

Present

£28.00.  
(incl. VAT)



The 'FOUR IN ONE' ROM  
that adds FORTY NEW COMMANDS  
to BBC 'B' and ELECTRON Computers

GRAPHICS, LOGO GRAPHICS  
TOOLKIT, GENERAL PURPOSE

ADDCOMM is a new advance in upgrading the BBC Microcomputer. The software held within the EPROM has been specially designed to simulate a SUPER EXPANDED BASIC. This new advance eliminates the untidy 'Star' commands necessary with other upgrades and its exceptional versatility allows the use of any real number, variable or expression within the statements.

ADDCOMM's commands have been selected to give as great a value for money as possible, being so versatile they negate the need to buy several different chips and offer a cross-section of adaptable statements that have not been provided in BBC BASIC.

# Reprinted from "The Micro User" Oct. '84

## Is it time to extend your BBC Basic?

Addcomm

Vine Micros

AS the name suggests, the Addcomm ROM chip adds some new commands to the BBC Micro. I was astonished on first typing \*HELP ADDCOMM to see more than two complete screenfuls of new commands and detailed syntax.

In all, 40 extra Basic-like commands are available for use in direct mode or, in most cases, within programs – even in multistatement lines.

The new commands may be divided into three main groups – Graphics (11 commands), Logo/turtle graphics (8 commands) and Utilities (21 commands).

A general impression of the extra functions available may be gleaned from Tables I, II and III.

The combination of a variety of basic toolkit type utilities on the same chip as extended graphics commands is very useful, especially if your sideways ROM slots are at a premium, such as if you do not have a ROM extension board fitted or are an avid chip collector!

The review version was 1.00 and the documentation provided was a pre-release of the inevitable "manual in preparation".

However even this pre-manual covered 15 sides of A4 and was quite adequate to explore the new commands.

Inevitably a few mistakes and some areas of insufficient clarity were present, though I'm sure these will be cleared up in the final manual.

Fitting the chip was straightforward – it simply plugs into a sideways ROM socket – though remember a 1.00 or greater operating system must be present to use it.

The Addcomm chip is fussy as to where it sits, a fact not mentioned in the

supplied documentation.

In a disc-based system unless it occupied a slot between Basic and the DFS a number of commands were ignored. Moving the chip to the socket directly below (to the left of) Basic solved all the problems.

Addcomm seemed to be fully compatible with either the Acorn or Watford DFS (and according to the documentation, works with the Amcom DFS too).

Also, it did not present any incompatibility problems when used alongside 12 other popular sideways ROMs.

The addition of this chip does, however, reserve 256 bytes of work-space in user RAM and PAGE defaulted to &F00 (CFS) or &1A00 (DFS). This memory may be regained by switching off the chip (type Addcomm) and resetting PAGE or pressing Break – though this can be a nuisance after every use of Ctrl + Break (which switches Addcomm on again) and may lead to incompatibility with some software.

Surprise at the \*HELP information had turned to amazement by the time I had typed in and run the short program (Program I) shown below.

The use of SCALE – in this case making the x and y axes range from 0 to 100 – made it very easy to understand, remember and use the drawing

```
5 NODE 2
10 SCALE 0,100,0,100
20 CIRCLE 50,50,30
30 ELLIPSE 50,50,30,15
40 CFILL 1,3,3,1
50 FILL 50,50
```

Program I

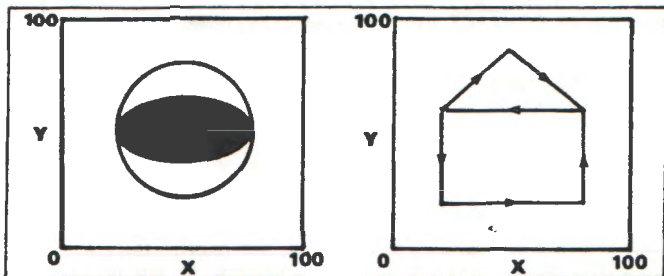


Figure 1

commands.

For example, line 20 draws a CIRCLE of radius 30 units starting at the centre of the screen ( $x=50, y=50$ ). This was much faster, especially in higher resolution modes, and much simpler to use than by employing trigonometrical functions such as SINES or COSINES.

The ELLIPSE command (line 30) was similarly useful. In addition, the syntax of both these commands allowed continuous or dotted lines, empty or filled shapes, or portions of the perimeters (ARCS) to be drawn.

But why the surprise? The filled in part of the drawing (Figure 1) was coloured orange – not one of the BBC colours normally provided. This was produced by a mixture of the logical colour red (1) and yellow (3) in the CFILL command.

You can therefore produce reasonable pinks, pale blues or on a black and white screen various types of shading. Furthermore, the FILL command acts like some of the extra PLOT functions (72-79, 88-85) available in Basic II, whereby extremely complex shapes may be filled in simply by giving the coordinates of any point within the object boundary.

With the addition of ready to use ROTATE and TRANSLocate commands BBC graphics become child's play.

The commands work like Basic and allow for the full passing of Basic variables – very useful in loops.

Implementation of the visual aspects of Logo provides eight commands which enable the user to move an invisible cursor around the screen drawing lines of specified length and

direction.

The commands are designed to be used in conjunction with the extended graphics facilities of Addcomm (SCALE, ROTATE, TRANS) and allow you to simulate the step by step building up of drawings or geometrical figures.

On my first attempt I experienced some difficulty using these commands and moved off in unpredictable directions, but with a little practice and revision of my geometry I was soon able to move in the direction I wished.

It would have been much more helpful – and educational – if a visible arrow-like cursor was available by choice when trying out routines.

Program II illustrates some of the features of Addcomm Logo in action drawing a simple house-shaped figure.

A fair range of useful utilities are available in Addcomm to assist in Basic program development (Table III).

Many of these commands (MEM, FKEYS, FIND, KILLREM) can even be used within programs, though it may not always be advisable.

The commands I used most were those which gave information about the program – MEM, FKEYS, LVAR, and FIND.

MEM, as you might expect, indicates the amount of memory used by the program, the amount reserved for variables (after a program run), and the amount of free memory. Values are displayed in both decimal and hexadecimal.

This command is similar to the \*FREE in Beebugsoft's Toolkit ROM but doesn't give details about PAGE, TOP, LOMEM and HIMEM.

## LOGO COMMANDS

PEN	Defines type of plot, for example continuous or dotted.
ADVANCE	Draws line for specified distance.
TURN	Changes direction of Logo cursor.
ANGLE	Initialises direction of Logo cursor.
LCIRCLE	Draws circle with Logo cursor as centre.
LELLIPSE	Draws ellipse with Logo cursor as centre.
LMOVE	Moves Logo cursor to specified position.
LPOS	Returns position of cursor.

Table I

## GRAPHICS COMMANDS

SCALE	Sets own screen scale for graphic commands.
UNSCALE	Reverts to normal BBC screen coordinates.
CIRCLE	Draws circle of specified size and position.
ELLIPSE	Draws ellipse of specified size and position.
ROTATE	Rotates figure by specified angle.
TRANS	Translocate figure to new position.
SMOVE	As for Basic MOVE but uses set scale.
SDRAW	As for Basic DRAW but uses set scale.
SPLOT	As for Basic PLOT but uses set scale.
FILL	Fills in colour in desired position.
CFILL	Mixes required colour.

Table II

```

5 MODE 4
10 SCALE 0,100,0,100
20 ROTATE 0,0,0
30 TRANS 0,0
40 ANGLE 270
50 LMOVE 20,60
60 FOR sides = 1 TO 2
70 ADVANCE 40
80 TURN 90
90 ADVANCE 60
100 TURN 90
110 NEXT sides
120 TURN 130
130 ADVANCE 39
140 TURN 280
150 ADVANCE 39
    
```

Program II

## HARDWARE REVIEW

### From Page 121

Ideally a combination of all the facilities of MEM and \*FREE would be useful for those really greedy for maximum information.

FKEYS lists the details of any previously defined function keys and is similar to a command available on many other utility chips.

All variables used in a program, except the resident integer variables A%-Z%, or variables beginning with specific characters may be listed using LVAR. Unfortunately its usefulness is somewhat limited as the current values of the variables are not given.

Probably the most often needed aid when programming in Basic is one to remind you where things are located.

FIND enables the program developer to locate strings (individual characters or whole words), variable names or Basic keywords anywhere in a program or in a selected part of a program by highlighting their occurrences in colour.

Following this of course there are the facilities for replacing the found characters either globally or selectively with the GREPL or SREPL commands.

But beware! Injudicious global replacing of Basic keywords may lead to disaster!

The next two facilities are for saving memory by shortening the final debugged working program either whole or in part.

KILLREM simply removes REM

statements whereas COMPACT goes even further and intelligently joins program lines together creating multistatement lines where possible.

This command does not, however, remove unnecessary spaces from within program lines as does the \*PACK in Toolkit. If you have both the Addcomm and Toolkit chips in your machine at the same time you could really blitz a program by using KILLREM, COMPACT and \*PACK in succession.

Incidentally, to improve subsequent readability of individual multistatement lines, lines can be split up again using Addcomm's LLIST option. This command may also be used within a program.

Two other useful commands are GOODPROG, which usually manages to fix a partly corrupted program, and VERIFY, which checks that the program saved to tape or disc is identical to the one currently in memory.

A number of other miscellaneous commands are also available on the chip. To aid graphics programming CHAR assists in the creation and editing of user defined characters and gives the codes required for VDU 23. This is much easier than having to work them out with pen and paper.

In addition, up to six separate windows may be set with SETWIN and selected with WIN. There is also a labelled GOTO facility - LGOTO - which jumps to a line containing a

selected variable (lower case only).

I am not so sure this features adds much to structured programming as REPEAT...UNTIL and PROCs are usually sufficient.

Equally, I am also rather dubious about the use of the three commands POPFOR, POPGOS POPREP, which allow you to literally pop out of FOR...NEXT loops, nested GOSUBs or REPEAT...UNTIL loops rather than by exiting them properly. However it was interesting to be able to nest GOSUB 2000 times without getting an error message!

Finally, Addcomm has in store a Basic utility command which I haven't seen anywhere else for the BBC - SORT.

This command will sort the data within a single dimension string array into order very quickly. It was extremely useful, not least in teaching programs illustrating the use of arrays.

In addition to the 39 commands described so far OPT is available to modify highlighting, spacing, colour, etc, when using many of the other Addcomm commands.

Overall, the number of facilities offered on this chip seems to be good value for money. The extra graphics commands are especially worthwhile and at the same time the user is provided with a number of valuable Basic programming aids.

On the minus side however - an inevitable minus with a facility that tries to be a jack-of-all-trades - the lack of a visible cursor in Logo graphics and incomplete variable dump must be weighted against the most useful SORT command.

Installation of this chip alters PAGE by 256 bytes which may be a nuisance and, as the Addcomm commands are not tokenised as are Basic keywords, memory may be at a premium in long programs in the high resolution modes.

These are, however, only slight deductions from a very useful sideways ROM.

A major problem, as with all extra commands added in this way, will be transportability of programs to machines lacking this chip.

Addcomm, therefore, will have to suffice until BBC Basic eventually incorporates a similar set of commands.

In all, a noble attempt by the programmer, R.P.D. Mallett, which deserves success especially when you consider that he is only 17 years old!

Mike Mahon

### UTILITIES

ADDCOM	Switches Addcomm off.
MEM	Details memory usage.
FKEYS	Displays contents of function keys.
LVAR	Displays program variables used.
FIND	Indicates occurrences of characters in program.
GREPL	Global replace.
SREPL	Selective replace.
KILLREM	Removes REM statements.
COMPACT	Shortens program, creates multistatement lines.
LLIST	Separates a multistatement line in listing.
GOODPROG	Cures a bad program.
VERIFY	Verifies a program from tape or disc.
CHAR	Defines and edit character.
SETWIN	Creates up to six screen windows.
WIN	Selects screen window.
LGOTO	Labelled GOTO.
POPFOR	Pop out of FOR loop.
POPGOS	Pop out of repeated GOSUBs.
POPREP	Pop out of REPEAT loops.
SORT	Sorts a single dimension string array.
OPT	Provides options within other commands.

Table III

ADDCOMM IS NOT LIKE OTHER ROMS.....

.....IT IS THE ONLY ROM.....

TO OFFER TRUE EXTRA BASIC STATEMENTS.

ADDCOMM adds 40 new statements and commands to BBC BASIC.

ADDCOMM scores highly over other ROMs because:

1. ANY VARIABLE can be used, (integers, floating point reals, strings, even multi-dimension arrays).
  2. SCIENTIFIC EXPRESSIONS and functions can be used within the statements in exactly the same way as normal BASIC statements.
  3. The statements and commands need NONE OF THOSE UNTIDY STARS in front of them.
  4. MULTI-STATEMENT LINES can be used to save space.
- 

ADDCOMM is for use in the:-

BBC Microcomputer with BASIC 1 or 2,  
O.S. 1.2.

Electron Microcomputer fitted with  
ROM board.

ADDCOMM has been designed to be as compatible as possible with other software, including:-

Acorn DFS: Watford DFS: Amcom (Pace) DFS:  
Wordwise: Disc Doctor.

---

ADDCOMM may now be ordered in one of two versions. The more popular 'on' version where the extra commands are available from power-up:

OR

The 'off' version where ADDCOMM must be enabled before use.

\*FX 163 can be used to enable and disable the ROM, and even to change the position of memory used by ADDCOMM. These features are available on versions 1.3 and onwards

We also offer a re-programming service to change one version to another at a charge of £1.00 (incl.). This offer includes changing any version prior to 1.3.

## Graphics

SCALING allows charts and diagrams to be drawn easily and accurately. Scales can be ANY SIZE, right up to the limits of the computer's ability and allowing graphs to be 'ZOOMED' into revealing the tiniest detail. And that's not all .... CIRCLES, ELLIPSES, POLYGONS, etc. can be drawn ANY SIZE at ANY ANGLE using a combination of statements, which also allow ROTATION and TRANSLATION of graphics around the screen giving exceptional graphic capability.

Even a FAST FILL ROUTINE is incorporated that not only fills even the most complicated shapes, but can produce shades of colours effectively giving over 2 BILLION COMBINATIONS in Mode 2 !!!

SCALE is a powerful and totally flexible statement that allows the user to take control of the 'scaling' of the screen. Integers, real numbers and variables can be used when defining the screen co-ordinates.

SMOVE, SDRAW and SLOT are similar in concept to MOVE, DRAW and PLOT, the existing BBC BASIC statements. Through using the screen as set up by SCALE they take on a power and flexibility without losing any of the facilities of the existing commands.

CIRCLE is a general purpose statement that allows a multitude of shapes to be drawn using a single statement, including:- any regular polygon, circles, arcs, solid sectors, triangles, squares .... all can be drawn utilising any of the standard plot codes available on the BBC. CIRCLE is normally used with SCALE to make the statement as flexible and easy to use as possible.

ELLIPSE is a command to complement CIRCLE and allow for the drawing of any elliptically based shape.

CFILL and FILL are used generally to fill a previously drawn area, CFILL being used to set the colour/pattern/shade by the use of existing colour codes, FILL being used to 'fill' an area, a point within the area being defined and the 'colour' filling the area up to pre-drawn bounds.

ROTATE is used to 'rotate' the axes about a point and by an angle both defined by the ROTATE statement on a scaled screen.

TRANS is used to translate the scaled area across the screen by the use of cartesian co-ordinates, subsequent graphics commands making use of the now displaced scaled area.

UNSCALE is used to return the non-SCALED screen co-ordinates, of a position on a SCALED screen, mainly for use with the POINT statement in BBC BASIC.



## General Purpose

These include a FAST SORTING statement, various program loop deleters (to jump out of loops, etc.) and perhaps most important of all, the ability to set up and control UP TO 7 TEXT WINDOWS, with each window having its own independent cursor position. SPLIT LISTING, i.e. to make listings more readable by splitting up multi-statement lines. Plus a feature UNIQUE TO ADDCOMM .... JUMPING TO A LINE VIA A LABEL, so that instead of meaningless numbers you can use names when jumping from one line to another!

POPFOR, POPGOS and POPREP allow the user to jump out of FOR - NEXT loops, GOSUB - RETURN subroutines and REPEAT - UNTIL loops without disturbing the running of the program. A situation that would normally occur without the use of these statements, sometimes causing errors.

LLIST can be programmed, unlike LIST, and under the user's control multiple statement lines can be split up into their component parts allowing for greater ease of reading.

SETWIN and WIN allows the user to define, using SETWIN and display, using WIN, up to seven complete windows on the screen. The user being able to select and return to previously de-selected windows at will, the cursor maintaining its last used position within that window.

LGOTO is similar to the GOTO statement but more powerful in that it enables the user to jump to 'labelled lines' allowing for neater and more readable programmes.

SORT allows the user to sort all or part of a string array into alphabetical order using a single statement. Invaluable for data processing.

OPT is used to control the options available for various statements or commands allowing for a more flexible and constructive use of ADDCOMM.

ADDCOMM is the command used to turn ADDCOMM off to give virtual total compatibility with games etc.

\*HELP is the single 'star' command in ADDCOMM and is intended to help the user by providing an instant guide to the syntax of all or individual statements within the ADDCOMM command list.

## Logo Graphics

Enter the amazing world of Logo Graphics with this creative and fun method of drawing on the screen. The Logo statements provide the necessary routines that when combined with BBC BASIC and ADDCOMM'S enhanced graphics gives an EXCEPTIONAL LOGO GRAPHICS SYSTEM.

LMOVE is used to position the LOGO cursor, usually used prior to further LOGO statements.

ANGLE is used to set the intended direction of the LOGO cursor, the angle being described in degrees.

ADVANCE is used to move the cursor in the direction as set by ANGLE by the distance described in the ADVANCE statement.

TURN gives the cursor a new direction by turning it a certain number of degrees anti-clockwise from its previous direction.

PEN defines the kind of trail left by the LOGO turtle using the internal plot codes.

LCIRCLE and LELLIPSE are the LOGO equivalents of the similar enhanced graphics commands, the shape centre being the current LOGO cursor position.

LPOS is used to return the position of the LOGO cursor from a SCALED screen.

The p  
the  
vers:  
Capal

```
10 REM ADDCOMM's Toolkit
20 INPUT "What is your name ";A$
30 FOR loop=1 TO 10
40 PRINT "Hello ";A$
50 NEXT loop
```

```
>MEM
Program :0067      103
Reserved:0014      20
Spare      :4285 17029
>LVAR
```

This is a simple demonstration of ADDCOMM's user-friendly TOOLKIT commands, it even makes programming enjoyable when you have ADDCOMM to do all the work.

```
A$
loop
>KILLREM
>SREPL:A$:name$
  20 INPUT "What is your name ";G$
Replace Y/N ? Y
  40 PRINT "Hello ";G$
Replace Y/N ? Y
>COMPACT
  50
```

The screen dump t  
demonstrates some  
easy to use GRAPH  
No other ROM offe  
statements.

```
>LIST
  20 INPUT "What is your name ";name$:
FOR loop=1 TO 10: PRINT "Hello ";name$:
NEXT loop
>
```



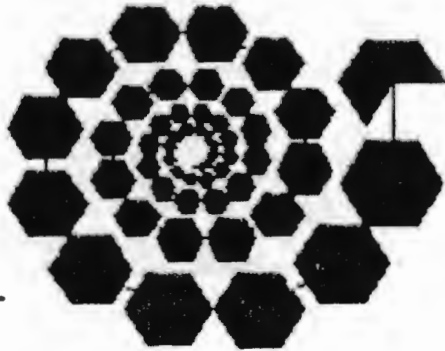
Escape at line 88.

>L.

```
10 MODE 1
20 SCALE 0,100,0,100
30 ANGLE 0
40 LMOVE 50,30
50 length=1
60 loop
70 ADVANCE length
80 LCIRCLE length/2,0,360,60,85
90 TURN30
100 length=length*1.05
110 LGOTO loop
```

>\_

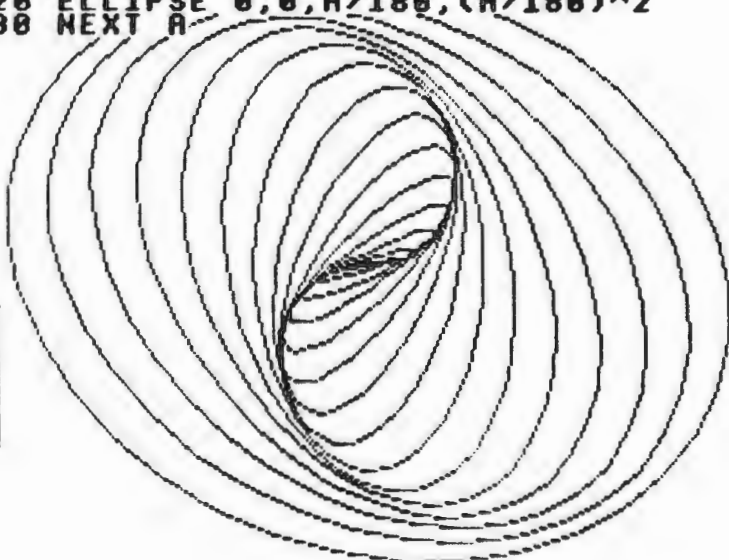
rogram and screen dump to  
ight show how simple and  
file ADDCOMM's LOGO GRAPHIC  
lity is to use.



>LIST

```
10 MODE 1
20 SCALE -1,1,-1,1
100 FOR A=10 TO 150 STEP 10
110 ROTATE 0,0,A
120 ELLIPSE 0,0,A/180,(A/180)^2
130 NEXT A
```

>\_



the right  
of ADDCOMM's  
C statements.  
s real BASIC

# Toolkit

ADDCOMM's toolkit commands offer the full feature set of routines that do the tedious work leaving you to get on with writing the program:

ANY variable/expression/statement/text can be used in its search and replace routines - not many ROMs can search and replace TOKENISED statements and commands. FULL USER CONTROL over the parts of a program changed, i.e. sections of a program (start and end line numbers), and even inside or outside of quotes. FULL HIGHLIGHTING (any screen mode), of found or replaced text. Lines appear in an easy to read split-up form and even if this is not desirable it can be turned off, (as can highlighting). INTELLIGENT program search algorithms ignore 'false' text (e.g. in the data that follows a GOTO statement).

Plus...an INTELLIGENT PROGRAM COMPACTER greatly reduces program size without affecting the running of the program. Also a clever function key lister capable of printing the EXACT DEFINITION e.g. '!';N ; and several more useful routines which include a character definer.

CHAR is a simplified 'character' designing facility using an enlarged grid that enables the user to quickly and easily construct new or changed graphic and pseudo alphanumeric characters.

COMPACT is used to reduce the size of all or part of an existing program by intelligently appending lines together to create longer but fewer lines in the program. The maximum line length being user defined.

FIND is used to search all or part of a program to find any desired character or set of characters, inside or outside quotation marks. The relevant lines being listed with the string highlighted.

GREPL and SREPL are two related commands that enable the user to replace a character or group of characters with any other character or group of characters completely within the control of the user, the same options being available as with FIND.

MEM returns to the display the amount of memory being used within BASIC, by the program, the variables and thus the available memory.

GOODPROG is a very useful last ditch command to enable the user to usually rescue 'BADPROGRAMS' and thus list an otherwise unlistable program.

VERIFY is a simple command to confirm that a copy to tape or disc is exactly the same as the original program in memory.

KILLREM intelligently deletes REM statements in all or part of the program as defined by the user.

LVAR is used to list all, or alphabetically selected, variable names currently within memory.

FKEYS is used to list to the screen the contents of all currently defined function keys enabling the user to quickly and easily re-define them.

# "ACORN USER" OCTOBER '84

## TOOLKITS ON TRIAL

Bruce Smith finds firm favourites among five utility ROMs

**T**HE ability of the BBC micro to accept a variety of extra ROM-based software that can be used concurrently with Basic or even replace Basic has always been an important aspect of its design, and it goes some way to compensating for the small amount of user memory available, especially in the high-resolution graphics modes. (With extra hardware, of course, the Electron has similar potential.)

Independent suppliers have been quick to jump onto the bandwagon of ROM-based software (normally termed firmware), producing a variety of programming goodies. As many of you are aware, the choice is confusing. What exactly does the firmware do? Is it as good as the others? If not, which one is the best? And so on.

In an effort to help you through this firmware jungle a comparative review of groups of firmware will be appearing in future issues of *Acorn User*. This month I start the ball rolling with a look at ROMs containing Basic programming aids or utilities. Future issues will focus on machine-code monitors, databases and wordprocessors.

### The Jargon

ROM-based software for the Beeb is often called a 'utility'. This word really means an aid to programming, a program that will allow you to perform a certain task with minimum fuss. Consider what would happen if you had a longish program in which you wanted to replace all occurrences of one variable name with another. No problem if the variable has been used only a few times. You simply edit it in using the cursor control and Copy keys. But what if there were ten, 20 or even 30 occurrences of the variable? To edit each one by hand would be laborious. The best answer is a utility program that would do the whole job for you. Enter the name to be replaced, the new name and the utility does the rest! Because of the usefulness of these types of commands a collection of them is generally termed a 'toolbox' or 'toolkit', as they are thought of as supplying a collection of tools for the programmer.

### A usable utility

The most obvious way to write a utility program is in Basic, and listing 1 provides an example that lists the values

assigned to all the resident integer variables in decimal and hexadecimal. Trouble is, writing a utility in this way has many drawbacks. First, where do you put it? The most obvious way is to incorporate it in the main program as a procedure, as demonstrated in the program. A library of utility procedures could be stored on tape or disc as ASCII files and simply spooled in as required. However, this eats up memory and means that the final program will need editing to remove the utility when it is complete—shades of Catch 22!

An alternative method would be to place the utility in a different part of

Executing machine code doesn't involve the complicated rigmarole of changing PAGE etc.—a simple CALL to the start address will suffice. Machine code also operates much faster than an interpretative language such as Basic.

Listing 2 provides a simple program compactor. Type it in, run it, and save the machine code to tape or disc using:

```
'SAVE 'SPACE' C00 + FF C00
```

Using the program simply involves 'RUNning it or, if already present, CALL &C00. The effect of the program is to remove all spaces, except those within quotes, from the program at PAGE so that it requires less memory space. This utility needs to be used with care. For example, the line:

```
500 DEFPROCdemo BYTE% = 0
```

is legitimate, but after deleting the space the line will be:

```
500 DEFPROCdemoBYTE% = 0
```

which will cause an error at run time. The way around this is to place a colon between the procedural name and the variable, thus:

```
500 DEFPROCdemo : BYTE% = 0
```

Similar care should be exercised when using the assembler to differentiate between a label name and the opcode to prevent them from turning into one long label name!

Using machine code in this way has its disadvantages — it still uses up memory. This may not be a problem if only one utility is required, but finding space for two or more useful utilities well out of the way is not easy. Also, once a machine code program has been assembled into a particular area of memory it can normally only be loaded back and used correctly in that same area, unless some fancy coding techniques have been employed to make it relocatable. Thus a program assembled to run in the soft character buffer is not much good if your program makes use of the definable character facility! A way round this would be to assemble code into several areas and load the required one in, but this is

## SUPPLIERS

### ADDCOMM

Vine Micros, Marshborough,  
Sandwich, Kent.  
Tel: (0304) 812278

### CARETAKER

Computer Concepts, Gadderden  
Place, Hemei Hempstead, Herts  
HP3 6EX.  
Tel: (0442) 63933

### TOOLKIT

Beebugsoft, PO Box 109, High  
Wycombe, Bucks HP10 8HQ.

### TOOLSTAR

Pace Software Supplies, 92 New  
Cross Street, Bradford BD5 8BS.  
Tel: (0274) 729306

### U-TOOLS

Intersoft, 26 Netherfield Road,  
Sandiacre, Nottingham NG10 5LN.  
Tel: (0602) 399974

memory by adjusting the value of PAGE. The program still uses up memory, assuming there's enough to put it there in the first place. To run the program you have to alter PAGE and change it back to its original setting to get back to the main program.

Another approach is to write the utility in machine code. The main disadvantage here is that you need a knowledge of assembler to do this yourself, unless the software is being bought. Machine code is more versatile than Basic and usually much less demanding of memory and so can be placed out of the way in a backwater of memory. The user-defined character space from &C00 to &CFF is a favourite location.

**Bruce Smith's example utilities in Basic and machine code (listings 1 and 2) are on yellow page 112**

fiddly and time-consuming, especially for tape users.

## Inside track

Home-brewed utilities therefore have their disadvantages - they have to be written, they use up precious memory and they are not always easy to use. The best compromise is to use machine code stored within EPROMs using the sideways ROM facility of the Beeb. The advantages are many. The utilities are always on hand because they form part of the micro itself, and that also makes them 'portable'; you are not dependent on disc drives or cassette recorders to get at them. Virtually all the commands are implemented as commands because the ROM itself contains an interpreter that recognises them. Thus our compacting program, if found in a toolbox, might be executed just by typing \*PACK or \*CRUNCH. Because ROMs occupy the same area as Basic they are transparent to the user, thanks to some clever electronics, and do not normally require any user memory.

There are disadvantages, of course. For a start, outlay - most items of firmware normally cost at least £22 and they need to be fitted into the machine. A dealer will often do this if you buy from him, but the mail order buyer is on his own.

## What's around

The only serious item of tape-based utility software I have encountered is BBCSoft's *Toolbox*. This is a neatly packaged book with accompanying cassette of programs. It obviously suffers from all the aforementioned problems and at £21 is not cheap. However, for anyone wishing to learn about the techniques involved in utility programming it is a useful introduction. Twenty-five utilities are provided, 13 of which are written in Basic and 12 in assembler. A good range of programs are provided and these include sorting routines, a variable dump, a REM stripper and program formatter. Each program is discussed in the accompanying text.

It has to be said, however, that EPROM-based software is really a must for the serious utility user.

**Addcomm** The *Addcomm* from Vine Micros is probably the best value for money out of the toolkits. Although it does not come out on top in table 2, the overall command content is undoubtedly the best. In addition to the 31 Basic programming utilities the EPROM provides some useful enhanced graphics facilities, including circle and ellipse drawing commands, colour fills and commands to scale and rotate your creations.

A simple implementation of Logo graphics is also included providing turtle graphics commands such as PEN, LMOVE, ROTATE, ANGLE and so on.

The commands are implemented not as operating system commands (ie, commands prefixed by asterisk) but instead they are entered directly as they are. This is because the *Addcomm* interpreter traps the errors that Basic would normally throw up and then tries to interpret the command that caused the error as an *Addcomm* command. This is good in theory but it can lead to confusion as it sometimes interprets Basic errors as *Addcomm* errors and displays the wrong error message.

Finally, *Addcomm* offers 40 commands to the user, though I consider only 37 of these usable. The other three allow the user to jump out of loops - naughty! They should be ignored by the user as they are not good programming practice. My plea to Vine Micros would be to delete these computer nasties from the instruction set and replace them with better error-handling.

**Caretaker** I'm a little disappointed with *Caretaker*. Over the past couple of years Computer Concepts has produced some of the best firmware for the BBC micro, but *Caretaker* falls some way short of the company's high standards. Of its 18 commands, I consider three to be space-fillers, although they are useful sometimes. CURSOR allows

the cursor to be turned on and off, while TABSTOPS and NOTAB allow the user to determine the new position of the cursor on the screen.

What disappoints is that it is rather run-of-the-mill, and shows none of the originality we've come to expect from CC. Useful enhancements, however, are those of single key command entry, emulating the Electron, as is the ability to save specified sections of program such as important procedures. *Caretaker* also scores on its EXCHANGE facility, which allows a good degree of control in selecting global or selective search and replace.

**Toolkit** Beebugsoft's *Toolkit* scores well in the five standard ratings section of table 2. The variable dump commands are extensive and include listing of procedure names and arrays in addition to integer and real variables. The enhanced line editor is useful, allowing program scrolling in any direction and listing of any erroneous line, highlighting the error at run time.

A couple of commands seem somewhat dubious, however. The ability to be able to perform OLD within a program would be needed on very few occasions, and I can think of no occasion when it has been necessary to perform NEW from within a program.

**Toolstar** This is not essentially a Basic toolkit but a general utilities aid, as it includes machine code and disc utili-

159

ITEM	ADDCOMM	CARETAKER	TOOLKIT	TOOLSTAR	U-TOOLS
PRICE	£28	£33.35	£27.00	£34.00	£26.95
No. Basic commands	31	18	26	8	26
Total No. commands	40	18	26	22	26
Bad program cure	●	●	●	●	●
Compact	●	●	●	●	●
Error-handling	●	●	●	●	●
Find	●	●	●	●	●
FKeys lister	●	●	●	●	●
Format listings	●	●	●	●	●
Global search/Replace	●	●	●	●	●
Graphics Enhancement	●	●	●	●	●
Help page	●	●	●	●	●
Insert program lines	●	●	●	●	●
PACK	●	●	●	●	●
Program status	●	●	●	●	●
Renumber	●	●	●	●	●
ROM command identity	●	●	●	●	●
Save/Load function keys	●	●	●	●	●
Save part program	●	●	●	●	●
Screen dump	●	●	●	●	●
Single key commands	●	●	●	●	●
Shift program	●	●	●	●	●
Turn off ROM	●	●	●	●	●
Variable dumps	●	●	●	●	●
Verify	●	●	●	●	●

Table 1. Facilities at your command

ties. As table 1 shows, only eight of its 22 commands are Basic-orientated and this should be borne in mind when it comes to your decision. In essence, *Toolstar* is suited to the advanced programmer who dabbles in machine code and Basic and uses discs.

The global search and replace facility is a good implementation and the toolkit's built-in software allows you to add your own RAM-based commands. A very interesting feature.

The *Toolstar* manual is excellent - by far the best I have seen produced for an item of commercial software; indeed it is much better than many computer user guides. Its 156 spiral bound pages are colour-coded, and copious use of examples makes it a joy to read.

**U-Tools** The *U-Tools* box offers a good, steady variety of utilities. One of the more interesting features is a command that will not only save a program but also lock it so that it can 'RUN, though it is a shame that it can be used efficiently only with machine code programs. A variety of program-crunching commands are on offer, plus an error-trapping wedge that lists an erroneous line before printing the error message.

	ADDCOMM	CARETAKER	TOOLKIT	TOOLSTAR	U-TOOLS
Ease of use	8	7	7	7	7
Range of utilities	8	7	6	5	7
Documentation	8	5	5	10	5
Error messages	5	8	9	8	6
<b>Five standard utilities ratings</b>					
Bad program core	7	7	8	8	7
FIND string	8	N/A*(5)	9	7	8
Global search & replace	8	9	8	9	7
Program compactor	9	9	7	N/A	9
Variable dumps	7	8	10	N/A	8
TOTAL	68	63	69	52	64

N/A = Not available.

\*FIND not directly available but can be implemented using global search and replace facility and specifying same search and replace strings.

Table 2. How they score on a scale of 1 to 10

*U-Tools* is the only toolbox to provide an intelligent screen dump of any of the graphics modes to an Epson or Epson-compatible printer such as Star. The dump provided is large - 23 x 16.5cm - and it is produced down the paper.

### Which one?

Before choosing which Basic toolbox you want to buy it is worth making out a list of items that you feel it should contain - in other words, the commands

that would be most useful to you.

My own preference is the *Addcomm*, followed by *Toolkit*, finances allowing. From table 1 it would seem that the best combination would be *Addcomm* and *Caretaker*, as they cover the entire range of utilities. However, these two ROMs seem to be incompatible as some *Addcomm* commands will not function with *Caretaker* present.

At the end of the day, however, it should be a personal choice.

ADDCOMM comes boxed complete with a User Guide and simple installation instructions. The 72 page User Guide is the handy A5 size and is ring-bound, the introductory pages take the new user through the additional statements and show by the use of various short programs the simple manner in which ADDCOMM benefits the programmer. Then follows in alphabetical order a description of each of the ADDCOMM commands, including syntax, comments and a sample program or line. Notes on compatibility, memory use, error messages and codes, with a list of abbreviations and summary of ADDCOMM statements and their syntaxes. There is also a short index.

Further comments from the Computer Media.

"Vine Micros" - their ADDCOMM chip vastly increases the power of the micro from the programmer's point of view. The result is truly as Vine describe it: a Super Extended BASIC. Enhanced graphics and turtle graphics are among the features offered.

"Efficient Chips"  
Education Equipment  
Supplement, Oct. 1984.

"The thing I like most about ADDCOMM is the fact that you don't need to use those messy OS commands, you can enter the commands just like BASIC statements in a program."

"ADDCOMM is a very useful and versatile piece of firmware and I would certainly recommend it to any BASIC programmer, it reduces the need to buy several ROMs as it contains almost all one could wish for."

"Marks: 10 out of 10."

"Let's hope other companies can give us such good value for money!"

VIEWFAX - Tubelink.

"ADDCOMM from Vine Micros: Highly recommended for graphics programming."  
Using the BBC Micro, No.1.

ROMS in preparation for the BBC Computer:

MATRIX ROM Provides BASIC statements for the manipulation of matrices; e.g. inversion, multiplication, etc.  
(Also Electron)

TAPE TO DISC ROM Simplifies the transfer of programs supplied on tape to the user's disc system.

ADDCOMM ORDER FORM

I enclose CHEQUE/POSTAL ORDER for £28.00. which includes V.A.T. and post and packing.

\*I require the on/off version of ADDCOMM.

Name.....

Address.....

.....

.....Post Code.....

\*Please delete where necessary.

The 'on' version (see page 5 of this brochure) will be sent where no stipulation is made.



