little daunting, but essentially, the character sequence assigned to the key produces a set of WW edit operations e.g. |!|M moves the cursor to the end of the line. A full explanation of these can be found in the WW manual.

SEARCH AND REPLACE

The main menu *Search and Replace* option is extremely useful for finding strings, but it can prove tedious if replacement isn't required. However, the WW+ language provides an alternative in the FIND command. As with other commands it can be used in immediate form from menu mode if preceded by a colon. For example:

```
:FIND "Albert"
```

will search for the occurrence of *Albert* in the currently selected segment or text area. But this too can prove tedious, especially if the string occurs many times; repeated searches mean moving the cursor, switching between edit and menu mode, re-typing the command, and switching back to edit mode many times. All in all, *Search and Replace* might seem the better choice. In fact, the following simple segment routine eliminates all the difficulties, and allows for repetitive searches without the need to leave edit mode:

```
SELECT TEXT
IF B$ <> "" THEN GOTO find-string
CLS
PRINT
PRINT "String to be searched for ?"
PRINT
PRINT ":- ";
B$ = GLK$

.find-string
CURSOR RIGHT
FIND B$
VDU7
DISPLAY
END
```

In this case, SELECT TEXT ensures that the main text area is the one to be searched; CLS simply clears the screen; and the WW+ function GLK$ picks up the user specified string, assigning it to the string variable B$. All the user has to do is type in exactly what they're searching for; no need even to use string quotes. In subsequent use, the string already assigned to B$ is searched for immediately. Simply pressing Shift and the appropriate function key will locate the next occurrence of the string.

The search mechanism can be reset by assigning a null string to B$, thus allowing a new string to be used. For instance from menu mode:

```
:B$=""
```

or alternatively from edit mode by putting the reset instruction in a separate segment, and invoking it with the relevant Shift-fn combination. Further enhancements could include reset to start of text, reset and hold, display current search string, and perhaps a warning message if the string isn't found.

Both WW and WW+ have the *Search and Replace* facility, but it's only available from the main menu in WW+. Useful though it is when developing segment programs, the segment menus don't have this option. But using the WW+ language it's easily implemented. The following is one possibility:

```
CLS
PRINT "String to be searched for ? ";
A$=GLK$
PRINT
PRINT "String to replace it ? ";
B$=GLK$
REPEAT
    REPLACE A$,B$
UNTIL EOT
VDU7
DISPLAY
END
```

This gives a straightforward global *Search and Replace* facility that will operate on any segment (except the one in which it's running) including the main text area. It's a real boon when developing segment programs, and multiple occurrences of the same name, variable, string etc. have to be changed.

One obvious enhancement would be to provide a selective Search/Replace capability as in the following:

```
CLS
PRINT "String to be searched for ? ";
A$=GLK$
PRINT
PRINT "String to replace it ? ";
B$=GLK$
PRINT
```

Continued on page 35

DFS OSWORD Bug

by Alan Wrigley

As we mentioned in the last issue of BEEBUG, there is a serious bug in DFS version 2.45, as supplied in the new Master MOS ROM, with regard to OSWORD calls &7D and &7E. &7D reads the disc's master sequence number, while &7E reads the disc size.

The problem with the new MOS is that all &7D and &7E calls are treated as if they were &7F calls, which is a general read/write function. This means that firstly they will not perform their intended function, and more seriously, if the data block happens to contain the appropriate combination of control bytes, whole chunks of the disc could have random data written to them.

Admittedly this will be a rare occurrence, but it *could* happen, and if you are worried about it the listing given here will overcome the problem by trapping calls to OSWORD &7D and &7E, and routing them to OSWORD &7F with the parameter block set up to perform the required task. This is done by using the read function to extract the information directly from the first two sectors of the disc.

Type in listing 1 and save it as *FixSrc*. Then run it, and it will assemble and save a machine code utility called *WordFix*. To install this in future, simply type **WordFix*. The utility requires pages &900 and &A00 so you should make sure nothing else is using these.

In the next issue, we hope to provide some information which will allow you to modify the DFS code itself for a more permanent solution, and also some news on the other DFS bug which has been highlighted in our Postbag pages, namely the inability to write to a DFS disc after switching from ADFS.

```
   10 REM        >FixSrc
   20 REM Program OSWORD Bug Fix
   30 REM Version B1.0
   40 REM Author  Alan Wrigley
   50 REM BEEBUG  Aug/Sep 1991
   60 REM Program subject to copyright
   70 :
  100 PROCassem
  110 *Save WordFix 900 9A0
  120 END
  130 :
 1000 DEFPROCassem
 1010 FOR pass%=0 TO 3 STEP 3
 1020 P%=&900:[OPT pass%
 1030 LDA &20C:STA oldvec:LDA &20D
 1040 STA oldvec+1:LDA #bugfix MOD 256
 1050 STA &20C:LDA #bugfix DIV 256
 1060 STA &20D:RTS
 1070 .bugfix CMP #&7D:BEQ trapped
 1080 CMP #&7E:BEQ trapped:JMP (oldvec)
 1090 .trapped PHA:STX &80:STY &81
 1100 LDX #block MOD 256
 1110 LDY #block DIV 256
 1120 LDA #6:JSR &FFD1:LDA &A01:SEC
 1130 SBC #48:LDX #block MOD 256
 1140 LDY #block DIV 256
 1150 STA block:LDA #&7F:JSR &FFF1
 1160 LDA block+10:BNE error:PLA:LDY #0
 1170 CMP #&7D:BNE trap7e
 1180 LDA &A04:STA (&80),Y:RTS
 1190 .trap7e LDA #0:STA (&80),Y:INY
 1200 LDA &A07:STA (&80),Y:INY
 1210 LDA &A06:AND #3:STA (&80),Y:RTS
 1220 .error PLA:LDY #10
 1230 .errloop LDA message,Y:JSR &FFEE
 1240 DEY:BNE errloop:RTS
 1250 .message EQUB 13:EQUS "rorre csiD"
 1260 .oldvec EQUW 0
 1270 .block EQUB 0:EQUD &A00:EQUB 3
 1280 EQUB &53:EQUB 0:EQUB 1
 1290 EQUB 33:EQUB 0
 1300 ]:NEXT
 1310 ENDPROC
```

Random Sampling

by Mike Williams

These days there is significant use of computers in the fields of simulation and modelling. In virtually every major sphere of human activity there are computer models which attempt to answer those 'what if' questions. A spreadsheet is one kind of model, usually a financial one. Constructing a set of relationships allows us to examine what happens to the model when one or more parameters is changed.

Weather forecasting is another area which now relies heavily upon computer models. Take the temperature and pressure at a three-dimensional grid of points over a selected part of the earth's surface; apply the laws of physics to see how temperature and pressure change over a period of hours (or days) and you have the basis of a weather forecast for 24 hours ahead. Such forecasts are less than 100% accurate, sometimes disastrously so, not because we do not understand correctly the underlying laws of physics, but because of the difficulty of obtaining sufficiently accurate data to start with, and the coarseness of the grid of points which we are forced to use coupled even today by the limited power of even the fastest mainframe computers.

Clearly many kinds of computer model lack total preciseness, but nevertheless have an increasingly useful role to play. This can arise because using a computer model, perhaps averaged over many 'runs' of that model, can still lead to sufficiently realistic results, and also because the building of a model greatly increases our understanding of the underlying relationships.

One area of modelling or simulation relies on the use of stochastic processes, i.e. on random sampling. For example, many activities in life revolve all too often around queues. Take an organisation which undertakes repairs. It has resources in the form of skilled personnel, and equipment. But people are subject to unexpected absence (through illness etc.). Repair equipment itself is liable to failure. And the items to be repaired don't arrive at regular intervals. It is undesirable to invest in people and resources so that both are often under-used; equally it is undesirable to have items waiting a long time for repair. So how can you investigate such a situation. The answer is to build a computer simulation model using random sampling, and that's the subject which I want to cover in this and at least the next workshop.

This month I shall be concentrating on random sampling. Now some of you might be saying "what is there to learn? Surely Basic has a random number generator". Of course this is true, but random numbers used in this way assume that every number in the given range is equally likely. But this is not the only distribution, maybe not even the most likely. One commonly found distribution is the *normal* distribution, where frequency is highest around some central value and tails off on either side. Heights and weights of people usually fall into this category - so do many events. How do we get a random sample from a normal distribution?

That's the kind of thing we are going to look at, with supporting programs. But first let us take a look at random numbers themselves. Basic, like virtually all computer-based systems, generates *pseudo* random numbers. These numbers will obey all the tests which can be devised for randomness, but fail in one significant, and sometimes useful respect: from a given starting point, a pseudo random number sequence is always reproducible. A real random number sequence cannot ever be reproduced again, other than by chance. That is the essence of its randomness.

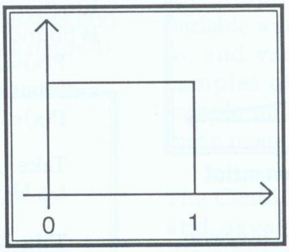


Figure 1. A uniform U (0,1) distribution

What sort of tests can be applied to determine randomness? Well, in a given range every value should occur equally often as any other value; that much is obvious. However, the sequence:

2 2 2 2 3 3 3 3 4 4 4 4 5 5 5 5

meets that test for random numbers in the range 2 to 5, yet the sequence is hardly random. Another test divides the sequence of values into pairs and counts the numbers of step-ups and step-downs, which should be nearly equal. The sequence above clearly fails on that account. More complex forms of the above can also be used with sequences of three and four numbers.

Yet again, it is undesirable for a sequence to repeat too quickly. With pseudo random numbers, once a value is repeated, the sequence following it will be exactly the same as occurred previously. Within a given range of values, the greater the number of values that occur before repetition the better. Various techniques exist for random number generation, and any algorithm such as that used by BBC Basic will have been thoroughly tested to ensure that it meets all reasonable criteria for randomness.

In theory, and we shall be looking at some mathematics along the way, a random number generator, like the RND function, is a method for generating pseudo random numbers from the distribution U(0,1) - see figure 1. What we now have to look at is how sampling from this distribution can be used to generate samples from another distribution, and in particular from a normal distribution.One such method is termed *Analytical Inversion*.

ANALYTICAL INVERSION

Sample U from a U(0,1) distribution. Then set U=F(X). By inverting this function we get $X=F^{-1}(U)$, and X is then a random value from the density function whose distribution is F(x).

The method is limited by the frequent impossibility of finding the distribution function explicitly. For example, given a normal density function:

$$P(x) = \frac{1}{\sqrt{2\Pi}} e^{-\frac{1}{2}x^2}$$

then the probability function (given by integrating the area under the normal distribution curve) is:

$$F(x) = \frac{1}{\sqrt{2\Pi}}\int_{-\infty}^{+\infty} e^{-\frac{1}{2}x^2}dx$$

which cannot be integrated. In many cases, even if the probability function can be found, it is impossible to invert it.

NEGATIVE EXPONENTIAL DISTRIBUTION

To make things clear let's look at an example using the negative exponential distribution (see figure 2) where:

$$P(x) = \alpha e^{-\alpha x}$$

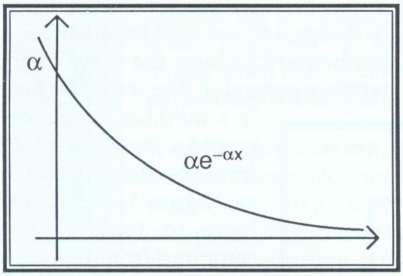


Figure 2. Negative exponential distribution

Using this gives a probability function

$$F(x) = \int_0^x \alpha e^{-\alpha x}dx = [-e^{-\alpha x}]_0^x = 1 - e^{-\alpha x}$$

So using the method of analytical inversion we have:

$$U = 1 - e^{-\alpha x}$$
$$e^{-\alpha x} = 1 - U$$
$$-\alpha x = \log(1 - U)$$
$$x = -\tfrac{1}{\alpha}\log(1 - U)$$

This can be simplified to:

$$X = -\tfrac{1}{\alpha}\log U$$

Thus if U is a random sample from the distribution U(0,1), then X is a random sample from the negative exponential distribution over the same interval.

Here is a function which uses this to generate randomly distributed values from a negative exponential distribution.

```
DEF FNNegExp(a)=-LOG(RND(1))/a
```

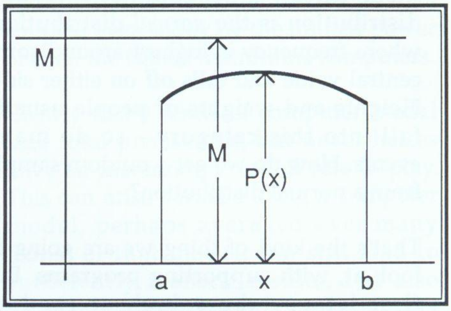


Figure 3. Rejection technique

THE REJECTION TECHNIQUE

Suppose that the density (distribution) function from which we wish to sample is bounded in the sense that:

P(x)=0 when x<a or x>b

and that:

P(x)<=M for some M (see figure 3).

Take U_1=U(0,1)
i.e. U_1 is from a U(0,1) distribution.

Take $X=a+(b-a)U_1$

Now take U_2=U(0,1) and if U_2<=P(X)/M then accept X as a sample value from P(x), otherwise reject it and repeat the whole process, i.e. sample U_1 and U_2 again. It can be shown that the proportion of 'successful' samples of x is 1/M(b-a). If M(b-a) is large then the process will be inefficient.

To generalise, suppose we want to sample from P(x). Suppose that S(x) is a density function from which we *can* sample. Choose a constant K so that K*S(x)>=P(x) for all x. Proceed as follows:

Sample X from distribution S(x)
Sample U = U(0,1)

If U<=P(X)/K(S(X) then accept X,
else reject X and try again.

REJECTION METHOD FOR NORMAL DISTRIBUTION

The normal distribution, an N(0,1) distribution, can be expressed as the density function:

$$f(x) = \frac{1}{\sqrt{2\Pi}} e^{-\frac{1}{2}x^2} \quad -\infty<x<\infty$$

or, in terms of mean m and standard deviation s:

$$f(x) \quad \frac{1}{\sqrt{2\Pi}\,\sigma} e^{-\frac{1}{2\sigma^2}(x-m)^2}$$

We will sample from:

$$f(x)=\sqrt{\frac{2}{\Pi}}\, e^{-\frac{1}{2}x^2} \quad 0\le x<\infty$$

and then attach a random plus or minus sign to this value.

Choose $s(x)=Ke^{-x}=Kg(x)$ and select a value of K so that the negative exponential g(x) just touches the normal curve (see figure 4).

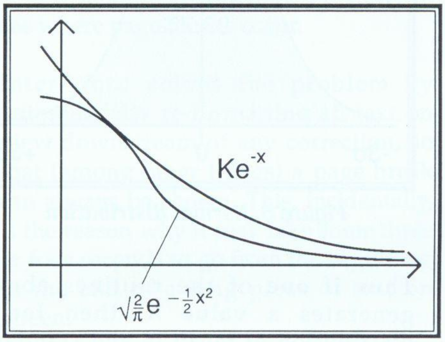


Figure 4. Rejection technique for normal

Now: $$\frac{Kg(x)}{f(x)}=K\sqrt{\frac{\Pi}{2}}\,e^{\frac{1}{2}x^2-x}$$

$$=K\sqrt{\frac{\Pi}{2}}\,e^{\frac{1}{2}(x^2-2x+1)}e^{-\frac{1}{2}}$$

$$=K\sqrt{\frac{\Pi}{2e}}\,e^{\frac{1}{2}(x-1)^2}$$

which has a minimum value of $K\sqrt{\frac{\Pi}{2e}}$ when x=1. Choose $K=\sqrt{\frac{2e}{\Pi}}$ as least possible value of K for rejection technique. Thus the method becomes:

Sample from density function e^{-x} to get X
Sample U from U(0,1)
If U<=f(X)/Kg(X) then accept X
I.e. accept X if $U\le e^{-\frac{1}{2}(x-1)^2}$

We can code this technique as follows:

```
1000 DEF FNNormal1
1010 LOCAL U,X
1020 REPEAT
1030 X=FNNegExp(1):U=RND(1)
1030 UNTIL U<=EXP(-0.5(X-1)^2)
1040 =U
```

THE CENTRAL LIMIT THEOREM

The Central Limit Theorem provides a relatively simple but approximate method for sampling from N(0,1). Suppose a variable x has some distribution of mean m, and variance σ. It is known that if samples of size n are taken, then the sample mean X is a random variable of mean m and variance $\frac{\sigma^2}{n}$

The Central Limit Theorem goes further and says that when n is large, the distribution of X is asymptotically normal. Thus if we sample n U(0,1) variables U_1, ... Un then we know that 'approximately' if:

$X=U_1+U_2+ \ldots . +Un$

then X has a mean n and variance $n\sigma^2$

Now mean=n/2 and $\sigma^2=1/12$ giving a variance of n/12. Choosing the value of n determines the speed and accuracy of the result. For example take n=12. Then:

$X=\sum_{1}^{12} U_i$ has mean 6 variance 1

$X=\sum_{1}^{12} U_i - 6$ is approximately N(0,1)

having mean 0 and variance 1 (see figure 5).

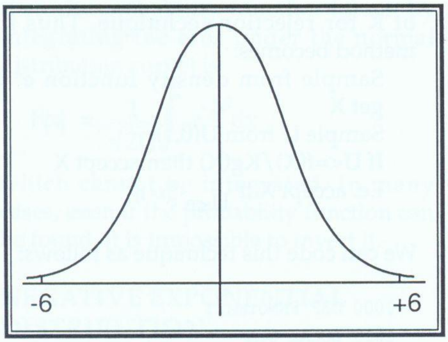


Figure 5. Approximation to normal distribution

Taking n=5

$X=\sum_{1}^{5} U_i$ has mean 2.5 and variance 5/12

$X=\sqrt{\frac{12}{5}}\left(\sum_{1}^{5} U_i - \frac{5}{2}\right)$ is approximately N(0,1)

Now for the coding:

```
1100 DEF FNNormal6
1110 LOCAL I%,X:X=0
1120 FOR I%=1 TO 12
1130 X=X+RND(1)
1140 NEXT I%
1150 =(X-6)

1200 DEF FNNormal5
1210 LOCAL I%,X
1220 FOR I%=1 TO 5
1230 X=X+RND(1)
1240 NEXT I%
1230 =1.54919334*(X-2.5)
```

We thus have three different techniques for obtaining a random sample from a normal distribution. The last two, coded as FNNormal5 and FNNormal6 are to be preferred, with FNNormal5 offering greater speed, and FNNormal6 providing better accuracy. These three techniques are demonstrated visually by a complete working program to be found on this month's magazine disc.

The routines as discussed and coded above only give random samples greater than or equal to zero, i.e. half a normal distribution. To provide a complete normal distribution, we need to add a random plus or minus sign. This can be easily achieved by multiplying the value returned by each function by SGN(RND(1)-0.5).

The other point about these routines is that the result is given in terms of a mean of zero and as a function of the standard deviation. On average 99.7% of all values in a normal distribution lie between plus or minus three standard deviations (see figure 6).

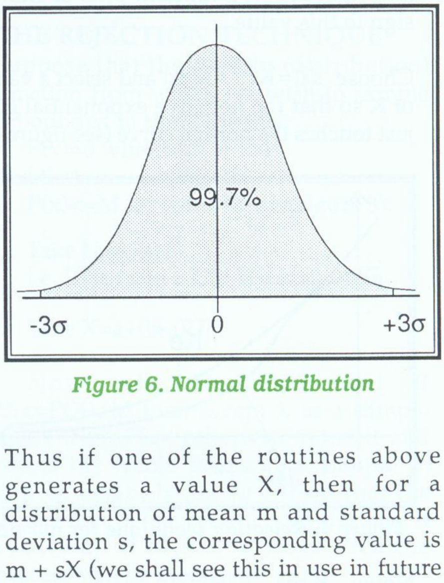


Figure 6. Normal distribution

Thus if one of the routines above generates a value X, then for a distribution of mean m and standard deviation s, the corresponding value is m + sX (we shall see this in use in future versions of these routines).

We now have a number of practical techniques for obtaining random samples from a normal distribution, and we shall be making use of these in the future to explore modelling and queueing systems.

View and Inter-Word Compared

Ruben Hadekel presents a personal viewpoint in comparing two popular word processors.

View and Inter-Word are probably the most widely used word processing programs for BBC B and Master series micros, though Inter-Word's predecessor, Wordwise Plus, is still very much in business (see Wordwise Users' Notes), and retains a large following.

Wordwise Plus is very good, but the need to shuttle between editing and preview modes to be sure of the format is something I find tiresome - it is amazing how often "orphan lines", and particularly paragraph headings as the last line on a page, occur until you take corrective measures.

In that particular respect View is also less than ideal, since you have to go to the Screen mode (equivalent to Preview) to see where page breaks occur.

Inter-Word solves the problem by automatically re-formatting all text on view downstream of any correction, so that (among other things) a page break can always be shown. This, incidentally, is the reason why it may take some three or four seconds to go from the beginning to the end of a long piece of text on screen.

Inter-Word lacks an overtype mode. Such a mode is never essential, and you can operate quite happily without it, but there are circumstances where it is convenient. BBC Acorn User for June 1989 gives a program that adds an overtype mode to Inter-Word. The effect is a bit jerky, but otherwise the program is quite effective. However, the View overtype mode, in conjunction with the Insert character and Split line commands, almost gives you the advantages of overtype and insert modes together.

View uses a printer driver, the default one being for Epson compatible printers (which in practice means most if not all present-day dot matrix printers are catered for). Inter-Word has no printer driver as such, although the Control Codes menu is roughly equivalent. It is, however, more easily accessible, being incorporated within the program. Moreover, a printer driver only caters for a limited range of facilities, while Inter-Word's embedded command facility gives access to all the effects that the printer may have, and the pre-print command is also handy for setting such things as one and a half line spacing, elite typeface etc.

The only way I know of obtaining such effects with View is to write an ad hoc printer driver, sacrificing some standard printer effect to achieve a different one.

Printer drivers can be used to print text on two printers with different control codes. With Inter-Word, you can easily alter the Control Codes menu for another printer, but if you wish to preserve the text unmodified, this can also be done. Create an empty text file, say INTRO2, which holds the Control Codes menu for the second printer. Load this, and then append your text by the LOAD TEXT AT CURSOR command, equivalent to READ in View.

Watford Electronics' Printer Driver ROM adds an embedded command facility to View, together with a few other features present in Inter-Word but not in View. I should make it clear that any references to it are based on advertised features, not on personal experience, which I do not have.

The above-mentioned features are probably the most important points of difference between the two programs. There are strong differences in terminology - for instance, Macro in View is more or less equivalent to Packages (1 to 15) in Inter-Word.

Table 1 lists the main points of difference between View and Inter-Word.

VIEW *Equivalent facilities or commands*	**INTER-WORD** *Equivalent facilities or commands*
Format	Not needed Text on view formatted automatically downstream of any correction.
Screen Required to show page breaks	**Preview** Hardly ever needed, except to show multi-column printing and some minor items of information.
Page length Resettable anywhere in text Maximum screen columns 74 Maximum print columns 132	**Page length** Settable for whole text only Maximum screen columns 106 Maximum print columns 106
Overtype	Not supported *(can be added - see text above)*
Printer driver	Control codes menu more or less equivalent.
Not supported but can be added *(see text above)*	**Embedded control codes**
Search/replace or change **Delimiters as extra feature**	**Search/replace** No delimiters supported
Marks and highlights Marks for block manipulation, highlights for printer effects. Highlights distort screen format of text. Printer effects implied by highlights only *(but can be shown with Watford Printer Driver ROM)*	**Highlighted text** Serves both purposes. Screen format not distorted. Some printer effects shown on screen.
Odd/even pages	Not supported
Print filename	No equivalent supported

VIEW *Equivalent facilities or commands*	INTER-WORD *Equivalent facilities or commands*
Not supported	**Pound sign definition** *(both £ and # can be printed)*
Not supported *(but available with Watford Printer Driver ROM)*	**Pad character**
Not supported *(but facility provided by program in BEEBUG, Vol.7 No.1)*	**Multi-column printing**
Not supported *(but facility provided by program in BEEBUG Vol.10,No.1)* *(Not to be confused with *Spool)*	**Spooling** *(i.e. saving in pure ASCII format - see footnote)*
Editing Basic programs Programming Shift-Break fn Can be used for programming text and commands	Not supported Programming Shift-Break fn Usable for any purpose, including programming compound characters such as +, circumflexed vowels, dead key effects etc.
Microjustification	Not supported
Not supported	**Optional line numbering on screen**
Delete to end of line	Not supported
Delete to character	Not supported
Not supported *(but achievable by Delete to space)*	**Delete word**
No protection against inadvertent overwriting of file.	Safety net REPLACE OLD FILE? (Y/N)

Table 1. Comparison of main features (shown in bold) of View and Interword

As regards ease of use, there is probably little to choose between the two programs. Whichever you are more accustomed to will almost certainly be the easier to use.

Note on Spooling: Saving in "pure ASCII" format is an essential preliminary if the text is to be transcribed for use with some other word processing program, and particularly on some other computer such as a PC compatible. The program DISCopy (from BAKsoft, 20 Leys Avenue, Cambridge CB4 2AW) allows you to format a PC compatible disc, and transcribe to it. A twin disc drive is desirable. See also the article on PC disc formats in this issue of BEEBUG.

SPECIAL

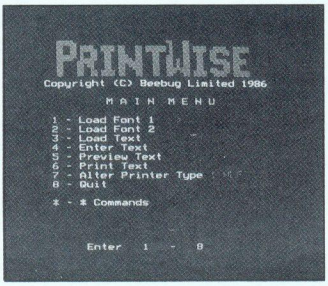
40% Members Discount on

Beebug C & Stand Alone Generator

Beebug C is the much acclaimed C programming language for the BBC Micro and Master 128. Although normally only available on more powerful computers, Beebug C is a full implementation of the Kernighan & Ritchie standard. Features include:

- ★ Runs on BBC B, B+ and M128.
- ★ Comprehensive set of ANSI functions.
- ★ OS functions vdu, osbyte, mode etc.
- ★ Command line interpreter.
- ★ Floating-point maths.
- ★ Linker for multi-source programs.
- ★ Maths functions.
- ★ Expandable run-time library.
- ★ Stand alone generator.
- ★ Full macro handling facilities.
- ★ Supplied on 2 ROMs & library disc

Normal price: ***£79.69*** inc VAT
Offer price: ***£47.81*** inc VAT
Stock code: 0074 40T, 0075 80T
Please add £3.50 carriage.



```
PRINTWISE
Copyright (C) Beebug Limited 1986
        M A I N   M E N U
1 - Load Font 1
2 - Load Font 2
3 - Load Text
4 - Enter Text
5 - Preview Text
6 - Print Text
7 - Alter Printer Type
8 - Quit

* - * Commands

        Enter  1  -  8
```

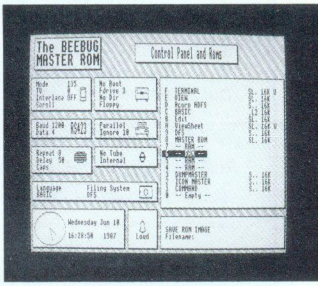
Printwise

Printwise is a low-cost publishing aid allowing you to create professional looking magazines, leaflets, posters etc - the possibilities are endless. Simply take your text file, use embedded commands to specify the font styles you require, and let Printwise do the rest.

- ★ 9 authentic fonts from 4pt to 40pt.
- ★ Font designer for creating new fonts.
- ★ Fonts may be used on the same line.
- ★ Italics, bold, condensed, reversed etc.
- ★ Proportional spacing.
- ★ Subscript and superscript.
- ★ Left, centre, right & full justification.
- ★ Suitable for Epson compatible printers.

Printwise is easy to use and requires no programming skills. It may be used with text files created on Wordwise, View, InterWord, Mini Office and almost any text editor.

Normal price: ***£30.66*** inc VAT
Offer price: ***£18.40*** inc VAT
Stock code: 0085 40T, 0086 80T
Please add £2.50 carriage.



```
The BEEBUG          Control Panel and Roms
MASTER ROM

Mode      135   No Boot        F  TERMINAL      SL. 16K U
TV        0     Fdrive 3       E  VIEW          SL. 16K
Interlace OFF   No Dir         D  Acorn ADFS    S.. 16K
Scroll          Floppy         C  BASIC         .L2 16K
                               B  Edit          SL. 16K
Baud 1200 RS423 Parallel       A  ViewSheet     SL. 16K U
Data 4          Ignore 10      9  DFS           S.. 16K
                               8  MASTER ROM    SL. 16K
Repeat 0        No Tube        7  -- RAM --
Delay  50       Internal       6  -- RAM --
Caps                           5  -- RAM --
                               4  -- RAM --
Language    Filing System      3  DUMPMASTER    S.. 16K
BASIC       DFS                2  ICON MASTER   S.. 16K
                               1  COMMAND       S.. 16K
                               0  -- Empty --

Wednesday Jun 10       Loud    SAVE ROM IMAGE
16:28:50   1987                Filename:
```

Master ROM

The Master ROM is a powerful 32K ROM packed with features to enhance the facilities of the Master 128.

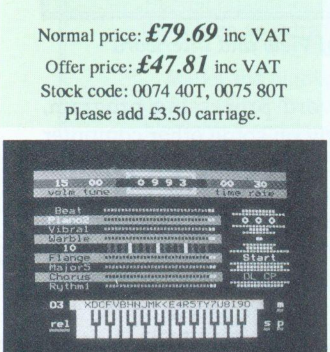
Disc Menu - A single command takes you to a full feature disc menu displaying all the items in the current directory. You can change directories or run, copy, delete, rename selected files.

Control panel - This displays all the computers status settings and the ROMs fitted. The cursor keys may be used to adjust any of the settings, which may be saved for future loading at any time.

Disc commands - A whole range of useful ADFS commands, including: *FIND, *FORMAT, *VERIFY, *BACKUP, *MERGE, *WIPE etc.

Plus: **16K-64K Printer buffer**
Simple 16K-64K RAM disc
Diary and automatic alarm

Normal price: ***£39.84*** inc VAT
Offer price: ***£23.90*** inc VAT
Stock code: 0087 ROM
Please add £2.50 carriage.



Studio 8

Studio 8 is a real-time studio system with digital recorder, rhythm and drum machines which give hours of entertainment.

Studio - Allows playing and recording in real time with keyboard and sequencer.

Editor - A full-screen editor allowing the precise editing of notes.

Envelope editor - allows the definition of up to 16 amplitude and pitch envelopes.

Instrument definer - Create up to 32 instruments by combining pitch and amplitude envelopes, volume & sustain.

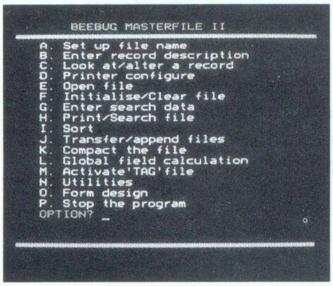
Plus many more features.

Normal price: ***£22.48*** inc VAT
Offer price: ***£13.49*** inc VAT
Stock code: 0009 80T
Please add £2.50 carriage.

Offer extended until 31st August 1991

OFFER

BBC & Master software



```
        BEEBUG MASTERFILE II
  A. Set up file name
  B. Enter record description
  C. Look at/alter a record
  D. Printer configure
  E. Open file
  F. Initialise/Clear file
  G. Enter search data
  H. Print/Search file
  I. Sort
  J. Transfer/append files
  K. Compact the file
  L. Global field calculation
  M. Activate'TAG'file
  N. Utilities
  O. Form design
  P. Stop the program
  OPTION? _
```

Masterfile II

Masterfile II is the best selling general purpose database for the BBC Micro and Master 128. It has many powerful features, but is easy-to-use and ideally suited for home, business or school. It is supplied ready to run and no programming skills are required.

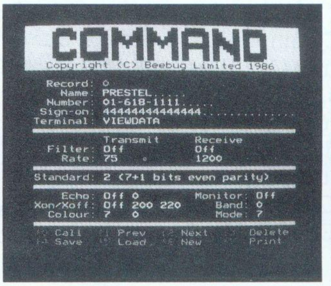
★ Menu driven for ease-of-use.
★ Fast 'tag' sorting.
★ Flexible print layouts.
★ Global calculations.
★ File size limited only by disc capacity.
★ File append options.
★ Search for a particular match.
★ Sorting on any field or fields.
★ Direct fast access to any record.
★ Form designer.
★ Label printing facility.
★ Record insertion.
★ Example database and tag files.

Normal price: ***£22.48*** inc VAT

Offer price: ***£13.50*** inc VAT

Stock code 0024 40T, 0025 80T, 0081 ADFS for Master 128 only

Please add £2.50 carriage.



```
   COMMAND
Copyright (C) Beebug Limited 1986

   Record: 0
     Name: PRESTEL.....
   Number: 01-618-1111.....
  Sign-on: 44444444444444...............
 Terminal: VIEWDATA

           Transmit      Receive
   Filter: Off           Off
     Rate: 75            1200

Standard: 2 (7+1 bits even parity)

     Echo: Off 0        Monitor: Off
 Xon/Xoff: Off 200 220     Band: 0
   Colour: 7   0           Mode: 7

   Call     Prev     Next       Delete
   Save     Load     New        Print
```

Command

The Command ROM is a powerful communications package that may be both menu or command driven. Command may be used with Hayes and other intelligent modems.

Text terminal - Use this to access scrolling text services such as Telecom Gold. XMODEM file transfer allows files to be sent around the world.

Viewdata terminal - A full feature terminal giving access to Prestel and other Viewdata services. Frames may be saved to disc, printed, tagged etc. and a full feature Viewdata editor is provided.

Telephone directory - Set up the name, number and log-on for services you wish to recall at a later date.

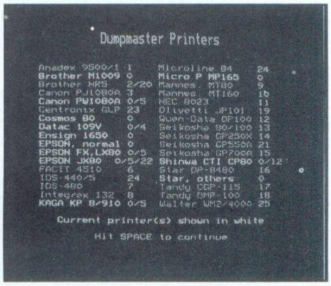
Command driven - A wide range of commands may be used in Basic to create applications for your own individual needs.

Normal price: ***£39.84*** inc VAT

Offer price: ***£23.90*** inc VAT

Stock code: 0073 ROM

Please add £2.50 carriage.



```
            Dumpmaster Printers

Anadex 9500/1  1      Microline 84       24
Brother M1009  0      Micro P MP165      0
Brother HR5    2/20   Mannes. MT80       9
Canon PJ1080A  3      Mannes. MT160      10
Canon PW1080A  0/5    NEC 8023           11
Centronix GLP  23     Olivetti JP101     19
Cosmos 80      0      Quen-Data DP100    12
Datac 109V     0/4    Seikosha 80/100    13
Ensign 1650    0      Seikosha GP250X    14
EPSON, normal  0      Seikosha GP550A    21
EPSON FX,LX80  0/5    Seikosha GP700A    15
EPSON JX80   0/5/22   Shinwa CTI CP80    0/12
FACIT 4510     6      Star DP-8480       16
IDS-440/5      24     Star, others       0
IDS-480        7      Tandy CGP-115      17
Integrex 132   8      Tandy DMP-100      18
KAGA KP 8/910  0/5    Walter WM2/4000    25

    Current printer(s) shown in white

         Hit SPACE to continue
```

Dumpmaster II ROM

Dumpmaster provides fast screen dumps using up to 8 shades from any screen mode. It includes a 'snapshot' facility to dump screens from programs while they are running. A wide range of printers are supported, including Epson compatibles and the Star LC10 colour.

Normal price: ***£31.68*** inc VAT

Offer price: ***£19.00*** inc VAT

Stock code 0053. ROM

Please add £2.50 carriage.

Exmon II

This acclaimed ROM adds over 60 commands to maniplulate and debug machine code programs. Features include: ★ set breakpoints ★ trace ★ set registers ★ full-screen RAM editor ★ single-step ★ disassembler ★ dual screens ★ search ★ line assembler ★ program relocator

Normal price: ***£32.70*** inc VAT

Offer price: ***£19.60*** inc VAT

Stock code: 0004 ROM

Please add £2.50 carriage.

Quickcalc

Quickcalc is an easy-to-use disc based spreadsheet enabling you to use the calculating power of your computer without any need to program. It is ideal for personal accounts, stock control, and general financial planning.

Normal price: ***£18.39*** inc VAT

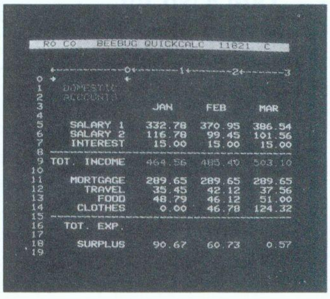
Offer price: ***£11.00*** inc VAT

Stock code: 0029 80T DFS

Please add £2.50 carriage.

Features include:

★ 20 x 50 default spreadsheet.
★ Load, save and print spreadsheet.
★ Replicate into row, column or area.
★ Simple histogram capability.
★ Individual column widths may be altered at any time,
★ Min, max and sum functions.
★ Very easy to use.



```
   RO CO    BEEBUG QUICKCALC   11821   C

   <------------0<-------1<-------2<-------3
 0 <            <
 1   DOMESTIC
 2   ACCOUNTS
 3                   JAN      FEB      MAR
 4
 5    SALARY 1    332.78   370.95   386.54
 6    SALARY 2    116.78    99.45   101.56
 7    INTEREST     15.00    15.00    15.00
 8 ------------------------------------------
 9 TOT. INCOME    464.56   485.40   503.10
10
11    MORTGAGE    289.65   289.65   289.65
12      TRAVEL     35.45    42.12    37.56
13        FOOD     48.79    46.12    51.00
14     CLOTHES      0.00    46.78   124.32
15 ------------------------------------------
16   TOT. EXP.
17
18     SURPLUS     90.67    60.73     0.57
19
```

Using PC Discs on a Beeb: Update

We present updated versions of previously published programs transferring files between PC format discs and a Beeb, and for formatting PC discs.

A good many BBC micro owners also have access to a PC or PC compatible, and need to be able to transfer files between the two machines. This can be achieved if a Beeb can be programmed to read and write PC format discs, and this was described with associated programs by Bernard Hill in BEEBUG Vol.6 No.10 (MS-DOS to BBC) and BEEBUG Vol.7 No.1 (BBC to MS-DOS).

NOTE: The MS-DOS/BBC file transfer programs will only work on systems fitted with the 1770 disc controller chip. This was fitted as standard to the model B+ and Master series. Model B micros with older 8271 chip must be upgraded if these programs are to function correctly (a relatively simple replacement task).

AUTOMATIC FILE TYPE SELECTION

The original programs required the user to set a variable (*asc%*) in the programs to TRUE or FALSE depending on whether ASCII text or data files are to be transferred. The same consideration also applies to the setting of the variable *View%* for transferring View files. This is inconvenient, and the following modifications, supplied by Kenneth Clarke, automate the choice of file type.

MS-DOS TO BBC TRANSFER (Vol.6 No.10)

Delete line 130 and then add the following lines:

```
 305 asc%=FNasc
1880:
1890 DEF FNasc
1900 PRINT''TAB(3)"Transfer as a Text o
r Data file?"''
1910 REPEAT:G=GET AND &DF:UNTIL CHR$G="
T" OR CHR$G="D"
1920 =(CHR$G="T")
```

The further additions listed here also allow the user to choose a printed directory listing of a DOS format disc:

```
1375 IF FNprint VDU2
1430 UNTIL?loc=0ORnf=M:PRINT:VDU3:ENDPR
OC
1930:
1940DEF FNprint
1950 PRINT''TAB(5)"Directory listing on
printer?"''TAB(11)"(Press Y or N)."''
1960 REPEAT:G=GETAND&DF:UNTIL CHR$G="Y"
ORCHR$G="N":CLS
1970 =(CHR$G="Y")
```

BBC TO MS-DOS TRANSFER (Vol.7 No.1)

Delete line number 150 and add the following lines:

```
 525 asc%=FNasc:IF asc% View%=FNView
2770 DEF FNasc
2780 PRINT'TAB(3)"Transfer as a Text or
Data file?"''
2790 REPEAT:G=GET AND &DF:UNTIL CHR$G="
T" OR CHR$G="D"
2800=(CHR$G="T")
2810DEF FNView
2820 PRINT'TAB(8)"Is this a View file?"
''TAB(11)"(Press Y or N)"
2830 REPEAT:G=GET AND &DF:UNTIL CHR$G="
Y" OR CHR$G="N"
2840 =(CHR$G="Y")
```

PC FORMATTING WITH DATE/TIME STAMPING

More recently, in BEEBUG Vol.9 No.8, we published a program by Kate Crennell for formatting PC style discs on a BBC micro. The following modification to that program by R.Poynter adds *Volume Creation Time and Date Stamping* to the original program. This is useful under MS-DOS as when a disc is 'diagnosed' under MS-DOS by the CHKDSK routine, it prints out the Volume Creation Time and Date if present. This was not catered for by the original PC Formatter and is added with these additions.

```
 705 IF INKEY-256=253 AND title$<>"" PR
OCfilestamp : REM Master 128?

1650:
1660 DEF PROCfilestamp
1670 DIM paramblk% 6:osword=&FFF1
1680 X%=paramblk% MOD256:Y%=paramblk% D
IV256
1690 A%=&E:?paramblk%=1:CALL osword : R
EM read CMOS Clock (BCD-format)
1700 FORoffset%=0TO6 : REM convert data
 from BCD to Binary
1710 paramblk%?offset%=paramblk%?offset
% DIV16*10 + paramblk%?offset% MOD16
1720 NEXT
1730 REM insert creation Time and Date
in DOS-format - secs are in multiples of
 2
1740 buf%!22=paramblk%?4*2048+paramblk%
?5*32+paramblk%?6 DIV2
1750 REM DOS stores code for current Ye
ar minus 1980 - e.g. 1991=11
1760 buf%!24=(?paramblk%-80)*512+paramb
lk%?1*32+paramblk%?2
1770 ENDPROC
```

Note that date and time stamping is only feasible on a Master 128 which is fitted with a real-time clock. The call to the new procedure, line 705, checks that the program is running on a Master before attempting to do this.

All three programs referred to above are included in their full, updated form on this month's magazine disc.

Wordwise User's Notebook (continued from page 22)

```
PRINT "Selective replacement Y/N ? ";
S$=GLK$
S%=FALSE
IF S$="Y" OR S$="y" THEN S%=TRUE

REPEAT
   FIND A$
   IF EOT THEN GOTO skip
   IF S%<>TRUE THEN GOTO replace
   VDU7
   DISPLAY
   R$=GCK$
   IF R$<>"Y" AND R$<>"y" THEN GOTO
skip
 .replace
   REPLACE A$,B$
 .skip
   CURSOR RIGHT
UNTIL EOT

VDU7
DISPLAY
END
```

In the above, the initial user responses are picked up using GLK$ which echoes what has been typed in. However, further on, when searching through text, user input is picked up using GCK$ which doesn't echo, and thus avoids corrupting the display.

It's a fairly simple implementation, which, when globally replacing strings, only displays the modified text after all changes have been made. However, if the user had replied "Y" or "y" for selective replacement, whenever the string to be replaced is found, the cursor indicates its position and an audible prompt is given by means of VDU7. The routine waits for user instructions. If "Y" or "y" is typed, the string is replaced, otherwise the next occurrence of the string is searched for.

BEEBUG Function/Procedure Library (5)

by Andrew Rowland

Andrew Rowland adds a varied collection of functions and procedures to our library of useful routines. More contributions for this feature will be welcome. Note (for newer readers) that parts 1 to 4 were published in BEEBUG Vol.9 No.9 to Vol.10 No.2 inclusive.

THE FUNCTION/PROCEDURE LIBRARY (PART 5)

Routine 32: Filesystem

Type:	FUNCTION
Syntax:	FNfilesystem
Purpose:	Returns a string which is the name of the currently selected filing system (see line 24590)
Parameters:	None
Notes:	Uses a LOCAL data pointer
Related:	None

```
  10 oldfs$=FNfilesystem
  20 IF oldfs$<>"TAPE" THEN *TAPE
  ...
1000 OSCLI oldfs$:END
```

Routine 33: ReadFns

Type:	PROCEDURE
Syntax:	PROCreadfns
Purpose:	Reads filenames from current directory
Parameters:	None
Notes:	The array fn$() must be DIMmed to the appropriate size before calling (31 for DFS, 47 for ADFS, 255 for NFS), as must the parameter block and buffer. Returns the number of files in nrfiles%.
Related:	None

```
  10 DIM pblk% 12,buffer% 250,fn$(255)
  20 PROCreadfns
  30 FOR I%=0 TO nrfiles%-1:PRINT fn$(I%
):NEXT
  40 END
```

Routine 34: Wildcard

Type:	FUNCTION	
Syntax:	FNwildcard(wild$,match$)	
Purpose:	Matches strings to an ambiguous specification	
Parameters:	wild$	Ambiguous specification
	match$	String to be compared
Notes:	Wild$ may contain # and/or * as wildcards. Returns TRUE if they match.	
Related:	None	

```
 100 FOR I%=0 TO nrfiles%
 110 IF FNwildcard(spec$,fn$(I%)) THEN
     PROCcopy(fn$(I%),to$)
 120 NEXT
```

Routine 35: Path

Type:	FUNCTION
Syntax:	FNpath
Purpose:	Gives full pathname of current directory
Parameters:	None
Notes:	ADFS only
Related:	Requires FNdir

Routine 36: Qsort

Type:	PROCEDURE	
Syntax:	PROCqsort(entries%)	
Purpose:	Performs Quicksort on array$	
Parameters:	entries%	Highest entry in array (NB array$(0) is used)

Notes:	PROCinit must be called first. Ensure s1% and s2% are sufficient for the longest string length you expect, and that mc% is also large enough. The sort routine may be adapted for numeric arrays, and made faster by using resident integer variables and concatenating lines.
Related:	Requires PROCswop

```
 10 PROCinit
 20 FOR I%=0 TO 11:READ array$(I%):NEXT
 30 PROCqsort(11)
 40 FOR I%=0 TO 11:PRINT array$(I%):NEXT
 50 END
100 DATA first,second,third,fourth,cecil
110 DATA cecil,xanadu,fred,cedric,herbert
120 DATA herbert jones,last
```

Routine 37: Capital

Type:	FUNCTION
Syntax:	FNcaps(A$)
Purpose:	Converts string to capitals
Parameters:	A$ string to be capitalised
Notes:	One liner for speed
Related:	None

Routine 38: Any Base

Type:	FUNCTION	
Syntax:	FNanybase(value$,base%)	
Purpose:	Returns the value (in decimal or hex) as a string in the base specified	
Parameters:	value$	Decimal string, or hex preceded by &.
	base%	Base required (2 to 36)
Notes:	Integers only. Will accept more than base 36, but may use some odd characters!	
Related:	None	

```
10 REPEAT:INPUT "Number, base",A$,B%
20 PRINT FNanybase(A$,B%):UNTIL FALSE
```

Routine 39: Base to Decimal

Type:	FUNCTION	
Syntax:	FNbaseTOdec(value$,base%)	
Purpose:	Converts string in the base specified to a numeric value	
Parameters:	value$	the string to be converted
	base%	the number base used
Notes:	Will convert from base 2 - 36 to an integer value	
Related:	None	

```
10 REPEAT:INPUT "Number, base",A$,B%
20 PRINT FNbaseTOdec(A$,B%):UNTIL FALSE
```

Routine 40: Chip

Type:	FUNCTION	
Syntax:	FNchip("<ROM title>")	
Purpose:	Checks for presence of a ROM	
Parameters:	A$	The ROM title as shown by *ROMS
Notes:	Returns FALSE if not present. Can be used to give warning that a ROM is needed but not fitted, or to check before loading a ROM image into sideways RAM.	
Related:	None	

```
10 IF NOT FNchip("VIEW") THEN
  *SRLOAD Image 8000 4
```

Routine 41: Master

Type:	FUNCTION
Syntax:	FNmaster
Purpose:	Returns TRUE if machine is a Master 128 or Compact
Parameters:	None
Notes:	Use before any Master specific commands. Returns FALSE if Z-Break was used.
Related:	None

```
10 IF FNmaster THEN *SHADOW
```

Routine 42: Vartop

Type:	FUNCTION
Syntax:	FNvartop
Purpose:	Returns top of memory used for declared variables
Parameters:	None
Notes:	Useful for debugging

	programs where memory is tight. Use in or after running the program.
Related:	FNfree

Routine 43: Free

Type:	FUNCTION
Syntax:	FNfree
Purpose:	Returns free memory between top of memory used for declared variables and Basic's stack
Parameters:	None
Notes:	Useful for debugging programs where memory is tight.
Related:	FNvartop

Routine 44: Error

Type:	PROCEDURE
Syntax:	PROCerror(err,err$)
Purpose:	Generates a user error (emulates Archimedes' ERROR)
Parameters:	err user error number err$ user error message
Notes:	Causes an error to be generated which will be handled by the error handler in the usual way.
Related:	None

```
10 PROCerror(255,"Wrong password")
```

```
24500 :
24510 REM Current Filing System
24520 :
24530 DEF FNfilesystem
24540 LOCAL A%,Y%,I%,A$,?&1C,?&1D
24550 A%=USR(&FFDA) AND &FF
24560 RESTORE 24590
24570 FOR I%=0 TO A%:READ A$:NEXT
24580 =A$
24590 DATA NONE,TAPE,TAPE3,ROM,DISC,NET,
TELESOFT,IEEE,ADFS,HOST,VIDEODISC
24600 :
24610 REM Read Filenames
24620 :
24630 DIM pblk% 12,buffer% 250,fn$(255)
24640 :
24650 DEF PROCreadfns
24660 LOCAL A%,X%,Y%,K%,files%,len%
24670 files%=22:REM max for Econet
24680 nrfiles%=0:osgbpb=&FFD1
24690 pblk%!1=buffer%:pblk%!9=0
24700 REPEAT pblk%!5=files%
24710 X%=pblk%:Y%=X% DIV 256
24720 A%=8:CALL osgbpb
24730 K%=files%-pblk%!5
24740 len%=?buffer%:A%=buffer%+1
24750 IF K% FOR X%=nrfiles% TO nrfiles%+
K%-1:A%?len%=13:fn$(X%)=$A%:A%=A%+len%+1
:NEXT
24760 nrfiles%=nrfiles%+K%
24770 UNTIL K%<files%
24780 ENDPROC
```

```
24790 :
24800 REM Wildcard
24810 :
24820 DEF FNwildcard(wild$,match$)
24830 wild$=FNstrip(wild$):match$=FNstri
p(match$)
24840 IF match$<>"" ELSE IF wild$="*" =T
RUE ELSE =wild$=""
24850 IF wild$="" =FALSE
24860 IF ASCwild$=ASC"#" OR (ASCmatch$ A
ND &DF)=(ASCwild$ AND &DF) =FNwildcard(M
ID$(wild$,2),MID$(match$,2))
24870 IF ASCwild$<>ASC"*" =FALSE
24880 =FNwildcard(MID$(wild$,2),match$)
OR FNwildcard(wild$,MID$(match$,2))
24890 :
24900 DEF FNstrip(A$)
24910 IFA$="" =""
24920 IF ASCA$=32 REPEAT A$=MID$(A$,2):U
NTIL ASCA$<>32
24930 IFA$="" =""
24940 IFRIGHT$(A$,1)=" " REPEAT A$=LEFT$
(A$,LENA$-1):UNTIL RIGHT$(A$,1)<>" "
24950 =A$
24960 :
24970 REM Current directory Path
24980 :
24990 DEF FNpath
25000 LOCAL pblk%,buffer%,A%,X%,Y%
25010 LOCAL I%,A$,B$,D$
25020 pblk%=&900:buffer%=pblk%+18
25030 A$=FNdir
```

```
25040 IF A$<>"$" THEN REPEAT OSCLI"DIR ^
":B$=FNdir:A$=B$+"."+A$:UNTILB$="$"
25050 A$=":"+CHR$(buffer%?1)+"."+A$
25060 OSCLI "DIR "+A$
25070 =A$
25080 :
25090 DEF FNdir
25100 pblk%!1=buffer%:pblk%!5=47
25110 pblk%!9=0:A%=6
25120 X%=pblk% MOD 256:Y%=pblk% DIV 256
25130 CALL &FFD1:D$=""
25140 FOR I%=buffer%+3 TO buffer%+2+buff
er%?2
25150 D$=D$+CHR$?I%:NEXT
25160 IF RIGHT$(D$,1)=" " REPEAT D$=LEFT
$(D$,LEND$-1):UNTIL RIGHT$(D$,1)<>" "
25170 =D$
25180 :
25190 REM String Quick Sort
25200 :
25210 DEF PROCinit
25220 DIM array$(20),mc% 100
25230 FOR pass=0 TO 2 STEP 2
25240 P%=mc%
25250 [OPT pass
25260 .caps  LDX #0
25270 .caps1 LDA s1%,X:BEQ notlower
25280        JSR cap:STA s1%,X
25290        LDA s2%,X:BEQ notlower
25300        JSR cap:STA s2%,X
25310        INX:BNE caps1
25320        RTS
25330 :
25340 .cap   CMP #ASC"a":BCC notlower
25350        CMP #ASC"z"+1:BCS notlower
25360        AND #&DF \ ensure capital
25370 .notlower RTS
25380 :
25390 .s1%   EQUS STRING$(20,CHR$0)
25400 .s2%   EQUS STRING$(20,CHR$0)
25410 ]NEXT
25420 ENDPROC
25430 :
25440 DEF PROCqsort(entries%)
25450 halfway%=entries% DIV 2
25460 IF halfway%=0 ENDPROC
25470 REPEAT
25480 limit%=entries%-halfway%
25490 ctr%=0
25500 REPEAT
25510 ptr1%=ctr%
25520 REPEAT
25530 ptr2%=ptr1%+halfway%
25540 $s1%=array$(ptr1%)+CHR$0
25550 $s2%=array$(ptr2%)+CHR$0:CALL caps
25560 IF $s1%>$s2% PROCswop:ptr1%=ptr1%-
halfway%
25570 UNTIL ptr1%<=0 OR $s1%<=$s2%
25580 ctr%=ctr%+1
25590 UNTIL ctr%>limit%
25600 halfway%=halfway% DIV 2
25610 UNTIL halfway%=0
25620 ENDPROC
25630 :
25640 DEF PROCswop
25650 LOCAL z$
25660 z$=array$(ptr1%)
25670 array$(ptr1%)=array$(ptr2%)
25680 array$(ptr2%)=z$
25690 ENDPROC
25700 :
25710 REM Capitals
25720 :
25730 DEF FNcap(A$)
25740 LOCAL B$,A%,I%
25750 B$=STRING$(LENA$,"*"):B$=""
25760 FORI%=1 TO LENA$:A%=ASCMID$(A$,I%)
:A%=A%+(A%>96 AND A%<123)*32:B$=B$+CHR$A
%:NEXT
25770 =B$
25780 :
25790 REM Anybase
25800 :
25810 DEF FNanybase(value$,base%)
25820 LOCAL A%,I%,A$,value%,digits%
25830 value%=EVAL(value$)
25840 I%=-1:REPEAT:I%=I%+1
25850 UNTIL value% DIV (base%^I%)=0
25860 digits%=I%-1
25870 FOR I%=digits% TO 0 STEP-1
25880 A%=(value% DIV (base%^I%)) MOD bas
e% +ASC"0"
25890 IF A%>ASC"9" A%=A%+7
25900 A$=A$+CHR$A%
25910 NEXT
25920 =A$
25930 :
```

Continued on page 44

VDU and FX Calls (5)

by Mike Williams

Let us continue with our exploration of FX calls, though we will not be dealing with all of these. Some are of limited application and/or adequately described in the User Guide (or Welcome Guide depending upon your machine). Others are too specialist or advanced technically to justify inclusion in this series, which after all is aimed at the near beginner. First of all this month, we will take a brief look at the calls controlling flashing colours.

FLASHING COLOURS

Flashing colours, by default the colours in the range 8 to 15, alternate between two colours. *FX9 and *FX10 can be used to control the length of time each of the two colours is visible on the screen. Note that *FX9,0 forces the first colour to be visible continuously, while *FX10,0 does the same for the second colour. With this call, times are expressed in fifthtieths of a second (see User Guide).

AUTO-REPEAT AND BOOT FILES

A similar group of calls controls the auto-repeat rate when pressing down keys on the keyboard. These are the calls FX11 and FX12. There are two parameters which you can control; one is the delay before a pressed key starts to auto-repeat, and the second is the repeat rate itself. The former is controlled by *FX11 with *FX11,0 disabling auto-repeat altogether, the latter by *FX12. With both these calls, as with several others, the main parameter specifies a time in centi-seconds (hundreths of a second). The defaults, for both parameters, can be reset with *FX12,0.

If you find the default settings inconvenient, work out what does suit you and then put these commands in a !BOOT file on a disc. Use:

```
*BUILD !BOOT
```

to initiate this process, and when prompted type in your commands, one per line. Finish by pressing Escape. Then type *OPT4,3 to complete the task. From then on, inserting that disc and pressing Shift-Break will cause the commands in the boot file to be executed. This is a convenient way to set up any defaults you require at the start of a session.

Of course, if you have a Master, then some of these default settings, like auto-repeat, can be saved in battery-backed RAM, so that they are invoked automatically when you switch your machine on. In addition, if you include any of these FX calls within a program, then do make sure you include the calls to restore your normal defaults when the program terminates. There's nothing worse than leaving a program to edit it, only to find auto-repeat set much too fast, or cursor editing disabled, not to mention other unwelcome settings.

BUFFER CONTROL

A couple of calls which I think are sometimes over-used are *FX15 and *FX21. Both of these in their different ways 'flush' one or more 'buffers'. The BBC micro is designed so that there is no direct connection between a program running on your machine, and any input or output device connected to your system.

For example, an area of memory is set aside as a *keyboard input buffer*. Whenever you type away at the keyboard, the characters (as ASCII codes) are not passed directly to any program, but enter the keyboard input buffer. When a program needs any input, it doesn't look at the keyboard directly (unless negative INKEY calls are used), but instead goes to the keyboard input buffer and extracts characters until its requirements are satisfied.

This has a number of advantages. It avoids possible problems of synchronisation. If you press a key before it's needed the resulting character just sits in the buffer. If the program looks for input before it has arrived, it just waits until something appears in the buffer. It is also the reason why you can type ahead of the ability of the system to keep up with you. Sometimes you may have heard a 'beep' from the micro while typing. One cause of this is that you have filled the input buffer (it's 256 bytes).

Of course, other problems can arise. If you keep a key depressed too long then auto-repeat comes into play, and several occurrences of the same character enter the keyboard input buffer. Sometimes this could be detrimental to the correct running of a program, and that's where 'flushing' the buffer comes in. The action of flushing simply clears all characters currently waiting in the buffer. Your next input will therefore be whatever new data is entered after the buffer has been flushed.

The call *FX15,0 will flush all buffers, while *FX15,1 flushes the current input buffer (usually the keyboard buffer - but see description of *FX2 in your manual) *FX21 performs the same kind of function, but the parameter supplied with the call specifies precisely which buffer to flush. I have seen programs, often games programs written in Basic, which seem to include many *FX15 or *FX21 calls. I find that they can often be removed with no ill affects, unless you are a particularly ham-fisted keyboard user. There is never any point in including more instructions in a program than you really need.

With reference to *FX21, you will see from your manual that *FX21,3 flushes the printer buffer, while *FX21,4 (to *FX21,7) will flush the sound channels. The latter can be useful to terminate a sound that seems reluctant to die, while flushing the printer buffer will prematurely terminate any printout, which again might be useful. For example, a program which outputs to a printer might offer the user the option to abort printing. The program can stop sending data to the printer buffer, but by using *FX21,3 it can also get rid of any outstanding data still in the printer buffer itself.

HANDLING ESCAPE

Although it is difficult to find many logical groupings among the remaining FX calls, a number of these are related to the Escape key and the consequences of pressing it. Again, if you write a program which alters the normal function of Escape, then do ensure that the program restores this on exit. This also applies to an error exit, i.e. where the trapping of an error through the use of ON-ERROR-GOTO leads to the termination of the program. This can be most evident during program development - run a program and then press Escape only to find it doesn't work or that your machine is left with inconvenient settings.

I have always avoided disabling the Escape function in most of my own programs, typically using Escape to take the program back to some convenient starting point (a main menu for example) when it is used to abort some operation. The menu can have an 'Exit' option, or use Shift-Escape to exit completely from a program. However, it is as well to know what else can be done, and the functions described here can have a role to play in the right context.

A simple FX call is *FX229. This can have one of two parameters. *FX229,1 causes Escape to generate an ASCII code (27), in just the same way that *FX4,2 causes the cursor keys and Copy to do likewise. This can make it much easier to cope with the pressing of the Escape key without all the usual side effects. Normally, pressing Escape, like any error condition, causes Basic to lose track of any FOR-NEXT or REPEAT-UNTIL loop which may be in operation, and any procedure or function which is being executed. By setting Escape to generate an ASCII code this can be checked for on input and any required action taken. The other version of this call (*FX229,0) simply restores the Escape key to its normal operation.

Another not dissimilar call is *FX200. *FX200,1 disables Escape so that it has no effect. Normal operation can be restored with *FX200,0. This could be used to prevent someone using Escape to prematurely terminate a program in appropriate cases, but if you use it, do make sure you have provided a way for the program to terminate without having to press Ctrl-Break and lose everything.

FUNCTION KEY DEFINITIONS

Most readers are no doubt familiar with the fact that the function keys can be programmed with any sequence of instructions they choose (within an overall limit of 255 characters). This uses the *KEY command. Readers will also be aware that many programs (like View for example) use not only the function keys alone, but also Shift-fkey combinations, Ctrl-fkey, and even Shift-Ctrl-fkey. This provides the opportunity for many more function key definitions than the single set of ten could manage alone.

But how do you program all these combinations yourself if needed? There are four FX calls which help:

*FX225	function keys alone (default)
*FX226	Shift-fkey combinations
*FX227	Ctrl-fkeys
*FX228	Shift-Ctrl-fkeys

Each of these FX calls is normally followed by a single parameter. If the parameter is 0, then pressing the function key combination has no effect; if the parameter is 1, pressing the function key combination generates the sequence with which it has been programmed (if any); if the parameter is in the range 2-255, then the function key combination generates an ASCII code based on the value of the parameter and the position of the function key.

For example, setting:

```
*FX225,240
```

would cause f0 to generate ASCII code 240, f1 to generate 241, f2 to generate 242 and so on. Writing:

```
*FX226,128
```

would cause Shift-f0 to generate 128, Shift-f1 to generate 129 and so on. This can be useful in mode 7 as these are Teletext colour codes, and this is in fact the default setting for Shift-fkey combinations.

On the other hand, if you want to program the Shift-fkey combinations to generate particular strings then you would proceed as follows:

```
*FX226,1
*KEY 0 <string 1>
*KEY 1 <string 2>
*KEY 2 <string 3>
```

and so on. Then pressing Shift-f0 would generate *string 1*, Shift-f1 would generate *string 2* etc. Executing *FX226,128 would return to the previous ASCII code generation; following this with *FX226,1 would restore the previous Shift-fkey definitions. Once programmed they would not need to be programmed again, unless changed or deleted.

One application of this technique can be used with View, Acorn's word processor. View itself uses the function keys on their own, with Shift, and with Ctrl, but this still leaves users with the option of programming the Shift-Ctrl-fkey combinations for themselves. Preceding the function key definitions with *FX228,1 permits this, and the whole operation can be automated by putting these instructions in a boot file, which ends with the command to invoke View itself:

```
*FX228,1
*KEY 0 <string 1>
*KEY 1 <string 2>
. . . . . . . . .
*KEY 9 <string10>
*WORD
```

Of course, any other appropriate commands could be included in the same boot file. For more information on this subject see BEEBUG Vol.8 No.9 where it has been covered in some detail.

DOWNLOAD ROUTINES

The model B in particular can be very short of memory space, especially in the higher resolution graphics modes. On a model B with the DFS fitted, the value of PAGE is raised from &E00 (on a cassette-based machine) to &1900 to provide buffer and other space for the filing system. On a Master 128 and Master Compact, these functions are located elsewhere and PAGE (by default) reverts back to &E00. That means that a disc-based model B loses &B00 (that's 2.75K in decimal - or 2816 bytes), quite a loss when working in modes 0, 1 or 2. Provided no more than one file is to be open at a time, the value of PAGE can be reduced to &1200, but this is sometimes still insufficient.

You cannot simply reduce the value of PAGE down to &E00 and load the program, as the filing system needs to use the memory from &E00 to &1200 to perform the load function itself. The solution is to load the program as normal, *and then move it down in memory.* The code to do this is usually referred to as a *Move-Down* routine, and can neatly make use of another FX call.

A typical move-down routine is as follows:

```
FOR I%=0 TO (TOP-PAGE) STEP 4
I%!&E00=I%!PAGE
NEXT I%
```

The use of the '!' indirection operator means that four bytes are moved at a time, not just one, hence the 'STEP 4'. If you are not sure about the detail of this routine then just take it from me that it works. On completion of the move-down, PAGE can be reset to &E00, and the DFS disabled with *TAPE.

So far so good, but the move-down routine is normally added at the start of the program to be moved, so that the move-down is accomplished first, but then it is part of the program being moved down. The answer here is to add lines at the start of the program which program a function key with the move-down routine, and then simulate the pressing of the relevant key; that's where the FX call comes in. Not only that, but the function key definition can conclude by deleting the extra lines of the move-down routine itself, thereby saving even more space.

Because it is being written as a function key definition, the move-down routine should preferably be written in an abbreviated form. Here's our final version of this routine:

```
   1 *KEY0 *T.|MF.I%=0TO(TOP-PAGE)STEP4:
I%!&E00=I%!PAGE:N.:PAGE=&E00|MOLD|MDEL.1
,3|MRUN|M
   2 *FX138,0,128
   3 END
```

Place this routine at the start of any Basic program, and on running the program it will be moved down in memory and then run, first deleting the move-down routine itself.

The *FX call which we have used to put this into effect is *FX138. This can be used to insert a character into any buffer. The keyboard input buffer is always buffer 0 (see earlier comments on *FX21), hence the first parameter. The second parameter is the code for the key press to be simulated, and 128 is the code for function key 0. So to summarise, the routine defines function key 0 with the move-down routine, and then simulates the pressing of this key by entering its code into the keyboard input buffer. And because the move-down routine is defined as a function key definition, it is unaffected by the fact that the defining program is being moved down in memory. Note, the pressing of any key can be simulated using *FX138, simply by specifying its ASCII code as the second argument to the FX function.

Well, that's all for this month. Next time we'll have one final look at some further FX calls before moving on to something else in Basic's rich repertoire.

BEEBUG Function/Procedure Library (continued from page 39)

```
25940 REM Any Base to Decimal
25950 :
25960 DEF FNbaseTOdec(value$,base%)
25970 LOCAL A%,I%,value%
25980 FOR I%=0 TO LENvalue$-1
25990 A%=ASC(RIGHT$(value$,I%+1))-ASC"0"
26000 IF A%>9 A%=A%-7
26010 value%=value%+A%*(base%^I%)
26020 NEXT
26030 =value%
26040 :
26050 REM check presence of ROM
26060 :
26070 DEF FNchip(A$)
26080 LOCAL I%,Y%,flag%
26090 FOR Y%=15 TO 0 STEP-1
26100 IF Y%?&2A1 PROCcheck(Y%,A$)
26110 IF flag% Y%=0
26120 NEXT
26130 =flag%
26140 :
26150 DEF PROCcheck(Y%,A$)
26160 flag%=TRUE
26170 ?&F7=&80:?&F6=&08
26180 FOR I%=1 TO LENA$:?&F6=?&F6+1
26190 IF (USR(&FFB9) AND &DF)<>(ASC(MID$
(A$,I%)) AND &DF) I%=LENA$:flag%=FALSE
26200 NEXT:ENDPROC
26210 :
26220 REM Master
26230 :
26240 DEF FNmaster=(INKEY-256=253 OR INK
EY-256=245)
26250 :
26260 REM VARTOP and FREE
26270 :
26280 DEF FNvartop=?2+?3*256
26290 :
26300 DEF FNfree=(?4+?5*256)-FNvartop
26310 :
26320 REM Error
26330 :
26340 DEF PROCerror(err,err$)
26350 $&C00=CHR$0+CHR$err+err$+CHR$0
26360 CALL &C00
26370 ENDPROC
```

512 Forum

by Robin Burton

Following on quite neatly from the previous Forum we're digging into the dark mysteries of batch files again. This month's article was prompted by a PC-using friend asking how such a short batch file could generate enough files to fill a directory in a test I outlined a couple of issues ago. There's also something new later which could double the value of your year's BEEBUG membership fee.

OK, perhaps batch files aren't dark and mysterious, but obviously some DOS users aren't too familiar with the lesser used functions like parameter substitution, 'IF variable == variable' and 'SHIFT'. I thought we should take a look.

DEFINING THE JOB

You may recall that my test required the (over) filling of a directory with files. Naturally I wanted to do it in the easiest manner possible, which in my book means the least amount of effort, particularly in terms of manual entry.

My test was of a hard disc partition, but the purpose here is investigating batch files, so floppy users needn't 'switch off', but should join in too (and run times using 800K discs aren't all that much longer than for a winchester).

Using floppies, I suggest you format a fresh disc and use the root directory to ensure maximum speed (these batch runs are pretty heavy going, even for a winchester). If you use a hard disc for these jobs, create a test directory in the root and use that.

N.B. DO NOT use the root directory of your winchester. If it fills to capacity DOS Plus (in the 512, at least) definitely doesn't like it - judging by my tests it can even damage the partition. This is not a problem in sub-directories, which appear to expand as required without practical limits.

In all cases it's best to run these tests in an EMPTY directory or they may not produce the correct results, because the order in which new files are created in the directory is important.

When you delete a file in DOS (unlike the DFS for example) the remaining directory entries are NOT concatenated. In DOS all deleted file entries in the directory stay precisely where they are until new files need to use those entries again. In an 'old', well used directory therefore, the gaps left by deleted files can be all over the place.

THE DIRECT APPROACH

How to go about it? The starting point is an empty directory, so the first job is to create one file as the source for the copy operation. Also, for quickness, the source file should be kept short, not more than a cluster long in fact.

The easiest way to produce a suitable file is to use 'COPY CON <filename>', then press Ctrl-Z followed immediately by Return, which will create a one byte file. For our tests, call this first file 'A', for reasons that will become clear as we proceed.

Now let's see if we can make the system do most of the work. Probably the most obvious strategy would be to copy the first file to another, then to copy both resulting files again and so on, doubling the number produced each time. In this way, after only a few commands, the number of files will be growing rapidly. However, the most obvious method isn't the best in this case and we can do much better.

I know that an 800K floppy only holds 192 directory entries, but ignore that for now. Concentrate on the fact that our real interest is the batch file itself, not filling the directory.

One difficulty with any approach to producing lots of files automatically is inventing suitable names that can be used with wildcards (the aim being to copy all existing files every time) without producing duplicate filenames. This would at least waste time and could cause a failure in other circumstances.

Using a bit of imagination it can be done. Starting with file 'A' we could for example do this, with these results:

Command	New files
COPY ??????? A*	7
COPY ??????? B*	28
COPY ??????? C*	84
COPY ??????? D*	210

It's crude, but it works! Only four copy commands and we have a total of 329 new files. For the job to be run repeatedly it only needs to be saved as a batch file containing commands like the above. If you find the results surprising and want to try these commands you can do so either in a batch file, or more simply by entering them manually in sequence. The results are precisely the same in both cases. If you weren't expecting so many files from so few commands the process is fully explained later.

We could be satisfied with this and get on with the job. However, all that can be learned from it is the power of DOS wildcard copy commands, but not much about batch files, so let's look further for a more flexible and elegant technique. This should produce at least as many files as the example above, ideally more, from the same or fewer manual entries.

Before we move on, there are a couple of points to note about the commands in the list above. Notice that the source filenames are specified as seven '?'s, rather than one '*'. This is necessary for two reasons. First, and the important one here, if a '*' is used instead, the instant result is a failure, because the first command will immediately attempt to copy the first source file to itself.

Even if that wasn't a problem, you'd find as the job progressed that a '*' would include existing filenames of eight characters, so a lot of copies (the sixteenth and every eighth one following for each new letter) would produce duplicate names and waste time. This point becomes more relevant in the next batch file.

MORE AUTOMATION

A better approach (call it TEST.BAT) looks like this. When you run it, you'll see that essentially it does the same job as the previous file, but this time the job is totally automated, we don't even need the first source file because that's included in the job too. How this job works is explained below in some detail so that you can follow what each of the commands does. You'll need to think about these commands more than the previous ones, because there's a lot of character substitution going on.

```
COPY %0 %1
:START
COPY ??????? %1*
IF %1 == D EXIT
```

```
   COPY %1 %2
   SHIFT
   GOTO START
```

Again, seven question marks are used so that eight character names aren't copied. This file (more lines but about the same amount of typing) is called with a number of parameters, rather than having values fixed in the file, so it's much more flexible, and the file will never grow no matter how many sets of files we want to produce. To exactly mimic our first batch file's actions therefore, we'd call it with the command:

```
TEST A B C D
```

You'll notice the '%1' and '%2' in the file; these are the parameter identifiers and this is how they work. In this command line entry the first string is 'TEST', and the batch variable %0 is always set to the batch filename (minus extension) by DOS automatically. The second parameter is 'A', so this value is substituted for '%1', 'B' is used for %2, 'C' for %3 (if it's specified) and so on.

The first line copies '%0.BAT' (i.e. TEST.BAT) to %1 (A) and this therefore creates our first source file for us. Since the batch file's extension prevents it qualifying as a source file in this job, it can be copied to the test directory each time without interfering with the run.

Next, line 2 is passed over the first time through because it's a label. The next executable line, line 3, says, "For every occurrence of any filename of up to seven characters in length with no file extension, copy that file to another giving it a name starting with the current value of %1 followed by the characters in the source filename.

As the job starts '%1' evaluates to 'A', so the first execution of line 3 copies every occurrence of '???????' (file 'A' only, the first time) to a new file called %1 plus whatever '???????' represents, creating in this case, file 'AA'.

Now this next bit is probably the most difficult part to understand, both in this batch file and the first one, because there's no equivalent situation in BBC micro filing system commands. Immediately following the first execution of line 3 we have two files in the directory. One of these, file 'A', has been processed, the second, file 'AA', has not.

Batch file commands are interpreted and evaluated in real time, just like manual ones, and because line 3 says "For EVERY occurrence of name ??????? copy that file...." the command processor immediately applies the command to the only file remaining unprocessed in the directory, our new file 'AA'.

Parameter %1 still equates to 'A', so file 'AA' is duly copied to file 'AAA'. This then becomes the only unprocessed file, so it's copied to 'AAAA' and so on until all (seven character) source filenames have been processed. Input to line 3 is then exhausted, so a new situation arises. Since no qualifying files in the directory remain uncopied, execution passes to the next command.

Ignore line 4 for the present (we'll come back to it) and look at line 5. At this stage %1 = 'A' and %2 = 'B', the second and third parameters from our original command entry. File 'A' is therefore copied to a new file called 'B'. The fifth line, 'SHIFT' then tells the batch file processor to move all parameters left by one place, so the old %1 (A) becomes the new %0 the old %2 (B) becomes the new %1, the old %3 (C), even though it's not referenced directly in our file, becomes the new %2 and so on. This operation takes place for as many parameters as were supplied, up to the maximum length of a command line.

The reason for the 'SHIFT' command is that you can only specify parameter names from %0 to %9 in batch files. However, because SHIFT operates on all parameters supplied in the original command line, any which were initially out of reach (the 10th, 11th, 12th, etc.) or simply not referred to (in our case %3, %4 etc.) can be accessed after an appropriate number of 'SHIFT's. Remember though, this is strictly a one way process. The old %0 value is permanently lost after every SHIFT. In other words, our original parameters, evaluated the first time through by the batch processor as:

```
    %0  %1 %2 %3 %4
  TEST  A  B  C  D
```

become:

```
  %0 %1 %2 %3
   A  B  C  D
```

Following the 'SHIFT', execution is then returned to line 2 by 'GOTO START' in the last line, so the whole process begins again, this time copying file 'A' to 'BA', 'AA' to BAA, and so on up to BAAAAAAA, then all the new 'B' files follow too. After this the parameters are again shifted left by one place and the process can continue for as long as is required.

There are a couple of extra points to note before we leave the explanations. First, I've used single alpha character parameter entries here for obvious reasons, but batch parameters can be anything you like in the real world, so long as they're valid for the purpose intended. Also, the filenames used could include extensions, and either could be made up of parameters, wildcards, hard coded values or any combination of any of these.

Next, back to line 4 for a moment. In this file the line:

```
    IF %1 == D EXIT
```

ends the run when parameter %1 becomes equal to 'D', because the 'EXIT' command is then executed. All you need to do is to change the 'D' to another character to make the run longer or shorter. Just make sure the value used in the test is the last parameter you want the file to process. Note also that the '==' is not a misprint; there must be two of them.

Finally, remember that upper case has been used here purely for clarity. In batch files you MUST ensure that the case is consistent between any test and the variables which will be supplied. Batch file '==' tests are case sensitive, so 'D' and 'd' are NOT the same so far as the batch file processor is concerned. Whichever case a batch file test specifies, the variables compared must be the same case or the test will fail. It's obviously simplest to stick to lower case throughout, as is usual for DOS commands and filenames.

MORE DEVIOUS

Back to operations. This batch file does much the same as the first one, and it's much more flexible too. We've also satisfied our second requirement, less manual entry, because the only thing that now increases, regardless of how many files we might want to generate, is the number of parameters and they're only one character each.

However, compare the results of two jobs and you'll find there's another difference between them. For %1 = 'A' you'll get the same eight filenames as before, 'A' to 'AAAAAAAA', but when %1 = 'B' things change. In the second version of the file, from set 'B' onwards, apparently similar operations produce different results. In the first file, 'A' to 'D' produce 330 files, but this time the same range produces almost 500!

Why more files? Actually I've already mentioned the reason, but you might have missed its significance. Look at line 5 again. It seems innocent enough, but this simple step creates the first file of the next set before that set is processed. However, the result is an increase in the number of files in the next set by the total number of all files in all preceding sets.

Similar operations could have been included in the first file, but in that case they would literally have doubled the number of lines in the job. Since every extra set in the first file already requires an extra command line, this would make it twice as bad - definitely a backward step if economy of effort is our aim.

In set 'B' this small change adds 8 files (the total of set A) giving 36 'B' files (8+7+6+5+4+3+2+1) instead of the 28 of the first job. Things get more complicated after that though. If that simple calculation were applied to set 'C', i.e. the sum of numbers 1 to 44 (8+36), you'd expect 990 files, yet only about an eighth of this number are produced, so what's happening? And why am I being vague about the totals? Wait and see.

Two factors are at work. First, for each succeeding set of files an ever growing number of existing filenames are eight characters long and can't be used again. Secondly, in each succeeding set of files, an increasing number of new filenames will have fewer unused characters than in previous sets, so (proportionally) fewer new files can be created from them.

By the way, I never ran my tests with more than five parameters, but if you have a hard disc and want to try, good luck! I'd guess with six parameters you'd have to leave the machine running for maybe 12 hours, with eight it would be more like a week (assuming it didn't crash).

SOMETHING EXTRA

If you are wondering why I didn't list more complete results from this job it's because we're having a competition.

Prizes of a hundred pounds worth of Essential Software's 512 products are up for grabs. There's a new package too, so existing users needn't feel left out. If you're unfamiliar with the product range consult BEEBUG Vol.8 No.5, Vol.9 Nos.1, 3 & 10, the September '90 and January '91 BBC Acorn User, or The 512 Technical Guide from Dabs Press. Alternatively, send an S.A.E. to the address below for an up to date list.

Each of the first five correct entries out of the hat after the closing date will each win a £20.00 voucher which can be spent on any Essential Software products (including the memory expansion).

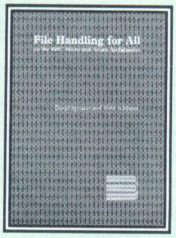
To enter you must work out the number of files which would be generated by the second batch file for each file set from 'A' to 'H' (in alphabetic order) and say whether or not the number of new files produced by any individual set (A to Z in alpha order) would reach a million. If yes, which one is it? There are no tricks, so ignore all practical limitations.

One entry only per BEEBUG member please, and you must include your name, address and your BEEBUG membership number. To give overseas members sufficient time to enter, the closing date is not until Friday, September 20th.

Winners will be notified directly and their names plus the answers will be published in the November issue of BEEBUG.

Please send entries/product list enquiries to Essential Software, P.O. Box 5, Groby, Leicester LE6 0ZB.

Best of BEEBUG


File Handling for All
on the BBC Micro and Acorn Archimedes
David Spencer and Mike Williams

File Handling for All
on the BBC Micro and Acorn Archimedes
by David Spencer and Mike Williams

Computers are often used for file handling applications yet this is a subject which computer users find difficult when it comes to developing their own programs. *File Handling for All* aims to change that by providing an extensive and comprehensive introduction to the writing of file handling programs with particular reference to Basic.

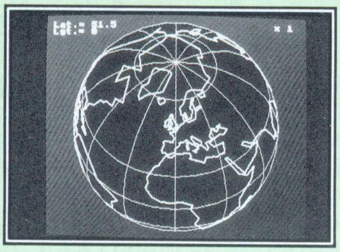
File Handling for All, written by highly experienced authors and programmers David Spencer and Mike Williams, offers 144 pages of text supported by many useful program listings. It is aimed at Basic programmers, beginners and advanced users, and anybody interested in File Handling and Databases on the Beeb and the Arc. However, all the file handling concepts discussed are relevant to most computer systems, making this a suitable introduction to file handling for all.

The book starts with an introduction to the basic principles of file handling, and in the following chapters develops an in-depth look at the handling of different types of files e.g. serial files, indexed files, direct access files, and searching and sorting. A separate chapter is devoted to hierarchical and relational database design, and the book concludes with a chapter of practical advice on how best to develop file handling programs.

The topics covered by the book include:

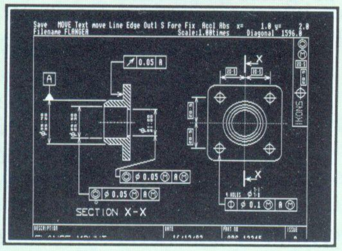
Card Index Files, Serial Files, File Headers, Disc and Record Buffering, Using Pointers, Indexing Files, Searching Techniques, Hashing Functions, Sorting Methods, Testing and Debugging, Networking Conflicts, File System Calls

The associated disc contains complete working programs based on the routines described in the book and a copy of Filer, a full-feature Database program originally published in BEEBUG magazine.



Applications I Disc

Business Graphics - for producing graphs, charts and diagrams
Video Cataloguer - catalogue and print labels for your video cassettes
Phone Book - an on-screen telephone book which can be easily edited and updated
Personalised Letter-Headings - design a stylish logo for your letter heads
Appointments Diary - a computerised appointments diary
Mapping the British Isles - draw a map of the British Isles at any size
Selective Breeding - a superb graphical display of selective breeding of insects
Personalised Address Book - on-screen address and phone book
The Earth from Space - draw a picture of the Earth as seen from any point in spac
Page Designer - a page-making package for Epson compatible printers
World by Night and Day - a display of the world showing night and day for any time an date of the year


Save MOVE Text move Line Edge Outl S Fore Fix Accl Abs
Filename FLANGER
Scale:1.00times Diagonal 1596.8
SECTION X-X

ASTAAD

Enhanced ASTAAD CAD program for the Master, offering the following features:

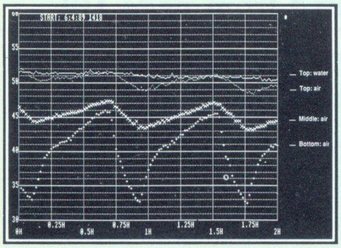
- full mouse and joystick control
- built-in printer dump
- speed improvement
- STEAMS image manipulator
- Keystrips for ASTAAD and STEAMS
- Comprehensive user guide
- Sample picture files

	Stock Code	Price		Stock Code	Price
ASTAAD (80 track DFS)	1407a	£ 5.95	**ASTAAD** (3.5" ADFS)	1408a	£ 5.95
EDIKIT (EPROM)	1451a	£ 7.75			
EDIKIT (40/80T DFS)	1450a	£ 5.75	**EDIKIT** (3.5" ADFS)	1452a	£ 5.75
Applications II (80 track DFS)	1411a	£ 4.00	**Applications II** (3.5" ADFS)	1412a	£ 4.00
Applications I Disc (40/80T DFS)	1404a	£ 4.00	**Applications I Disc** (3.5" ADFS)	1409a	£ 4.00
General Utilities Disc (40/80T DFS)	1405a	£ 4.00	**General Utilities Disc** (3.5" ADFS)	1413a	£ 4.00

Please add p&p

BEEBUG Ltd, 117 Hatfield Road, St Albans, Herts AL1 4JS.

Applications II Disc



Crossword Editor - for designing, editing and solving crosswords

Monthly Desk Diary - a month-to-view calendar which can also be printed

3D Landscapes - generates three dimensional landscapes

Real Time Clock - a real time digital alarm clock displayed on the screen

Running Four Temperatures - calibrates and plots up to four temperatures

Julia Sets - fascinating extensions of the Mandelbrot set

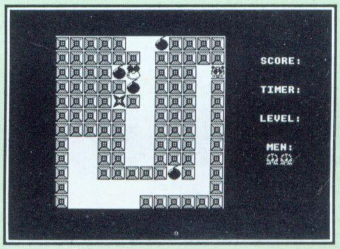
Foreign Language Tester - foreign character definer and language tester

Label Processor - for designing and printing labels on Epson compatible printers

Share Investor - assists decision making when buying and selling shares.

Best of BEEBUG

Arcade Games



George and the Dragon - Rescue 'Hideous Hilda' from the flames of the dragon, but beware the flying arrows and the moving holes on the floor.

Ebony Castle - You, the leader of a secret band, have been captured and thrown in the dungeons of the infamous Ebony Castle. Can you escape back to the countryside, fighting off the deadly spiders on the way and collecting the keys necessary to unlock the coloured doors?

Knight Quest - You are a Knight on a quest to find the lost crown, hidden deep in the ruins of a weird castle inhabited by dangerous monsters and protected by a greedy guardian.

Pitfall Pete - Collect all the diamonds on the screen, but try not to trap yourself when you dislodge the many boulders on your way.

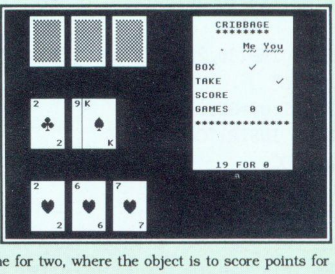
Builder Bob - Bob is trapped on the bottom of a building that's being demolished. Can you help him build his way out?

Minefield - Find your way through this grid and try to defuse the mines before they explode, but beware the monsters which increasingly hinder your progress.

Manic Mechanic - Try to collect all the spanners and reach the broken-down generator, before the factory freezes up.

Quad - You will have hours of entertainment trying to get all these different shapes to fit.

Board Games



Solitaire - an elegant implementation of this ancient and fascinating one-player game, and a complete solution for those who are unable to find it for themselves.

Roll of Honour - Score as many points as possible by throwing the five dice in this on-screen version of 'Yahtze'.

Patience - a very addictive version of one of the oldest and most popular games of Patience.

Elevenses - another popular version of Patience - lay down cards on the table in three by three grid and start turning them over until they add up to eleven.

Cribbage - an authentic implementation of this very traditional card game for two, where the object is to score points for various combinations and sequences of cards.

Twiddle - a close relative of Sam Lloyd's sliding block puzzle and Rubik's cube, where you have to move numbers round a grid to match a pattern.

Chinese Chequers - a traditional board game for two players, where the object is to move your counters, following a pattern, and occupy the opponent's field.

Aces High - another addictive game of Patience, where the object is to remove the cards from the table and finish with the aces at the head of each column.

	Stock Code	Price
Arcade Games (40/80 track DFS)	PAG1a	£ 5.95
Board Games (40/80 track DFS)	PBG1a	£ 5.95
File Handling for All Book	BK02b	£ 9.95
File Handling for All Disc (40/80T DFS)	BK05a	£ 4.75
Joint Offer book and disc (40/80T DFS)	BK04b	**£ 11.95**

	Stock Code	Price
Arcade Games (3.5" ADFS)	PAG2a	£ 5.95
Board Games (3.5" ADFS)	PBG2a	£ 5.95
File Handling for All Disc (3.5" ADFS)	BK07a	£ 4.75
Joint Offer book and disc (3.5" ADFS)	BK06b	**£ 11.95**

Please add p&p. UK: £1.00 first item (50p for every additional item), Europe and Eire: £1.60 first item (80p every additional item), Elswhere: £2.60 first item (£1.30 every additional item)

Tel. (0727) 40303 Fax. (0727) 860263

```
aper."
 1470 PRINT TAB(13,9)"On paper measure 1
6 squares in mm horizontally & verticall
y."
 1480 PRINT TAB(13,10)"Give these mm val
ues to hori & vert in PROCaspect."
 1490 PRINT TAB(13,11)"Re-SAVE the progr
am."
 1500 PRINT TAB(10,13)"2  Enter grid int
erval (25/32/50/64):"
 1510 REPEAT:PRINT TAB(48,13)SPC(32):VDU
 31,48,13:INPUT""st$:UNTIL st$="25" OR s
t$="32" OR st$="50" OR st$="64"
 1520 st%=VAL(st$)
 1530 INPUT TAB(10,15)"3  Enter grid ori
gin X value (0 to 1279): "xo%
 1540 IF xo%<0 THEN xo%=0
 1550 IF xo%>1279 THEN xo%=1279
 1560 INPUT TAB(10,16)"   Enter grid ori
gin Y value (0 to 1023): "yo%
 1570 IF yo%<0 THEN yo%=0
 1580 IF yo%>1023 THEN yo%=1023
 1590 PRINT TAB(10,18)"4  After plotting
 grid:"
 1600 PRINT TAB(20,19)"Press  P  for PRI
NTER dump"
 1610 PRINT TAB(20,20)"Press  Q  to  QUI
T"
 1620 PRINT TAB(30,22)"But now press  C
 to Continue"
 1630 REPEAT:g$=GET$
 1640 UNTIL INSTR("Cc",g$)
 1650 ENDPROC
 1660 :
 1670 DEF PROCdump
 1680 REM Put your own printer dump here
 1690 REM This one is for PrintMaster &
Epson FX80
 1700 *GDUMP 3 1 3 1 20
 1710 ENDPROC
 1720 :
 1730 DEF PROCaspect
 1740 hori=170
 1750 vert=170
 1760 ENDPROC
```

Listing 3

```
   10 REM Program GFL
   20 REM Ground Front Large
   30 REM Plots plan of room and overlay
s grid squares
   40 REM Version B2.0
   50 REM Author  Cliff Blake
   60 REM BEEBUG  Aug/Sept 1991
   70 REM Program subject to copyright
   80 :
  100 MODE 7:PROCdata:PROCinput
  110 MODE 0:PROCtitle
  120 PROCplot:PROCgrid
  130 VDU 7
  140 REPEAT
  150 REPEAT:g$=GET$:UNTIL INSTR("RrPpIi
Qq",g$)
  160 IF g$="R" OR g$="r" THEN CLS:RUN
  170 IF g$="P" OR g$="p" THEN PROCdump
  180 IF g$="I" OR g$="i" THEN CLS:CHAIN
 "INDEX"
  190 IF g$="Q" OR g$="q" THEN CLS:END
  200 VDU 7
  210 UNTIL FALSE
  220 END
  230 :
 1000 DEF PROCdump
 1010 REM Put your own printer dump here
 1020 REM This one is for PrintMaster &
Epson FX80
 1030 *GDUMP 3 1 3 1 15
 1040 ENDPROC
 1050 :
 1060 DEF PROCgrid
 1070 step=side*scale*mult
 1080 FOR x=dx TO 1279 STEP step
 1090 MOVE x,0:PLOT 21,x,850
 1100 NEXT x
 1110 FOR y=dy TO 850 STEP step
 1120 MOVE 0,y:PLOT 21,1279,y
 1130 NEXT y
 1140 ENDPROC
 1150 :
 1160 DEF PROCplot
 1170 x=x(0)*scale:y=y(0)*scale
 1180 MOVE x,y
 1190 FOR n%=1 TO final%
 1200 x=x(n%)*scale:y=y(n%)*scale
 1210 DRAW x,y
 1220 NEXT n%
 1230 ENDPROC
 1240 :
 1250 DEF PROCtitle
 1260 VDU 5
 1270 MOVE 0,988:PRINT title$
 1280 MOVE 0,940
 1290 PRINT;side;unit$;"squares"
 1300 MOVE 0,892
 1310 PRINT"Screen scale factor is ";sca
le
 1320 MOVE 640,988
 1330 PRINT hori$;unit$;"horizontal offs
et"
 1340 MOVE 640,940
 1350 PRINT vert$;unit$;"vertical offset
"
 1360 VDU 4
 1370 ENDPROC
 1380 :
 1390 DEF PROCinput
 1400 PRINT TAB(2,0)title$
 1410 PRINT TAB(5,2)"Select Grid Square
Units"
 1420 PRINT TAB(9,4)"1...centimetres"
 1430 PRINT TAB(9,6)"2...inches"
 1440 REPEAT:g%=GET-48:UNTIL g%>0 AND g%
<3
 1450 IF g%=1 THEN c%=27:unit$=" cm ":un
its$=" cm ":mult=1
 1460 IF g%=2 THEN c%=31:unit$=" inch ":
units$=" inches ":mult=2.54
 1470 PRINT TAB(5,8)"How many";units$;"p
er side?"
 1480 REPEAT:VDU 31,c%,8:INPUT""side$:UN
TIL side$<>""
 1490 side=VAL(side$)
 1500 IF side<10 AND g%=1 THEN side=50
 1510 IF side<6 AND g%=2 THEN side=12
 1520 :
 1530 PRINT TAB(5,11)"Select Grid Offset
 in";units$
 1540 PRINT TAB(9,13)"Horizontal?"
 1550 REPEAT:VDU 31,21,13:INPUT""hori$:U
NTIL hori$<>""
 1560 dx=VAL(hori$)*scale*mult
 1570 PRINT TAB(9,15)"Vertical?"
 1580 REPEAT:VDU 31,21,15:INPUT""vert$:U
NTIL vert$<>""
 1590 dy=VAL(vert$)*scale*mult
 1600 ENDPROC
 1610 :
 1620 DEF PROCdata
 1630 READ title$:READ scale
 1640 DIM x(50),y(50):n%=-1
 1650 :
 1660 REPEAT
 1670 n%=n%+1
 1680 READ x(n%),y(n%)
 1690 UNTIL x(n%)=9E9 AND y(n%)=9E9
 1700 final%=n%-1
 1710 ENDPROC
 1720 :
 9000 REM *** TITLE for room plan ***
 9010 DATA "GROUND FRONT LARGE ROOM"
 9020 REM *** SCALE is chosen to get the
 room plan within the grid area ***
 9030 DATA 2.4
 9040 REM *** COORDINATES must be in cen
timetres as measured ***
 9050 DATA 88,0
 9060 DATA 392,0
 9070 DATA 392,78
 9080 DATA 431,78
 9090 DATA 471,118
 9100 DATA 471,229
 9110 DATA 431,269
 9120 DATA 392,269
 9130 DATA 392,348
 9140 DATA 274,348
 9150 DATA 274,312
 9160 DATA 257,312
 9170 DATA 257,275
 9180 DATA 251,269
 9190 DATA 140,269
 9200 DATA 134,275
 9210 DATA 134,309
 9220 DATA 116,309
 9230 DATA 116,347
 9240 DATA 0,347
 9250 DATA 0,66
 9260 DATA 7,60
 9270 DATA 11,53
 9280 DATA 76,6
 9290 DATA 84,4
 9300 DATA 88,0
 9310 REM  *** TERMINAL VALUES ***
 9320 DATA 9E9,9E9
```

Listing 2

```
   10 REM Program Enter
   20 REM BEEBUG  Aug/Sept 1991
   30 REM Subject to copyright
  100 ONERRORGOTO2630
  110 pr$="Press Return":DIMA$(50)
  120 MODE3:PROCset:REPEAT:PROCmen:UNTIL
M%=49
  130 CLS:VDU31,5,3:PROCdis
  140 REPEAT:REPEAT
  150 IFPOS<5REPEAT:VDU9:UNTILPOS>4
  160 IFVPOS<3THENPROCdn:VDU10:ELSEIFVPO
S=24THENPROCup:VDU11
  170 C%=POS:D%=VPOS:PROCval:VDU31,C%,D%
:B%=GET:PROCky:UNTILB%>31
  180 IFN%=1PROCins ELSEPROCscr
  190 UNTILFALSE
  200 END
 1000 DEFPROCset
 1010 E%=0:O%=-3:N%=-1:B%=0:U%=0:V%=0:W%
=0:PROCblk:B$=STRING$(76," "):A$=B$:kw$=
STRING$(60,"+"):pg%=0:ti$=STRING$(50,"#"
)
 1020 Z%=OPENUP"TEXT":pg%=BGET#Z%:IFpg%=
0 ptr%=&200:ELSE PTR#Z%=2*pg%-1:ptr%=BGE
T#Z%:ptr%=ptr%*256+BGET#Z%:ptr%=ptr%*77+
&200
 1030 pg%=pg%+1:bal%=(EXT#Z%-ptr%)/77-2:
index%=PTR#Z%:CLOSE#Z%
 1040 ENDPROC
 1050 DEFPROCblk:FORA%=0TO50:A$(A%)=STRI
NG$(75," "):NEXT:ENDPROC
 1060 DEFFNnum(a$):PRINTTAB(0,2)a$;"    =
===";:INPUTTAB(LENa$+1,2)""a%:=a%
 1070 DEFFNstr(a$):PRINTTAB(0,2)a$;"
     ";:INPUTTAB(LENa$+1,2)""a$:=a$
 1080 DEFFNcol(A):=LEFT$((STR$(A)+"     "
),5)
 1090 DEFFNlast:LOCALA:R%=0:FORA=49TO0ST
EP-1:IFA$(A)<>STRING$(75," ")R%=A+1:A=0
 1100 NEXT:=R%
 1110 DEFPROCky
 1120 IFB%=127ANDVPOS=3ANDPOS=5 B%=0:END
PROC
 1130 IF(B%=127ORB%=136)ANDPOS=5 VDU8,8,
8,8,8
 1140 IFB%>31 AND B%<128:ENDPROC
 1150 IFINKEY(-1)PROCshf:ENDPROC
 1160 IFINKEY(-2)PROCctr
 1170 IFB%=9 T%=FNtab:VDU31,T%,VPOS;
 1180 IFB%=13VDU13,10
 1190 IFB%=27PROCmen:VDU31,C%,D%;:PROCdi
s
 1200 IFB%>135ANDB%<140VDU(B%-128)
 1210 IFB%=239 N%=N%*(-1)
 1220 IFB%>239ANDB%<246 ENDPROC
 1230 IFB%=246 PROCcase
 1240 IFB%=248 VDU31,5,3
 1250 B%=0:ENDPROC
 1260 DEFPROCshf
 1270 IFB%=9 D%=VPOS:PROCst:PROCrul:VDU3
1,0,D%
 1280 IFB%=138PROCdn ELSEIFB%=139PROCup:
ELSEIFB%=137PROCrt:ELSEIFB%=136PROClf
 1290 IFB%=148PROCkw
 1300 IFB%=149kw$=STRING$(60,"+"):PROCdi
s
 1310 IFB%=145:B$=A$(VPOS+O%):C%=POS:D%=
VPOS:B%=32:REPEAT:PROCscr:UNTILPOS=0:VDU
31,C%,D%
 1320 B%=0:ENDPROC
 1330 DEFPROCctr
 1340 IFB%=9D%=VPOS:PROCct:PROCrul:VDU31
,0,D%
 1350 IFB%=138 O%=26:PROCdis:ELSEIFB%=13
9 O%=-3:PROCdis
 1360 IFB%=14B$=A$(VPOS+O%)
 1370 IFB%=15A$(VPOS+O%)=B$:PRINTTAB(5,V
POS);B$;CHR$11;
 1380 IFB%=16FORA%=VPOS+O%TO49:A$(A%)=A$
(A%+1):NEXT:A$(50)=STRING$(75," "):VDU11
:PROCdis
 1390 IFB%=17VDU10:FORA%=50TOVPOS+O%+1 S
TEP-1:A$(A%)=A$(A%-1):NEXT:A$(VPOS+O%)=S
TRING$(75," "):PROCdis
 1400 IFB%=18 PROCasci
 1410 IFB%=19B$=A$(VPOS+O%):A$(VPOS+O%)=
STRING$(75," "):PRINTTAB(5,VPOS);SPC(75)
;
 1420 IFB%=20PROCcen
 1430 IFB%=21PROCedit
 1440 IFB%=22PROCmove
 1450 IFB%=23PROCdelblk
 1460 B%=0:ENDPROC
 1470 DEFPROCscr
 1480 IFB%=127THENVDU8,32,8:B%=32:A$(VPO
S+O%)=LEFT$(A$(VPOS+O%),POS-5)+CHR$(B%)+
RIGHT$(A$(VPOS+O%),79-POS):ENDPROC
 1490 IFPOS<5VDU31,5,VPOS
 1500 VDUB%:IFPOS=71THENVDU7
 1510 VDU8:A$(VPOS+O%)=LEFT$(A$(VPOS+O%)
,POS-5)+CHR$(B%)+RIGHT$(A$(VPOS+O%),79-P
OS):VDU9
 1520 ENDPROC
 1530 DEFPROCins
 1540 IFPOS=70 AND B%<>127 VDU7
 1550 IFB%=127ANDPOS=5VDU8,8,8,8,8
 1560 IFB%=127THENVDU8
 1570 J%=VPOS+O%:K%=POS-5
 1580 REPEAT:IFB%<>127THENA$=LEFT$(A$(J%
),K%)+CHR$(B%)+RIGHT$(A$(J%),75-K%)ELSEA
$=LEFT$(A$(J%),K%)+MID$(A$(J%),K%+2)
 1590 IFB%<>127THENA$(J%)=LEFT$(A$,75)EL
SEA$(J%)=A$+CHR$32:IFK%>70 A$=A$+"*"
 1600 IFB%=127 ANDRIGHT$(A$,2)="  "THENV
DU8
 1610 IFRIGHT$(A$,2)="  "THENC%=POS:D%=V
POS:L%=VPOS+O%:REPEAT:PRINTTAB(5,VPOS);A
$(L%);:L%=L%+1:UNTILL%>J%ORVPOS=24:PRINT
TAB(C%,D%);CHR$9;:B%=0
 1620 IFB%>0 THEN IFB%<>127 B%=ASC(RIGHT
$(A$,1))ELSE IFB%=127 A$(J%)=LEFT$(A$(J%
),74)+LEFT$(A$(J%+1),1)
 1630 IFB%>0J%=J%+1:K%=0
 1640 UNTILB%=0:ENDPROC
 1650 DEFPROCcase
 1660 J%=VPOS+O%:K%=POS-4
 1670 A%=ASC(MID$(A$(J%),K%,1)):B%=A%EOR
32:IFB%<32ORB%>127 ENDPROC
 1680 IFB%EOR32=A% PROCscr
 1690 ENDPROC .
 1700 DEFPROCrt
 1710 REPEAT:VDU9:UNTILMID$(A$(VPOS+O%),
POS-4,1)=" "OR POS=79:ENDPROC
 1720 DEFPROClf
 1730 REPEAT:VDU8:UNTILMID$(A$(VPOS+O%),
POS-4,1)=" "OR POS=5:ENDPROC
 1740 DEFFNtab
 1750 T%=POS:REPEAT:T%=T%+1:UNTILT%=U%OR
T%=V%ORT%=W% ORT%>79:IFT%>79 =0 ELSE=T%
 1760 DEFPROCst
 1770 IFU%=0THENU%=POS ELSEIFV%=0THENV%=
POS ELSEIFW%=0THENW%=POS
 1780 ENDPROC
 1790 DEFPROCct
 1800 IFW%=POS W%=0ELSEIFV%=POS V%=0ELSE
IFU%=POS U%=0
 1810 ENDPROC
 1820 DEFPROCasci:f$=FNstr("Enter filena
me "):IFf$="" ENDPROC
 1830 Q%=FNnum("To what screen line? "):
Z=OPENUPf$:IFZ=0 R$=FNstr(ns$+"FILE, "+p
r$):ENDPROC
 1840 Q%=Q%-1:
 1850 REPEAT:A$="":
 1860 REPEAT:B%=BGET#Z:IFB%>31A$=A$+CHR$
B%
 1870 UNTILLENA$=75 ORB%=13 ORB%=10 OR E
OF#Z:
 1880 A$(Q%)=A$+STRING$(75-LENA$," "):Q%
=Q%+1:
 1890 UNTILEOF#Z ORQ%>49
 1900 CLOSE#Z:PROCdis:ENDPROC
 1910 DEFPROCcen
 1920 H%=0:REPEAT:H%=H%+1:UNTILMID$(A$(V
POS+O%),H%,1)<>" ":I%=76:REPEAT:I%=I%-1:
UNTILMID$(A$(VPOS+O%),I%,1)<>" "
 1930 A$(VPOS+O%)=STRING$((75-LEN(MID$(A
$(VPOS+O%),H%,I%-H%)))/2," ")+MID$(A$(VP
OS+O%),H%,I%-H%+1):A$(VPOS+O%)=A$(VPOS+O
%)+STRING$(75-LEN(A$(VPOS+O%))," ")
 1940 C%=POS:D%=VPOS:PRINTTAB(0,VPOS);FN
col(VPOS+O%+1);A$(VPOS+O%);TAB(C%,D%);:E
NDPROC
 1950 DEFPROCedit
 1960 H%=FNnum("Page"):IFH%=0 OR H%>pg%-
1:PROCrul:ENDPROC
 1970 Z%=OPENUP"TEXT":IFH%=1 ptr%=&200:E
LSE PTR#Z%=2*H%-3:ptr%=BGET#Z%:ptr%=ptr%
*256+BGET#Z%:ptr%=ptr%*77+&200
 1980 E%=1:P%=ptr%:PTR#Z%=P%:INPUT#Z%,pg
%,sz%,kw$,ti$:PTR#Z%=P%+154:FORA%=0 TO s
z%-1:INPUT#Z%,A$(A%):NEXT:CLOSE#Z%:bal%=
FNlast:PROCdis:ENDPROC
 1990 DEFPROCmove
 2000 H%=FNnum("Top boundary"):I%=FNnum(
"Bottom  boundary"):J%=FNnum("Insert at"
):IFH%*I%*J%=0:PROCrul:ENDPROC
 2010 B%=I%-H%:IFJ%+B%>50:H%=FNnum("Too
low, hit any key to abort"):ENDPROC
 2020 IFJ%<H%:FORA%=H%TOI%:A$=A$(I%-1):F
ORB%=I%TOJ%+1STEP-1:A$(B%-1)=A$(B%-2):NE
XT:A$(J%-1)=A$:NEXT
 2030 IFJ%>I%:FORA%=H%TOI%:A$=A$(H%-1):F
ORB%=H%TOJ%-1:A$(B%-1)=A$(B%):NEXT:A$(J%
-1)=A$:NEXT
 2040 PROCdis:ENDPROC
 2050 DEFPROCdelblk
 2060 H%=FNnum("Top boundary"):I%=FNnum(
"Bottom  boundary"):J%=FNnum("PRESS 0 to
 abort"):IFH%*I%*J%=0:PROCrul:ENDPROC
 2070 FORA%=H%-1TOI%-1:A$(A%)=STRING$(75
," "):NEXT:PROCdis:ENDPROC
 2080 DEFPROCdis
 2090 LOCAL A:C%=POS:D%=VPOS:PROChd:PROC
rul:PROCval:PRINTTAB(0,3);
 2100 FORA=O%+3TOO%+23
 2110 PRINTFNcol(A+1);A$(A);
 2120 NEXT:VDU31,C%,D%:ENDPROC
 2130 DEFPROChd
 2140 PRINTTAB(0,0)"KEYWDS <";SPC60;TAB(
8,0);kw$;TAB(68,0)"> PAGE:";pg%;TAB(9,1)
;"TITLE :"ti$;TAB(68,1);"Bal :";bal%-FNl
ast;" "
 2150 ENDPROC
 2160 DEFPROCrul
 2170 PRINTTAB(5,2)STRING$(15,"====:");T
AB(0,24)STRING$(79,"=");TAB(U%,2)"T";TAB
(V%,2)"T";TAB(W%,2)"T";TAB(0,2)"line:"
 2180 ENDPROC
 2190 DEFPROCval
 2200 VDU31,0,1;:IFN%=1PRINT"INS-ON ";EL
SEPRINT"INS-OFF"
 2210 VDU31,30,24;:IFO%=26PRINT"BOTTOM
OF  PAGE";ELSEPRINTSTRING$(16,"=");
 2220 ENDPROC
 2230 DEFPROCkw
 2240 IFkw$=STRING$(60,"+") kw$="/"
 2250 REPEAT
 2260 J%=VPOS+O%:K%=POS-4
 2270 k%=ASC(MID$(A$(J%),K%,1))AND 95
 2280 IFk%>64 AND k%<91 kw$=kw$+CHR$k% E
LSEk%=0
 2290 C%=POS:D%=VPOS
 2300 VDU9
 2310 UNTILK%=75 OR k%=0
 2320 IFRIGHT$(kw$,1)<>"/" AND LENkw$>0
kw$=kw$+"/"
 2330 IFLENkw$>60 REPEAT:k%=INSTR(kw$,"/
",2):kw$=MID$(kw$,k%):UNTILLENkw$<61
 2340 IFkw$="/"kw$=STRING$(60,"+")
 2350 PROChd:VDU31,C%,D%:ENDPROC
 2360 DEFPROCup
 2370 IFO%=26THENENDPROC
 2380 O%=O%+1:C%=POS:D%=VPOS:VDU28,0,23,
79,3:PRINTTAB(0,20):VDU26:PRINTTAB(0,23)
;FNcol(O%+24);A$(O%+23);TAB(C%,D%);:ENDP
ROC
 2390 DEFPROCdn
 2400 IFO%=-3ENDPROC
 2410 O%=O%-1:C%=POS:D%=VPOS:VDU28,0,23,
79,3:PRINTTAB(0,0)CHR$11;FNcol(O%+4);A$(
O%+3):VDU26,31,C%,D%;:ENDPROC
 2420 DEFPROCmen:C%=POS:D%=VPOS:VDU22,7
 2430 PRINTTAB(15,3)"M E N U"'''"1:INPUT"
'"2:TITLE"'"3:SAVE"'"4:RESET"'"5:EXIT"
 2440 REPEAT:M%=GET:UNTIL(M%>48ANDM%<54)
 2450 IFM%=50PROCtit
 2460 IFM%=51PROCsa
 2470 IFM%=52PROCset
 2480 IFM%=53PROCend
 2490 VDU22,3:ENDPROC
 2500 DEFPROCsa:CLS:IFE%=1:Z%=OPENUP"tex
t":PTR#Z%=ptr%:PRINT#Z%,pg%,sz%,kw$,ti$:
PTR#Z%=ptr%+154:FORA%=0TOsz%-1:PRINT#Z%,
A$(A%):NEXT:CLOSE#Z%:PROCset:ENDPROC
 2510 IFFNlast=0 PRINT"There is no TEXT"
:Z%=INKEY(200):ENDPROC
 2520 IFLEFT$(kw$,1)="+"PRINT"There are
NO keywords!":Z%=INKEY(200):ENDPROC
 2530 IFLEFT$(ti$,1)="#"PRINT"There is N
O title! PRESS C to continue":Z%=GETAND9
5:IFZ%<>67:ENDPROC
 2540 sz%=FNlast:IFsz%>bal% PRINT"This p
age is too long to be the last one!":R%=
inkey(200):ENDPROC
 2550 Z%=OPENUP"text":ext%=EXT#Z%:PTR#Z%
=ptr%:PRINT#Z%,pg%,sz%,kw$,ti$:PTR#Z%=pt
r%+154
 2560 FORA=0TOFNlast-1:PRINT#Z%,A$(A):NE
XT
 2570 newptr%=PTR#Z%:newptr%=(newptr%-&2
00)/77
 2580 PTR#Z%=0:BPUT#Z%,pg%
 2590 PTR#Z%=index%:BPUT#Z%,(newptr% DIV
 256):BPUT#Z%,(newptr% MOD 256)
 2600 CLOSE#Z%:ENDPROC
 2610 DEFPROCtit:VDU28,0,15,79,13:PRINTT
AB(0,0)"Enter the title here":INPUTLINET
AB(0,1)ti$:ti$=LEFT$(ti$,50):ENDPROC
 2620 DEFPROCend:CHAIN"TMENU":ENDPROC
 2630 CLOSE#0:PRINTERR;" at ";ERL:REPORT
:PRINTpr$:B%=GET:IFB%=13GOTO130ELSEPROCe
nd
```

HINTS *and tips* HINTS *and tips* HINTS *and tips* HINTS *and tips* HINTS *and tips*

This month's collection of hints contains more invaluable information for BBC micro users, with emphasis this month on the Master 128. Further hints and tips for this page are always welcome.

IMPROVED MASTER DATE DISPLAY

Peter Hopkins

The following routine will give a fuller version of the date held by the internal clock (fitted as standard on a Master 128). If the current date on the system is normally held in a variable called *TIME$* then line 1010 should be altered to:

```
   WATCH$=TIME$
```

and line 1170 omitted.

```
 1000 DEF FNdate
 1010 WATCH$=FNget_date
 1020 RESTORE 1150
 1030 REPEAT:READ M$
 1040 UNTIL MID$(WATCH$,8,3)=LEFT$(M$,3)
 1050 RESTORE 1160
 1060 REPEAT:READ D$
 1070 UNTIL LEFT$(WATCH$,3)=LEFT$(D$,3)
 1080 DAY=VAL(MID$(WATCH$,6,1))
 1090 DAY1=VAL(MID$(WATCH$,5,2))
 1100 IF DAY=0 OR DAY >3 ABB$="th"
 1110 IF DAY=1 ABB$="st"
 1120 IF DAY=2 ABB$="nd"
 1130 IF DAY=3 ABB$="rd"
 1140 =D$+" "+STR$(DAY1)+ABB$+" "+M$+
" "+MID$(WATCH$,12,4)
  1150 DATA January,February,March,April,
May,June,July,August,September,October,
November,December
  1160 DATA Monday,Tuesday,Wednesday,
Thursday,Friday,Saturday,Sunday
  1170 DEF FNget_date:A%=14:X%=0:Y%=9:
CALL&FFF1:=$&900
```

ROMAN PAGE NUMBERS IN EDIT

Andrew Rowland

Page numbers can be printed in Edit (bundled in with the Master 128) using .r0. For example, a footer which prints the page number in the centre (the default) could be programmed using .fo as follows:

```
   .fo.ce
   Page .r0
   .en
```

But did you know that it can be in Roman numerals as well? The command .af0 n changes the format, where the possible values for 'n' are:

0	normal
8	capital Roman, e.g. I, II, V, etc.
9	lower case Roman, e.g. i, ii, v etc.

For example, the footer:

```
   .fo.af0 9.rf
   Introduction p. .r0
   .en
```

would print 'Introduction p. iv' at the bottom right of page four.

PRINTING SMALL TYPE

Peter Hopkins

If you have a need for printing in a small typeface, then set the printer into superscript and condensed mode, and also move the line spacing up:

```
 100 VDU2
 110 REM set superscript
 120 VDU1,27,1,83,1,0
 130 REM set condensed
 140 VDU1,15
 150 REM set line space to 7/72
 160 VDU1,27,1,65,1,49
 170 VDU3
```

Remember to reset your printer to its normal defaults when you have finished. The simplest way is to execute the 'Initialise printer' sequence:

```
   VDU2,1,27,1,64,3
```

I use this technique to produce audio cassette inlay cards for my own use.

Magscan

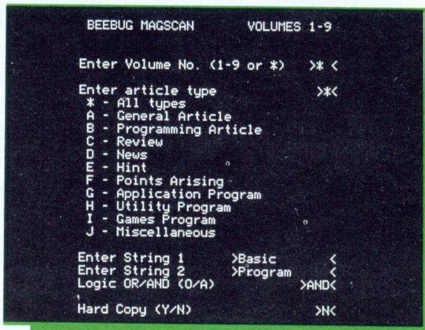
Comprehensive Magazine Database for the BBC Micro and the Master 128

An updated version of Magscan, which contains the complete indexes to all BEEBUG magazines from Volume 1 Issue 1 to the latest Volume 9 Issue 10

Magscan allows you to locate instantly all references to any chosen subject mentioned anywhere in the 90 issues of BEEBUG magazine.

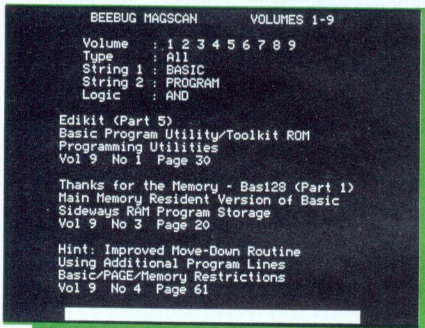
Just type in one or two descriptive words (using AND/OR logic), and you can find any article or program you need, together with a brief description and reference to the volume, issue and page numbers. You can also perform a search by article type and/or volume number.

The Magscan database can be easily updated to include future magazines. Annual updates are available from BEEBUG for existing Magscan users.



```
BEEBUG MAGSCAN          VOLUMES 1-9

Enter Volume No. (1-9 or *)    >* <

Enter article type              >*<
 * - All types
 A - General Article
 B - Programming Article
 C - Review
 D - News
 E - Hint
 F - Points Arising
 G - Application Program
 H - Utility Program
 I - Games Program
 J - Miscellaneous

Enter String 1       >Basic       <
Enter String 2       >Program     <
Logic OR/AND (O/A)              >AND<

Hard Copy (Y/N)                 >N<
```

Specifying a Magscan search



```
BEEBUG MAGSCAN        VOLUMES 1-9

 Volume    : 1 2 3 4 5 6 7 8 9
 Type      : All
 String 1  : BASIC
 String 2  : PROGRAM
 Logic     : AND

Edikit (Part 5)
Basic Program Utility/Toolkit ROM
Programming Utilities
Vol 9  No 1  Page 30

Thanks for the Memory - Bas128 (Part 1)
Main Memory Resident Version of Basic
Sideways RAM Program Storage
Vol 9  No 3  Page 20

Hint: Improved Move-Down Routine
Using Additional Program Lines
Basic/PAGE/Memory Restrictions
Vol 9  No 4  Page 61
```

Entries retrieved from Magscan files

Some of the features Magscan offers include:

- ◆ full access to all BEEBUG magazines
- ◆ rapid keyword search
- ◆ flexible search by volume number, article type and up to two keywords
- ◆ keyword entry with selectable AND/OR logic
- ◆ extensive on-screen help
- ◆ hard copy option
- ◆ easily updatable to include future magazines
- ◆ yearly updates available from BEEBUG

Phone your order now on (0727) 40303

or send your cheque/postal order to the address below. Please quote your name and membership number. When ordering by Access, Visa or Connect, please quote your card number and the expiry date.

Magscan complete pack, contains disc, manual and release notes: **£9.95**+p&p

Stock codes:	0005b	5.25"disc 40 track DFS	1457b	3.5" ADFS disc
	0006b	5.25"disc 80 track DFS		

Magscan update, contains disc and release notes: **£4.75** +p&p
(for update, please return old disc, label or evidence of purchase)

Stock codes:	0011a	5.25"disc 40 track DFS	1458a	3.5" ADFS disc
	0010a	5.25"disc 80 track DFS		

Postage: **a** - 60p UK, £1 Europe + Eire, £2.40 Elswhere **b** - £1.50 UK, £2.50 Europe + Eire, £4.80 Elswhere

BEEBUG Ltd, 117 Hatfield Road, St Albans, Herts AL1 4JS. Telephone (0727) 40303 Fax (0727) 860263

LETTER FROM LITHUANIA

A lot of thanks for your kind letter and copies of magazines. All the programs I found very useful. Also thank you for publishing my letter in the June issue of BEEBUG (Vol.10 No.2). I received five letters from BEEBUG readers in response to my first letter in BEEBUG (Vol.9 No.10)., and I now have all BEEBUG issues from volume two. I can hardly express my thanks to BEEBUG, and to Mr.Vincenzi who kindly paid for a subscription to BEEBUG for me. Also my thanks to all those BEEBUG readers who replied; unfortunately the post in Moscow works so poorly that many letters to Lithuania are being lost there.

Kriukelis Saulius

As always, BEEBUG readers have shown themselves very willing to help another BBC micro user.

BEEBUG BEST FOR BBC USERS

Previously I subscribed to "The Micro User", and although it used to be very BBC micro oriented it is now centred around the Archimedes. Thus I appreciate your efforts in providing an informative and lively magazine for the Beeb. However, I am slightly disappointed with the lack of games features - even a couple or so pages devoted to hints and tips would be appreciated.

Rowland Fraser

For some time now, BEEBUG has been the only magazine which continues to be entirely devoted to the needs of users of the BBC micro and Master series. Regarding games, it is not so many years ago that we regularly had readers complaining that we provided too much coverage, and we thus reduced the games content of the magazine accordingly. More recently we have started publishing more games programs, and games reviews (as in Vol.10 No.2), and we will continue to feature these in the future. What do other readers think?

KEYBOARD/JOYSTICK OPTION NEEDED

Could you please consider publishing a simple program which could easily be incorporated in published programs to enable the choice of keyboard or joysticks for controlling the movement/firing of objects/sprites on the screen. I had in mind a line, say:

`IF X THEN move=move+1 OR IF ADVAL etc.THEN move=move+1` (move joystick right) and similarly for other functions. I am sure that many BEEBUG readers would welcome such a program.

R.R.Boyce

What Mr.Boyce seeks is not too difficult if it can be built in to a program, and those programs which provide a choice of keyboard or joystick control do just that. To provide a general solution is more complex, and I suspect that it is not practical to cover ALL programs. If any reader feels able to meet the challenge, even partially, then we would be pleased to consider the result for publication.

USING VIEW FOR MAGSCAN FILES

Not many weeks ago I purchased MagScan from you. A very useful tool it is too, and I am very pleased with it. Wanting to keep the index up to date, I looked at the hex dumps of the files Vol1 to Vol9b and came to the conclusion that they were straightforward View files, but some of the supplied MagScan files (not all) refuse to load. What is the reason?

Arthur Adams

MagScan files are not View files as such, but all MagScan files can be loaded into View by using NEW and then the READ command. To save a View file for use by MagScan set markers at the start and end of the file and use the WRITE command, e.g.: `WRITE Vol10 1 2`

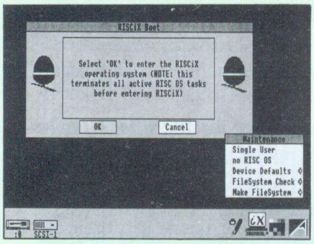
RISC USER

The Archimedes Magazine & Support Group

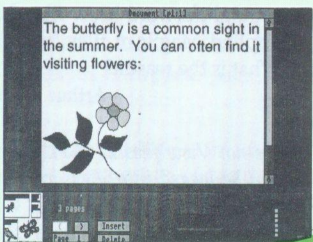
Now in its fourth year of publication, RISC User continues to enjoy the largest circulation of any magazine devoted solely to the Archimedes range of computers. It provides support for all Archimedes users at work (schools, colleges, universities, industry, government establishments) and home.

Existing Beebug members, interested in the new range of Acorn micros, may either transfer their membership to the new magazine or extend their subscription to include both magazines.

A joint subscription will enable you to keep completely up-to-date with all innovations and the latest information from Acorn and other suppliers on the complete range of BBC micros. RISC User has a massive amount to offer to enthusiasts and professionals at all levels.


RISCiX Boot

Select 'OK' to enter the RISCiX operating system (NOTE: this terminates all active RISC OS tasks before entering RISCiX)


The butterfly is a common sight in the summer. You can often find it visiting flowers:

Here are some articles and series published in the most recent issues of RISC User:

UNIX FOR THE ARC
It is now possible to upgrade 400/1 and 500 series Archimedes to UNIX. An article about installing and running UNIX on the Arc.

PICTURE KIT
A look into this exciting new development for the Archimedes, which allows you to take photographs with Canon's Ion camera (which stores the image on disc), view them instantly on your computer screen and print them in colour or black and white.

DRAWFONT
A powerful application which provides the means to manipulate outline fonts in Draw.

DESKTOP FOLIO
A review of this colourful and sophisticated package, which combines a wordprocessor, basic DTP and interactive publishing in an easy-to-use system, primarily designed for the education market.

FORM DESIGNER
An excellent application which makes the creation of printed forms and tables very easy indeed.

MULTI-MODE SPRITE VIEWER
A program, which demonstrates the use of colour translation between different screen modes, and can be used as a stand-alone application for displaying your own sprites

VIRUS ALERT
An extremely useful piece of software which will automatically warn you if your machine catches a virus.

USING ANSI C
A series of articles on programming Desktop applications in C.

WP/DTP
A regular column which offers advice on using different DTP and WP packages.

ARCADE
An occasional column offering a round-up of the latest games for the Archimedes.

INTO THE ARC
A regular series for beginners.

SUBSCRIPTION DETAILS

As a member of BEEBUG you may extend your subscription to include RISC User for only:

Destination	Additional Cost
UK,BFPO &Ch Is	£ 9.10
Rest of Europe and Eire	£14.00
Middle East	£17.00
Americas an d Africa	£19.00
Elsewhere	£20.00

Personal Ads

BEEBUG members may advertise unwanted computer hardware and software through personal ads (including 'wants') in BEEBUG. These are completely free of charge but please keep your ad as short as possible. Although we will try to include all ads received, we reserve the right to edit or reject any if necessary. Any ads which cannot be accommodated in one issue will be held over to the next, so please advise us if you do not wish us to do this. We will accept adverts for software, but prospective purchasers should ensure that they always receive original copies including documentation to avoid any abuse of this facility.

We also accept members' Business Ads at the rate of 40p per word (inclusive of VAT) and these will be featured separately. Please send us all ads (personal and business) to MEMBERS' ADS, BEEBUG, 117 Hatfield Road, St. Albans, Herts AL1 4JS. The normal copy date for receipt of all ads will be the 5th of each month.

AMX Pagemaker for BBC and BBC+ with manual £25, System Delta database ROM & disc for BBC, BBC+, Master 128, Master Compact with manual £25, Acorn teletext adaptor for BBC, BBC+, Master with Advanced teletext ROM £35, Acorn DNFS ROM £10, 32k/4Mb Solidisk ROM/RAM expansion board (needs attention) £10. Tel. (0934) 742071 after 6.30pm.

WANTED: View Printer driver generator on 3.5" disc with instructions, for BBC Master Compact. Tel. (0763) 208365.

WANTED: Replay system Mk2 or tape to disc for BBC B. Tel. (0691) 655818 after 6pm.

Acorn User magazines issue 1 to Oct '89 complete. Offers? Tel. (0634) 727312.

The following are now surplus to my requirements; Mini Office II for Master £8, dust cover for BBC B £2.50, Fax*File organiser software £5, Care Master dual ROM cartridge £5, W.E. operating manual for DFS £3.50, 5 games on tape for BBC B all originals the lot £2.50m AMX Max ROM £10, all software comes complete with manuals etc. Tel. 061-442 5158.

WANTED: Three button Tracker-Ball or similar device for BBC B user port together with suitable soft-ROM-ware, only cursor and button programming really needed. Tel. (0304) 368644.

WANTED: Sideways ROM socket board for BBC B. Tel. (0935) 21301.

M128 computer, View and Viewsheet manuals and one ROM cartridge £260, Philips 8833 colour monitor £150, dual 5.25" 40/80T drives in plinth £130, Morley AA ROM board £27, 3.5" drive £40, JUKI 6100 daisywheel printer £120, Datachat 1223 modem with BBC lead and software £35, W.E. printer monitor ROM £7, AMX Art £15, Exmon II £15, all excellent and complete (upgraded). Acorn DTP for Archimedes £80. Tel. 051-606 0289.

Watford Apollo modem for BBC B or Master, four speed, complete with comms ROM and manual £25, also Fleet Street Editor Program complete in 80T £15, and approx 100 used but good 5.25" discs £5 plus post. Tel. (0795) 429521.

BBC B issue 4, 7 manuals £60. Write to; Mr G Joynson, 351 Old Chester Road, Birkenhead, Merseyside L42 2DT.

M128 with 3.5" DD and printer, Interword, Edit, View, Viewsheet, Dumpmaster, DFS, ADFS, Welcome Guide, both reference manuals £320 plus carriage or buyer collects. Tel. (0395) 263638.

Epson RX 80F/T printer in complete working order, sale due to upgrade, includes power cord, 1 metre cable for connection to model B or Master, dust cover, official operation manual and three spare ribbons £75. Tel. (0442) 259054.

HELP WANTED: BEEBAID's disc indexing system disc and its manual by Jay Soft, also a sample of a disc saved for the downloading of share prices from teletext and data disc ready for transfer to Sharemaster. I need to convert data saved from Commstar page saved, this is in conjunction with Sharemaster Teletext Update Module by Winrow Dataspeed. We do not have teletext here. Please write to; Mr R Tyrer, 1 conifer Road, Tokai, 7945, South Africa or fax 021 728354, will pay postage and for discs.

WANTED: details and/or circuit of the Master Compact RS232 interface. Write to; 1 Lees Farm Drive, Madeley, Telford, Shropshire, TF7 5SU.

100 BBC B tapes, games, utilites all originals many boxed, value over £300 sell the lot for £40. Tel. (0392) 874964.

BBC B issue 7 without DFS but fitted with ECONET interface good clean condition £125 o.n.o. Tel. (0621) 816304 after 7pm.

512 co-processor mounted in Universal 2nd processor complete with discs, mouse, user guide etc. In mint condition £180. Tel. (0252) 540396.

WANTED: Gemini's BEEBCALC ie LOAD BCALC either on disc or ROM. Tel. (0745) 825036 anytime.

M128 £195, Philips colour monitor £95, W.E. dual 5.25" DD in plinth with PSU £95, 3.5" DD £25, Music '87 synthesiser £15, Nightingale modem £15, AMX mouse £10, Marconi Trackerball £10, Micropulse ROMbox £30, Vine ROMboard3 £9, Morley Eprom Programmer £18, 6 ROM cartridges £4 each, Genie cartridge £20, C.C. Mega 3 £30, Interbase £25, Spellmaster £25, W.E. Wapping Editor £30, Quest paint £15, AMX Pagemaker and Superart (M128 discs) £10, MOS+ disc £3, Reference manuals 1&2 £12, View and Viewsheet guides £8, Sideways RAM User Guide £3, or the complete system for £690. Tel. (0332) 572009.

Master 512k, two 5.25" 40/80 switchable disc drives, green screen monitor, Brother M1009 printer, Overview, Interword, AMX Stop Press, BEEBUG C, some shareware, Dabs 512 User Guide etc. £485. Hybrid Music 5000 and 4000, Ample Programming Manual £200. Tel. (04243) 3606.

Mini Office II for BBC B/B+ (40T), includes all packaging and instructions £10. Tel. (0463) 83 658.

Morley Teletext receiver in excellent condition £35, full 144k Watford ROM/RAM board with all extras installed £35. Tel. (0532) 653643.

WANTED: Teletext adaptor by Acorn, preferably with ROM and manual. Tel. (0634) 861627 eves or w/ends.

BEEBUG MEMBERSHIP

Send applications for membership renewals, membership queries and orders for back issues to the address below. All membership fees, including overseas, should be in pounds sterling drawn (for cheques) on a UK bank. Members may also subscribe to RISC User at a special reduced rate.

BEEBUG SUBSCRIPTION RATES

BEEBUG SUBSCRIPTION RATES		BEEBUG & RISC USER
£18.40	1 year (10 issues) UK, BFPO, Ch.I	£27.50
£27.50	Rest of Europe & Eire	£41.50
£33.50	Middle East	£50.50
£36.50	Americas & Africa	£55.50
£39.50	Elsewhere	£59.50

BACK ISSUE PRICES (per issue) 1 July 1991

Volume	Magazine	5"Disc	3.5"Disc
6	£1.00	£3.00	£3.00
7	£1.10	£3.50	£3.50
8	£1.30	£4.00	£4.00
9	£1.60	£4.75	£4.75
10	£1.90	£4.75	£4.75
Binders	£4.20		

All overseas items are sent airmail. We will accept official UK orders for subscriptions and back issues, but please note that there will be a £1 handling charge for orders under £10 which require an invoice. Note that there is no VAT in magazines.

POST AND PACKING

Please add the cost of p&p when ordering individual items. See table opposite.

Destination	First Item	Second Item
UK, BFPO + Ch.I	£ 1.00	£ 0.50
Europe + Eire	£ 1.60	£ 0.80
Elsewhere	£ 2.40	£ 1.20

BEEBUG
117 Hatfield Road, St.Albans, Herts AL1 4JS
Tel. St.Albans (0727) 40303, FAX: (0727) 860263
Manned Mon-Fri 9am-5pm
(24hr Answerphone for Connect/Access/Visa orders and subscriptions)

BEEBUG MAGAZINE is produced by BEEBUG Ltd.

Editor: Mike Williams
Assistant Editor: Kristina Lucas
Technical Editor: Alan Wrigley
Technical Assistant: Glynn Clements
Production Assistant: Sheila Stoneman
Advertising: Sarah Shrive
Managing Editor: Sheridan Williams

CONTRIBUTING TO BEEBUG PROGRAMS AND ARTICLES

We are always seeking good quality articles and programs for publication in BEEBUG. All contributions used are paid for at up to £50 per page, but please give us warning of anything substantial that you intend to write. A leaflet 'Notes to Contributors' is available on receipt of an A5 (or larger) SAE.

Please submit your contributions on disc or cassette in machine readable form, using "View", "Wordwise" or other means, but please ensure an adequate written description is also included. If you use cassette, please include a backup copy at 300 baud.

In all communication, please quote your membership number.

BEEBUG Ltd (c) 1991
ISSN - 0263 - 7561
Printed by Arlon Printers (0923) 268328

Magazine Disc

August/September 1991
DISC CONTENTS

TEXBASE: A FREE FORMAT INFORMATION SYSTEM - the first two of a suite of programs which create a free format database: *TMenu* creates the boot file and *Enter* allows you to enter text. Next month we will add the searching and deleting routines.

PRACTICAL WAYS WITH GRAPHICS - three programs which demonstrate how to transfer graphics to the computer screen: Photo (on-screen image of a photograph), Grid (prints out a grid) and GFL (plots a floor plan).

MULTIPLE WINDOWS - a utility which allows you to access up to 10 different predefined text windows on the screen.

WORDWISE USER'S NOTEBOOK - four segment programs for users of Wordwise Plus, which allow you to preview text, delete line functions and perform search and replace routines.

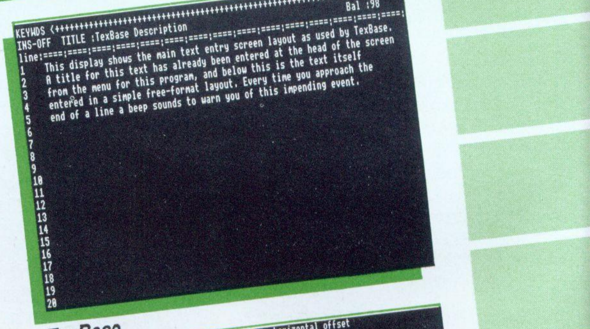
DFS OSWORD BUG - a short utility which overcomes the problem with the OSWORD bug in DFS version 2.45, supplied in the new Master ROM.

BEEBUG WORKSHOP: RANDOM SAMPLING - a program which demonstrates random sampling from a normal distribution.

USING PC DISCS ON A BEEB: UPDATE - updated versions of three useful programs which allow you to use PC discs on the BEEB: MS-DOS to BBC Transfer, BBC to MS-DOS Transfer, PC Formatting with Date/Time Stamping.

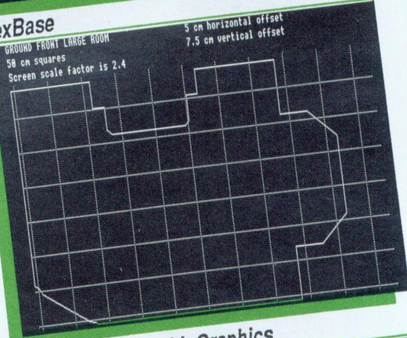
BEEBUG FUNCTION/PROCEDURE LIBRARY (5) - a varied collection of functions and procedures to add to your library.

MAGSCAN DATA - bibliography for this issue (Vol.10 No.4).



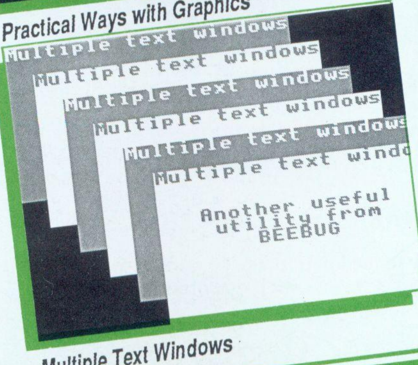
```
       <+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++> PAGE:1
                                                                  Bal :98
KEYWDS <++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
INS-OFF  TITLE :TexBase Description
line:====;====;====;====;====;====;====;====;====;====;====;====;====;====;
   This display shows the main text entry screen layout as used by TexBase.
   A title for this text has already been entered at the head of the screen
   from the menu for this program, and below this is the text itself
   entered in a simple free-format layout. Every time you approach the
   end of a line a beep sounds to warn you of this impending event.
```

TexBase



```
GROUND FRONT LARGE ROOM              5 cm horizontal offset
50 cm squares                        7.5 cm vertical offset
Screen scale factor is 2.4
```

Practical Ways with Graphics



```
Multiple text windows
 Multiple text windows
  Multiple text windows
   Multiple text windows
    Multiple text windows
     Multiple text windows
      Multiple text windows

           Another useful
           utility from
              BEEBUG
```

Multiple Text Windows

ALL THIS FOR £4.75 (5.25" & 3.5" DISC) + 60P P&P (30P FOR EACH ADDITIONAL ITEM)
Back issues (5.25" disc since Vol.3 No.1, 3.5" disc since Vol.5 No.1) available at the same prices.

DISC (5.25" or 3.5") SUBSCRIPTION RATES	UK ONLY	OVERSEAS
6 months (5 issues)	£25.50	£30.00
12 months (10 issues)	£50.00	£56.00

Prices are inclusive of VAT and postage as applicable. Sterling only please.

BEEBUG, 117 Hatfield Road, St.Albans, Herts AL1 4JS

Special Offers to BEEBUG Members August/September 1991

BEEBUG'S OWN SOFTWARE

Code	Product	Members Price inc Vat	Code	Product	Members Price inc Vat
1407a	ASTAAD3 - 5" Disc (DFS)	5.95	PAG1a	Arcade Games (5.25" 40/80T)	5.95
1408a	ASTAAD3 - 3.5" Disc (ADFS)	5.95	PAG2a	Arcade Games (3.5")	5.95
1404a	Beebug Applics I - 5" Disc	4.00	PBG1a	Board Games (5.25" 40/80T)	5.95
1409a	Beebug Applics I - 3.5"Disc	4.00	PBG2a	Board Games (3.5")	5.95
1411a	Beebug Applics II - 5" Disc	4.00	1600a	Beebug magazine disc	4.75
1412a	Beebug Applics II - 3.5" Disc	4.00	0077b	C - Stand Alone Generator	14.56
1405a	Beebug Utilities - 5" Disc	4.00	0081b	Masterfile ADFS M128 80 T	16.86
1413a	Beebug Utilities - 3.5" Disc	4.00	0024b	Masterfile DFS 40 T	16.86
1450a	EdiKit 40/80 Track	5.75	0025b	Masterfile DFS 80 T	16.86
1451a	EdiKit EPROM	7.75	0074b	Beebug C 40 Track	45.21
1452a	EdiKit 3.5"	5.75	0075b	Beebug C 80 Track	45.21
0005b	Magscan Vol.1 - 8 40 Track	12.50	0084b	Command	29.88
0006b	Magscan Vol.1 - 8 80 Track	12.50	0073b	Command(Hayes compatible)	29.88
1457b	Magscan Vol.1 - 8 3.5" ADFS	12.50	0053b	Dumpmaster II	23.76
0011a	Magscan Update 40 track	4.75	0004b	Exmon II	24.52
0010a	Magscan Update 80 track	4.75	0087b	Master ROM	29.88
1458a	Magscan Update 3.5" ADFS	4.75	1421b	Beebug Binder	4.20

0% Finance

A3000 Learning Curve

For a limited period we are again able to offer 0% finance for the Acorn Learning Curve Package

As well as the A3000 computer, this package includes the ***1st Word Plus*** word processor, ***Genesis*** educational database and the ***PC Emulator*** to run DOS programs.

Also included is a ***video*** on how to set-up the computer and information on the National Curriculum.

Please phone for written details: 0727 40303

OTHER MEMBERS' OFFERS

These offers are available for a limited period only. while stocks last. Orders are dispatched on a first come first served basis. To order phone 0727 40303.

COMPLETE BUSINESS PACK FOR THE MASTER 128

OverView

offers a database, spellchecker, index creation utility and a business graphics package

Members Offer £22.35 inc VAT + p&p

Normal price £93.60 inc. VAT

OverView from Acorn combines all View family programs into one package, and complements View and ViewSheet supplied with the Master 128.

- OverView adds **ViewStore, ViewSpell, ViewPlot, ViewIndex** and the **Printer Driver Generator.**
- The ROMs are supplied in an Acorn ROM cartridge.
- Switch between packages without having to save files.
- The postage cost reflects the size/weight of the package.

Stock code 1020e

Post & Packing	UK	Europe	Americas, Africa Middle East	Elsewhere
a	£ 1.00	£ 1.60	£ 2.40	£ 2.60
b	£ 2.00	£ 3.00	£ 5.00	£ 5.50
c	£ 3.10	£ 6.50	£10.50	£11.50
d	£ 3.60	£15.50	£24.50	£25.50
e	£ 6.00	£15.50	£24.50	£25.50