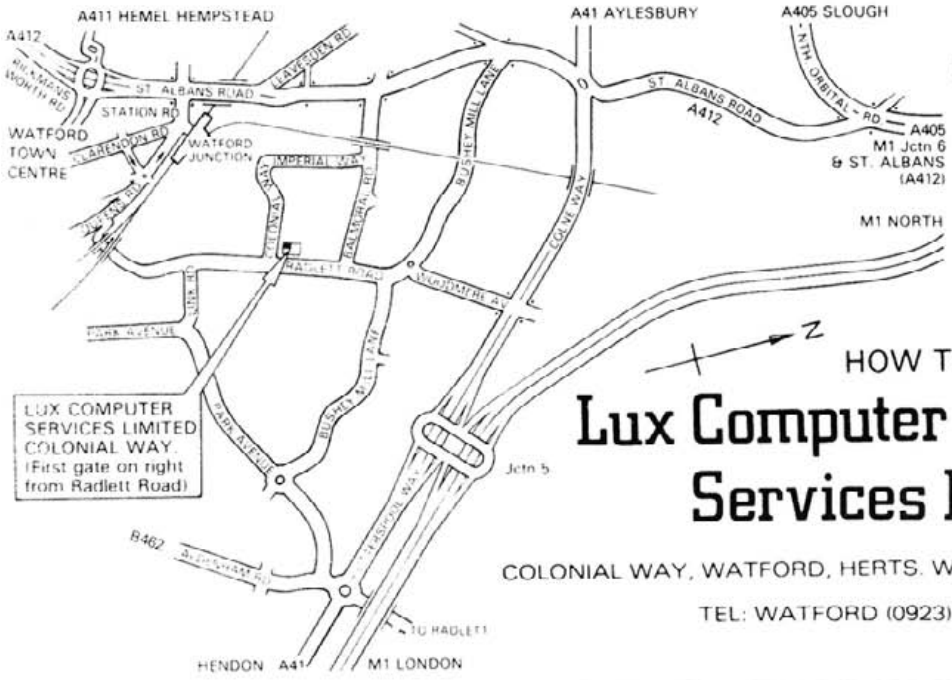
hard core

THE JOURNAL OF THE BRITISH APPLE SYSTEMS USER GROUP

April 1983 £1 VOLUME 3 No. 2



We've moved!

HOW TO FIND

Lux Computer Services Ltd

COLONIAL WAY, WATFORD, HERTS. WD2 4AT

TEL: WATFORD (0923) 47367

BASF DISKS

- individually numbered
- 5¼" single sided, single density
- suitable Apple II, IIe & ///
- box contains ten disks

1 BOX £ 15:50 NO

10 BOXES £140:00 EXTRAS

WE PAY VAT, POSTAGE & PACKING

Send order & remittance to:
SYSTEMICS LIMITED
21-23 THE BRIDGE
HARROW
MIDDLESEX HA3 5AG
01-863 0079

Access & Barclaycard accepted

ZAPPLE

A SOUND EFFECTS & MUSIC BOARD

- Built-in software means easy to use from Basic.
- Software included to easily create "Sound Tables".
- ALLOWS SOUND WITHOUT SLOWING GRAPHICS.
- Built-in amplifier gives volume control of all Apple sound output.
- 3 voices and noise, expandable to 9 voices.

Special introductory offer.
We pay the VAT, etc.
Send only £65.

Volume control and amplifier available separately. Send only £18.

Dealer enquiries welcome.

Meekrose Ltd.,
85 Waterdell, Leighton Buzzard, Bedfordshire LU7 8PL.
Tel: Leighton Buzzard (0525) 370621

MEEKROSE

hardcore

THE JOURNAL OF
THE BRITISH APPLE SYSTEMS USER GROUP
P.O.BOX 174 WATFORD WD2 6NF

Basug Committee

Norah Arnold	Chairman
John Sharp	Secretary
Fran Teo	Treasurer
Jim Panks	Membership Secretary
Bob Raikes	Events Organiser
John Rogers	Software Librarian
John Wellsman, John Martin, David Bolton	
EDITOR	Tony Williams
Assistants	Tony Game
	J. P. Williams
	Brigitte Dold
Advertising	Sheila Hirst

BANKERS

National Westminster Bank Ltd.,
Brompton Square, London SW3 1HH.

COPYRIGHT (C) - The contents of this journal are copyright of the British Apple Systems User Group and / or the respective authors. However permission will normally be granted for non-commercial reproduction by user groups affiliated to the International Apple Core, provided the author and source are properly credited.

The opinions and views expressed are those of the various contributors, and are not necessarily supported by the British Apple Systems User Group.

CONTENTS

4	Editorial	Tony Williams
	Chairman's Corner	Norah Arnold
	Lonely Apples	
5	Membership	Jim Panks
	S.I.G.'s	
	DOS 3.3 Upgrade Kits	
	The BCS	
7	Diversi-DOS	
13	Book Reviews	Jim Panks
14	Software Library	
15	So You Want To Write A Book?	Tony Williams
18	A Relocating Linker	Richard Teed
20	Local Groups	
21	The Linguist	Tony Williams
22	Visicalc	Frances Teo
26	Be An Apple Executive	Martin Rogers
30	Education	Norah Arnold
31	The Spelling Problem	Graham Davies
34	Readers' Letters	
36	Higher Text Extended	Peter Trinder
39	BASUG Disk No.46 Enhanced	D. J. Bullar
40	Workshops	John Sharp
41	Beginners Page	John Sharp
43	Seedlings	
44	What is MICRONET?	John Sharp
46	Diary	

INTERNATIONAL APPLE CORE TM

MEMBER OF THE INTERNATIONAL APPLE CORE

Cover Picture - Peter Trinder
See Higher Text Extended article in this issue.

Editorial

by Tony Williams

Just two weeks before this issue is put to bed your intrepid editor is taking what he considers a well earned rest, but he will not tell you where, since that would only make you unhappy. In his absence the job of pasting up the magazine has been manfully shouldered by Yvette, one of our most notable grass widows 1) because she was pressganged into the job and 2) because she has sheets of Letraset, both unbeatable qualifications. My thanks go to her in particular in advance. My pettifogging criticisms will only come later when I see the results (I still rather touchingly cherish the hope that I am indispensible despite the world's efforts to prove the contrary). Before I depart, however, I would like to draw your attention to two matters. One is the article on Diversi-Dos - which is not so much an article as a set of instructions for a rather remarkable contribution to the Software Library. This is a new approach by BASUG and your committee believes that where a product is outstanding and good value we are justified in putting BASUG behind it. The authors are confident that the honour system is a cost-effective way of distributing software, but whether it works depends entirely on you.

The second matter concerns a different but related point. The issue of piracy will not go away and is likely to get worse, affecting in first place the small games software author. BASUG members might like to know that Michael Blood is carrying out a fact-finding mission on behalf of the fledgling Computer Trade Association. If you have anything you want to tell him, write to him at: Abalon Hill Games, 650 High Road, North Finchley N12 ONL or call him on 01-445 3044. Incidentally, if any members should recall an article I wrote in early 1982 called "Ripoff News" offering to act as a clearing house for information about software piracy, you might be interested to know that the response from members has been nil, yes, not a single response has come in. Let's hope that Michael Blood gets further in his researches.

Chairman's Corner

by Norah Arnold

'Any ideas for Chairman's Corner?' I enquired just prior to the last national committee meeting.

'Write about how things have changed since BASUG began', I was told. 'Everything is so much more orientated towards business use now, whereas it used to be all for the hobbyist.'

Many new machines have appeared which are directed towards business use, Lisa for instance. Many small businesses use micros for various purposes. The amount of business applications software has increased gradually. But are things changing as fast as we think they are?

I had a look at Pete and Pam's catalog for September 1982, in which the software was classified by use. There were almost two hundred and fifty games on offer, with an average price of about seventeen pounds. About one hundred and twenty-five business programs were listed, but the average price was much higher, probably around one hundred pounds.

Next I turned to the data base of the Herts local group. Sixty per cent had described themselves as hobbyists of one kind or another. Twenty-eight per cent were interested in using Apples for both business and hobby, while only eight per cent had specified business use only. The other four per cent had not indicated where their interests lay.

The Herts group may be unrepresentative of BASUG as a whole so one should not read too much into this. However, it would seem that as far as BASUG is concerned, the business users are definitely appearing amongst our ranks but the hobbyists are still holding their own.

Lonely Apples

David Steward of

[illegible]

would like his Apple to meet others of a like mind with a view to forming a local group.

Membership

by Jim Panks

I would like to thank all those that sent renewals early and also completed the forms in full.
The major amount of work in compiling the new data base was done by my wife, who spent many hours bashing away at the keyboard and trying to decipher the shorthand/scrawl on a small percentage of the forms. Never mind, the majority of the work has been done and now I can slow down.
As many of you who have met me know, I hold very strong views on some aspects of B.A.S.U.G. I think that the amount of support given to some activities held for your benefit are so bad that the people involved lose faith overnight. I therefore ask all members with ideas to submit them. All the committee asks is that you tell us what you want and then support the ideas when we arrange them.
Well I hope to see some of you at the exhibitions that will be taking place in the near future.

(Apologies to Jim whose column was missed from the last issue)

MEMBERSHIP NUMBERS

Your long awaited membership cards will be sent to you as soon as BASUG Ltd is finalised. In the meantime you can find your membership number on the address label on your Hard Core envelope.

S.I.G.'s

Business users of the Apple will be pleased to learn that Jim Watson of Margate has volunteered to coordinate the setting up of a

BUSINESS SPECIAL INTEREST GROUP.

Jim will be looking for help, as like other business users, he has his own business to run. All offers of help or ideas for the group should be sent to Jim c/o the BASUG PO Box.

Important

DOS 3.3 UPGRADE KITS

Have you upgraded your DOS from 3.2 to 3.3 yet ? If not, you'd better hurry.
It seems that Apple in their wisdom have announced that they have discontinued manufacture of DOS 3.3 upgrade kits !!
BASUG is taking action on your behalf, of course, but first we need to know how many of you still have the old 3.2 and the intention of upgrading. Please let us know AS SOON AS POSSIBLE

The British Computer Society

The following message has been received from J. O'Sullivan, Vice-Chairman of the British Computer Society.

"I am writing to Micro User Groups to invite your members to attend meetings of the BCS London Branch.
You will see from the enclosed programme card that we normally meet on the third Thursday of each month, in the Charing Cross Hotel, London. The card also lists other BCS branches in the Greater London area. All meetings are published on the BCS page of "Computing" every week. There is no charge for admission.
... The BCS is the professional body for all those working in any field of computing with over 25000 members and also acts as a learned society. London Branch is the largest of the Society's branches. The Society has an Affiliate Class of membership for all those with a non-professional interest in computing, and for this no formal qualifications are required, and there is only a nominal subscription.
If you require any further information on the BCS, or any of our meetings, please contact me at BCS Headquarters direct.
I look forward to meeting you at one of our meetings.

J. O'Sullivan
British Computer Society
13 Mansfield St
London W1M OBP

STORYBOARD

A READING SKILLS PROGRAM

CLOZEMASTER

A READING SKILLS PROGRAM
BY CHRISTOPHER JONES

QUESTIONMASTER

BY TONY WILLIAMS GRAHAM DAVIES

Write your own CAL routines

Wida Software

Specialists in Educational Software for Schools

ENGLISH FRENCH AND GERMAN

Send 9 x 6 s.a.e. for catalogue
2 Nicholas Gardens · London W5 5HY

Tel. (01) 567 6941

We've opened up the whole country.

MASS MICROS

- *WE SUPPLY NATIONWIDE*
- *WE INSTALL NATIONWIDE*
- *MAINTAINED NATIONWIDE*
- *BEST PRICES NATIONWIDE*
- *LEASING SCHEMES*

Our unique service allows you to buy anywhere in the UK with the confidence and knowledge that the equipment we supply is at the right price and supported.
Our experience of putting together the correct system for you ensures full satisfaction.

- COMPLETE RANGE OF SOFTWARE, PRINTERS AND PERIPHERALS
- 12 MONTHS NEXT DAY SERVICE INCLUDING PARTS, LABOUR AND TRAVEL TO YOUR SITE

AUTHORISED DEALERS
commodore
ICARUS
SUPERBRAIN
apple
TABS

MASS MICROS

- **ASK FOR SERVICING QUOTES ON NEW AND EXISTING EQUIPMENT**

Wellson House, Brownfields,
Welwyn Garden City, Herts.
Tel: Welwyn Garden (07073) 31436/7
Telex 298641

Please send for FREE information pack on Apple ☐ CBM ☐ Superbrain ☐
Printers ☐ Maintenance ☐

Name .....

Company .....

Address .....

Ref [redacted] Tel No .....

Diversi-DOS (tm)

/Editor: This latest addition to the BASUG Software library marks a departure from previous practice, but one which fits in with our philosophy. As you will see from the instructions printed below, payment is to be made by the honour system which in view of the value of the program, I am sure you will abide by.
In case the title misleads you, this is not another Omni-Dos, but a Fast Dos and - hold it folks - 16k printer buffer combined. It is compatible with Apple Writer I but not Apple Writer II, as far as we have been able to ascertain/

Diversi-Dos

PLEASE COPY THIS DISK AND GIVE IT TO EVERYONE YOU KNOW ! ! ! ! ! ! ! ! !It is legal to copy and distribute Diversi-DOS, subject to certain restrictions (see "Licensing Information"). However, to legally run the Diversi-DOS program on this disk, please send $25 directly to:

DSR, Inc.
5848 Crampton Ct.
Rockford, IL 61111

You will receive a User Support Number and a Validation Sticker as proof of payment. Your honesty will allow us to distribute future programs this same, low-cost way.

Putting Diversi-DOS onto a disk
* *

The first 3 tracks of a DOS 3.3 disk contain the Disk Operating System (DOS). When the disk is booted, the DOS on these 3 tracks is loaded into the Apple memory. This DOS then controls all disk operations.
Diversi-DOS modifies the first 3 tracks on a disk. When the disk is booted, the Diversi-DOS operating system will be loaded. Diversi-DOS will then control all disk operations.
To add Diversi-DOS to a disk, enter "2" from the main menu. Then insert the DOS 3.3 disk and press <return>. Diversi-DOS will be written onto the first 3 tracks on this disk. From then on, whenever you boot the disk, it will load Diversi-DOS instead of the old DOS 3.3.
The other way to add Diversi-DOS to a disk is to initialize a blank disk (press "9" from the main menu and type: INIT HELLO). Diversi-DOS will be automatically placed on the disk.

Unprotected Commercial Disks:
In most cases, Diversi-DOS can be added to any unprotected disk. Commercial programs which uses DOS 3.3 in the standard way should work with Diversi-DOS. However, any program which references specific locations within DOS may crash with Diversi-DOS, since these locations may have been changed. To be safe, it is best to have a back-up copy of the commercial program before adding Diversi-DOS. Note: To install GPLE, "BRUN PLE.48".

Running Protected Programs
* *

Many protected disks will work with Diversi-DOS by booting Diversi-DOS and running the startup program on the disk (usually named "HELLO"). For your convenience, option "3" in the main menu will do this for you. You will be prompted to insert the protected disk and press <return>. The name of the start-up program will be displayed on the screen, and this program will be RUN (or BRUN if needed). This bypasses the loading of the old DOS 3.3 operating system off the protected disk, so that Diversi-DOS will stay in memory when the protected program runs.

After you have run a protected disk this way, you can make the process more convenient by making a custom Diversi-DOS boot disk for that protected program. Press "7" from the main menu to enter BASIC. Then enter the following program.

```
10 INPUT "PUT IN DISK AND HIT RETURN"; A$
20 PRINT CHR$(4);"RUN HELLO"
```

After you enter this program, put in a blank disk and type INIT HELLO. You can now use this disk to boot Diversi-DOS and start the protected program. First put in this "boot disk" and turn on the computer or enter "PR#6". Then put in the protected disk and press <return>.
Note: When you run a protected disk using option "3" on the Diversi-DOS master, it will tell you the name of the start-up program. If the name is not "HELLO", then use the correct name in line 20 of the above program. If it is a binary program, then use "BRUN" instead of "RUN" in line 20.

If It Doesn't work:

Many protected programs will not work when you try this procedure. This is because they use a non-standard DOS.

Making Back-Up Copies
* * * * * * * * * * * * * * * * * * * *

To make back-up copies of this Diversi-DOS master disk, choose "4" from the main menu. You will then be asked to remove the master disk and put a blank disk in the same drive. This disk will be initialized, and the Diversi-DOS master files will be written onto the disk.
When the copying process is finished, insert a new blank disk and press <space>, if you want to make another copy. Note that you never need to re-insert the master disk to make copies.

When you press <space>, a copy will be made on Drive #1. If you wish, you can press <ESC> to make a copy on Drive #2. This will allow you to make multiple copies more rapidly by alternating drives.

Keyboard/Print Buffer Utilities
* *

Menu option #6 will write the file BUFFER onto a disk. Then, when you BRUN BUFFER, the keyboard type-ahead buffer, and print buffer will be installed. This routine assumes that you have a parallel printer card in slot #1, and a 16K RAM card in slot #0. If you don't have this hardware, you will have to modify the BUFFER routine (see below).

Keyboard Buffer:

With the keyboard buffer installed, anything you type on the keyboard is stored in memory so it is not missed. If the computer is reading the disk, for example, characters which you type will be stored until the computer is ready. Thus, you can enter another command while the computer is processing a previous command.
Since this feature is done entirely in software, some programs will not be compatible. For example, if a BASIC program tries to read the keyboard with PEEK's instead of the usual GET or INPUT, then the PEEK will never find a character. That character will be waiting in the memory buffer. The result is a program which does not respond to the keyboard. If you find a program which does not work properly with the keyboard buffer, you can disable it.

Print Buffer:

Whenever the computer tries to send a character to a printer, it normally has to wait until the printer is ready. Since the printer is much slower than the computer, there is a large amount of time spent waiting. The print buffer utility solves this problem by saving the characters to be printed in memory until the printer is ready for them. Even though the characters have not actually been printed, the computer can go on to its next job, since the characters are safely stored and will be printed when the printer is ready.
The print buffer is installed when you BRUN BUFFER. You must have a 16K RAM card in slot #0, and a parallel printer card in slot #1. To turn on the printer, enter PR#1 as usual. If you are turning on the printer from a program, you must enter:

```
10 PRINT CHR$(4);"PR#1"
```

That is, PR#1 should be entered as a DOS command from within a program. Future PRINT statements will send output to the print buffer, and then to the printer, when ready. The output will be displayed on the screen as it goes into the print buffer.

Modifying the BUFFER file:

To modify the BUFFER file, press "5" from the main menu. You will be asked a series of questions about your configuration.

The old Apple "Serial Interface Card" cannot be supported due to its design limitations. In general, most parallel printer cards will work with the "PARALLEL" option, and most serial printer cards will work with the "COMMUNICATIONS" option (i.e. Versa-Card, CCS 7710, etc.).

All 16K RAM cards should be compatible. If you have a 32 to 128K card from Saturn Systems or Legend Industries, you can use it for a larger print buffer. If you have another brand of card, ask the

manufacturer if it uses the same bank switching scheme as Saturn or Legend. If it doesn't, you can still use it as a 16K card. After you are finished with the changes, you will be asked to insert a disk. The new BUFFER file will be written onto this disk. Then, if you want to put the new BUFFER file onto another disk, use option "6" from the main menu.
Notes:

The print buffer utility disables the software features on the printer card. Commands to the printer card will no longer have an effect (i.e. ctrl-I 80N with a parallel card). After you BRUN BUFFER, control of the printer must now be done with POKE's (see "Diversi-DOS POKE's"). For example, instead of ctrl-I 80N, enter POKE 49028,96. This turns off output to the screen and prints the full width of the printer. When you BRUN BUFFER, the INIT and CHAIN commands are disabled. If an INIT is issued, it will be ignored, but no error will occur. To restore the INIT command, re-boot a Diversi-DOS disk.
The BUFFER utility may not work with some programs which change locations $36 to $39. Also, the Applesoft CHRGET routine is modified. This might cause problems with certain programs which also patch here (i.e. compilers). The following program will install the BUFFER utility without patching CHRGET:

```
10 ?CHR$(4);"BRUN BUFFER"
20 POKE 46965,16
```

To use BUFFER with Applewriter I, use the printer address $AE8E and turn off the screen (POKE 44671,96).
Flushing the Buffers:

If you type some wrong characters, enter ctrl-X to cancel them. The keyboard buffer will be emptied.
Similarly, to flush the print buffer, enter ctrl-X twice in a row. All characters waiting to be printed will be lost. Pressing RESET does not flush the print buffer. To avoid any problems, it is best to set the printer "off-line" before pressing RESET.

Diversi-DOS Error Messages
* *

Diversi-DOS prints error message numbers, instead of the full message. These error message numbers are listed on page 200 of the DOS 3.3 manual. You may want to tape a print-out of these messages to the top of your computer:

```
 1......Language Not Available
 2,3....Range Error
 4......Write Protected
 5......End of Data
 6......File Not Found
 7......Volume Mismatch
 8......I/O Error
 9......Disk Full
10......File Locked
11......Syntax Error
12......No Buffers Available
13......File Type Mismatch
14......Program Too Large
15......Not Direct Command
```

Random Access Text Files
* *

Diversi-DOS contains an optional faster way to specify the Record and Byte parameters when reading or writing a random access text file from Applesoft BASIC. CALL 1005 sets these parameters to the values in the Applesoft variables named "RE" and "BY" respectively. The following example will illustrate how to use this feature.
Standard DOS way:

```
10 PRINT CHR$(4);"OPEN RANFILE,L500"
20 BY=0
30 FOR RE=0 TO 100
40 PRINT CHR$(4);
   "WRITE RANFILE,R";RE;",B";BY
50 PRINT "TEST DATA"
60 NEXT
70 PRINT CHR$(4);"CLOSE RANFILE"
```

Diversi-DOS way:

```
10 PRINT CHR$(4);"OPEN RANFILE,L500"
20 BY=0
30 PRINT CHR$(4);"WRITE RANFILE"
40 FOR RE=0 TO 100
50 CALL 1005
60 PRINT "TEST DATA"
70 NEXT
80 PRINT CHR$(4);"CLOSE RANFILE"
```

The CALL 1005 is over 60 msec faster than the standard DOS PRINT.

Advanced Programming Information
* *

The following information is for advanced programmers only.

To write the instructions to a text file for editing:

```
LOAD HELLO
1120 ?D$;"OPEN INSTR"
1121 ?D$;"WRITE INSTR"
1165 ?D$;"CLOSE"
RUN
```

Then choose the option to send instructions to a printer.

Printer Cards:

The file "ASMDIV" contains the printer card data. To add a new card type, BLOAD ASMDIV and enter the card data as described below. Then BSAVE ASMDIV,A$2000,L$70FA. Then reboot the Diversi-DOS disk and select option "5". Your new card type will be listed as an option.

To add a new card:

Location $7800 contains the number of cards (currently 9). Change it to $A to add a new card.

The first card description begins at $7840. The next begins at $7880. Each description is $40 bytes long. The first open space is at $7A80 which is where your new card data will begin.

Data is output to the printer using the following subroutine (PSLOT is the printer card slot #):

```
POUT    PHA ;PRINTER OUTPUT ROUTINE
SCHECK  LDA ASTATUS+(PSLOT*ASADD)
        AND #MSTATUS
        EOR #ESTATUS
        BNE SCHECK
        PLA
        AND #MDATA
        STA ADATA+(PSLOT*ADADD)
        RTS
```

The printer card data format is as follows:

Byte/Size	Name	Description
$00/$20	PINIT	Init routine
$20/$10	PNAME	Card Name
$30/ $1	----	Must be 0
$31/ $2	ASTATUS	Status address
$33/ $2	ASADD	Slot adder
$35/ $1	MSTATUS	Status mask
$36/ $1	----	Unused
$37/ $1	ESTATUS	EOR byte
$38/ $1	MDATA	Data mask
$39/ $2	ADATA	Data address
$3B/ $2	ADADD	Slot adder
$3D/ $3	----	Unused

The subroutine PINIT is called to initialize the printer card when BUFFER is BRUN. At the time of the call, the X-reg contains the value "N0" where N is the slot #. $36 and $37 contain $CN00. A-reg is $8D.

Polling the Keyboard:

After you BRUN BUFFER, location $BF87 contains the routine which checks the keyboard and printer. If you write an assembly language routine, you should JSR $BF87 every 10 msec or less. A,P are destroyed / X,Y are preserved. Note that before BRUN BUFFER, $BF78 contains an RTS.

Interrupts:

Diversi-DOS does not disturb $45, so interrupts will not crash Diversi-DOS. Even so, DOS disables interrupts for up to 3 seconds during an RWTS call. After you BRUN BUFFER, you can JSR $AEF0. After this, Diversi-DOS will do a CLI in RWTS every 11 msec or less.

Reset Hooking:

Reset should be vectored by changing $9D5E and $9D5F (not $3F0,$3F1). This insures that DOS is reset also.

Text Files:

The reading of text files is only speeded up if input is taken from the monitor GETLN routine (JSR $FD6A or $FD6F).

Licensing Information

* * * * * * * * * * * * * * * * * * *

Although Diversi-DOS is copyrighted, DSR allows copying and distribution of the program subject to the following restrictions:

1. Anyone receiving a copy of Diversi-DOS must send a $25 license fee directly to DSR, Inc. within 2 weeks.
2. Each computer running Diversi-DOS must have a Diversi-DOS validation sticker attached.
3. Any company, club, or individual may charge up to $5.00 for distributing copies of Diversi-DOS, provided it is clear that an additional $25 fee must be sentdirectly to DSR. A catalog listing should read: "Diversi-DOS (distribution fee only) $5.00".

This method of distribution offers tremendous advantages. The cost of Diversi-DOS is less than 1/2 of what it would be if distributed conventionally.

Also, you can test the program before you pay.
To obtain a Diversi-DOS license and validation sticker, please send $25 for each computer to:

DSR, Inc.
5848 Crampton Ct.
Rockford, IL 61111
815 877-1343 (Visa/MC)

Note: The $25 fee covers the cost of the validation sticker only. If you would also like a new disk with the latest Diversi-DOS version, send $30.
Publishers:

There is no royalty fee for publishers who sell their programs with Diversi-DOS included. The only restriction is that each disk must display the Diversi-DOS "Request for Payment" message during the boot. To add this message to a disk, boot the master disk and select option #2 (add Diversi-DOS to unprotected disk). Put in your disk, and press <space> instead of <return> when you see the prompt "Press return for normal start program".

Publishers who do not wish this message to appear on the screen can license Diversi-DOS for a $200, one-time fee for all their programs. To do this, the following message must appear in the written documentation:
This disk contains a high-speed operating system called Diversi-DOS(tm), which is licensed for use with this program only. To legally use Diversi-DOS with other programs, you may send $30 directly to:
DSR, Inc., 5848 Crampton Ct., Rockford, IL 61111. You will receive a Diversi-DOS utility disk with documentation.

Alternatively, publishers may include the 2 Diversi-DOS files, HELLO and ASMDIV, on their disk at no charge. The disk must be initialized with standard DOS 3.3. The user then has the option of adding Diversi-DOS to the disk, and sending the $25 to DSR, or using standard DOS without paying an additional fee.
Diversi-DOS is a highly modified version of DOS 3.3. DSR is licensed by Apple to distribute the DOS 3.3 code present on the Diversi-DOS master disk. However, to add Diversi-DOS to a disk, publishers must still obtain a license from Apple for DOS 3.3. For more information, call Terri Hasbrouck (408 996-1010).

Diversi-DOS POKE's
* * * * * * * * * * * * * * * * * * * *

Initializing Data Disks:

The following program modifies the Diversi-DOS INIT function. After running this program, INIT will initialize non-bootable "Data" disks with 32 more free sectors. If you boot a data disk by mistake, you will have to press <reset>.

```
10 POKE 42344,76
20 POKE 44723,4
30 POKE 46922,96
```

Reset Hook:

The following POKE's will re-RUN a BASIC program when <RESET> is pressed:

```
POKE 40286,252
POKE 40287,164
```

Language Card Re-load:
To avoid reloading a RAM card with BASIC on every boot:

```
POKE 49107,173
```

Motor Start Time:

To speed up DOS even more, you can change the motor start time to 1/2 sec. To do this, change the HELLO program as follows:

```
LOAD HELLO
85 POKE 47102,236
SAVE HELLO
```

Over 99% of disk drives will work correctly this way. However, an occasional drive will destroy disks.
To restore the 1 second delay:

```
85 POKE 47102,216
```

Print Buffer Utility:

After you "BRUN BUFFER", the following POKE's will modify the printing:

POKE 49028,96 - Turns off output to the screen when printing
POKE 49028,76 - Turns it back on

POKE 44762,96 - Turns off line feed after carriage return

If you have an old Integer BASIC machine with an APPLESOFT ROM card:

```
BLOAD BUFFER
POKE 32915,N (N is ROM card slot #)
BSAVE BUFFER,A$8000,L$6FA
```

If the ROM card is not in slot 0, change DOS as follows:
POKE 42424,N*16+128
POKE 42432,N*16+129

If you have an Integer machine and use a RAM card in slot #0 to load Applesoft, you can still get a 4K print buffer as follows:

(After BRUN BUFFER)
POKE 48885,1
POKE 48975,128
POKE 48990,139

To install the buffers from withing a program without printing on the screen:

```
10 PRINT CHR$(4);"BLOAD BUFFER"
20 POKE 33152,96:CALL32768
```

Updates
* * * * * * * * * * * * * * * * * * * *

This is Diversi-DOS version 2-C. "2" refers to the Master Disk version. "C" refers to the DOS version (displayed during a CATALOG). If the program is updated, all licensed users will be notified by mail. The updated disk may then be purchased for $5.00 to cover the cost of materials.

Licensed users will also be notified of any future programs released by DSR.

Special note:

Anyone interested in computing for the handicapped, please write to:

Diversi-DOS Mover routine
* * * * * * * * * * * * * * * * * * * *

Menu option 7 will save the file "DDMOVER" onto a disk. When you BRUN DDMOVER, Diversi-DOS is moved onto a RAM card, giving more room for BASIC programs (you cannot BRUN DDMOVER after BRUN BUFFER).

DDMOVER requires Diversi-DOS version C, a 16K or greater RAM card in slot #0, and Applesoft ROM on the motherboard. If these are not all present, DDMOVER has no effect.
After DDMOVER, you will notice the following changes:
1. CATALOG shows "Free Sectors"
2. Integer BASIC is unavailable
3. You cannot BRUN BUFFER
4. INIT formats non-bootable disks
5. HIMEM=48896 (don't set any higher)
6. An additional DOS command "PAD"

will print the address and length of the last BLOADed file.
If you BRUN DDMOVER from within a program, the program is lost. The following statement will cause DDMOVER to RUN the program "HELLO" after moving DOS:
0 POKE 40206,0:?CHR$(4);"BRUN DDMOVER"

If this is the first statement of a program named "HELLO", the effect is to move DOS to a RAM card, if present. If a RAM card is not present, DDMOVER returns, and the HELLO program continues. If the card is present, DOS is moved and the HELLO program is RUN again. When DDMOVER is called again, it will return, since DOS has already been moved. After this statement, IF PEEK(55) > 190, then DOS was moved.
Note: Some existing programs may not work with a moved DOS (including FID, and RENUMBER). The following will fix the FID program:
BLOAD FID
POKE 2132,169:POKE2133,120
BSAVE DDFID,A$803,L$124E

PATCH file
* * * * * * * * * * * * * * * * * * * *
Menu option 8 will save the file "PATCH" onto a disk. BRUN PATCH converts standard DOS 3.3 into DIVERSI-DOS. This can be used to convert a running DOS to Diversi-DOS, i.e. with hard disks. If it doesn't work, try this:
BLOAD PATCH
POKE 36816,0
BSAVE PATCH,A$8C00,L$470

If it still doesn't work:
POKE 36780,0
This second change leaves the patch area $BCDF to $BCFF open, but disables the CALL 1005 feature.

Checksum Feature
* * * * * * * * * * * * * * * * * * * *
When this disk boots, it now performs a checksum of itself. If any bytes of the program are changed, you will hear 2 beeps. This indicates a bad copy of Diversi-DOS.
Note: If you modify Diversi-DOS, the menu will show version number 2-C*. The "*" shows that changes have been made. To eliminate the beeps, reboot and make a copy (option 4).

Book Reviews

A Review of two books on BASIC

by Jim Panks

Many of our members are new to computers, and the problem of learning BASIC can be difficult. There are many books around on the BASIC language and many claim to teach it in a few weeks.
My personal experience is that learning from a book is not the easiest way and that authors try to pack to much into too few pages. I believe that if you are to learn from a book it must take you slowly through from the very easy and basic steps to the harder and more complex ones. Many books, I am afraid, do not use this method.

The two books in question are both from the publishing house Prentice-Hall. In the past few months they have published many books specific to the Apple. The first is an American book and the second is by two Irish authors.

I have spent some time reading the books, but because of the time available I have not tried all of the exercises in them. Both books claim to teach how to operate the Apple in BASIC.

APPLE BASIC.

The first book is called APPLE BASIC and is written by an American Richard Haskell. It is A4 size, contains 182 pages and first I was impressed by the number of pictures and illustrations. Almost every page has a large picture or illustration and many have two.

The author has written two other books on BASIC and they are based around the Pet and TRS 80 Color Computer which would account for his style. In the preface the author suggests the book be used as an introduction to BASIC in schools and colleges as well as for home study. I think that it would be more suited in a school than for home use.

This book starts with the very basic operations required to plug the machine in. It then explains how to boot a disk and how to find out which language is present. By page four we are into strings and print statements. I must admit that this book is ideal for younger members of the family, and the way in which it is written appears to be aimed that way.

At the end of each chapter an exercise is set to test the reader, although no answers are given. The idea is good, but if no teacher is at hand the poor reader is stuck.

This book is very American and I think that the older reader would find it slightly boring. The major part of the book is placed around a program of Hangman; the idea is to use as many aspects of BASIC as possible and use them in the program. The book tends to use graphics a lot and the more serious side of programming is not at all well represented.

I did like the explanation of loops and nested loops. The author uses a railway line and stations to explain the way in which loops work; a very neat idea.

The appendices and in particular the explanation of the hexadecimal system are very good. The index is comprehensive, unlike many books today.

My conclusion is that this book is a good buy for the children and well worth the £9.70 (Paperback) or £14.95 (Hardback). It contains some useful information in the appendices and, by and large, will give a good grounding in the use of BASIC.

THE APPLE PERSONAL COMPUTER FOR BEGINNERS

The second book is THE APPLE PERSONAL COMPUTER FOR BEGINNERS and is written by Seamus Dunn and Valerie Morgan. It is slightly larger than A5 and contains 300 pages. I found that the size was about right, not taking up too much needed space on the bookshelf. The format of the book is very different from the first book, although the aims are similar. There are no pictures but plenty of examples of what the screen should look like. I find that this approach is good and means that there is more room for text.

The book starts with very basic items and by the second chapter is getting down to the use of BASIC, I found the layout very workmanlike and it appeared to be aimed at the person who wants to learn about BASIC at home. At the end of each chapter a set of exercises are given and unlike the first book a solution is given in the appendix. This is very useful and will help those without outside help.

The solutions are neat and it is obvious that the style may be different to other programmers. The object is to make the reader understand the chapter.

One bad point was the lack of information on the use of tape with the Apple. I know that most users have access to disk, but many start with tape, and this book is aimed at the beginner; one comment that may be useful for later editions.

As it is the book deals with each subject in a clear and thorough way and great use is made of actual routines. Chapters are grouped so that the subjects are used to make a complete program at the end.

A chapter is devoted to machine code. It does not go very deep, but the idea is to give a small insight into the use of machine code in BASIC programs.

I like the chapter on structured programming and found that it gave a real understanding of the subject. This book is really designed for the reader sitting at a machine, but I found that I was able to enjoy reading it away from the computer, and still understand it.

The appendix includes a section on the different languages available on the Apple other than BASIC. It looks into the difference between Applesoft and Integer, although all the program material in the book is in Applesoft. There is a section on PASCAL and its uses. I also liked the many tables included.

The authors have had some previous experience in writing computer books and this shows in the way that the book is written. Unlike the first book, which I concluded would be better for children at school than for adults, this book is ideal for those wishing to learn at home. It is not as attractive to children because of the lack of pictures, but I am sure that they would learn more from this book.

My conclusion on this book is that if you do not need pictures to help you learn then it is excellent value at £9.95 (Hardback) or £6.95 (Paperback) and it contains most of the information needed to learn to program in BASIC.

My final comment is that, although aimed at the beginners end of the market, the two books reviewed are very different in the way that they approach the problem. I find that the style is completely different, with the first book relying heavily on pictures and graphic type programmes to get the message across, and the second on very well structured chapters with masses of information and more useful programs.

I am encouraged by the standard of the books now coming from Prentice-Hall. It was not that long ago that computer books were scarce, expensive and very specialised. More books of this standard will help to spread the word of computers.

Software Library

Changes in software library in store

Besides the public domain/contributed software we have had on offer since the club began, we have decided to offer software written by members at a modest price and BASUG will pay a royalty to the authors. Some of this will be announced in the next HARDCORE but meanwhile, we have received a fast DOS from the STATES. We can distribute this on the understanding that you, the purchaser agrees to send the balance to the authors. The package is good, as the following printout from the disk shows. All documentation is on the disk, but is also printed here for your reference.

If you have any software which you think ought to be marketed through BASUG, please contact us.

So You Want To Write A Book ?

by Tony Williams

"You only have to press a button and it all comes out!"
I am now emerging from a two year word processing odyssey, the end of which will be a 350-page book on cybernetics published by a well-known American scientific publishing house. Being an Apple user and word processing devotee when starting the project it was predestined that I would be writing it on disk. It was also written in the stars that when the publishing house offer ed me more money to present the project in "camera clean copy" I would accept. This choice is likely to face many of our talented writing members and they may be interested to hear just what went wrong in my case and how it came to take so long. This article is only marginally technical in content as I am more concerned to pass on to you my experience in dealing with technical editors and publishers - and their accountants. It isn't any of your concern to know how much I can make from work of this kind, but since this is a major factor in the process, I shall have to tell all.

First let me say a word or two about the project itself. My part was to translate from Russian a work on cybernetics by a rather famous Moscow scientific populariser - a wide-ranging eclectic piece on the problems of systematising the "inventive process." I shall not bore you with the details of translation. I was offered a total of nine dollars a page for presenting the translation in "camera ready form" - and although even two years ago this was a woefully poor rate I accepted, partly because the book was long enough to warrant it, some 150 pages in the original and not too difficult. I failed to elucidate one vital statistic, namely what they would have offered for a simple typewritten draft - this becomes more important as the story progresses.
I completed the actual translation in several weeks, filling five disks of Apple Writer I files. The publishers had sent me precise instructions on layout, paragraph indentation, etc, and mailed me a parcel of special single sheet paper on which the final was to be printed. The book contained a mixture of subscripts, superscripts, chemical formulae, some technical drawings which I could reproduce by hand and others which could only be tackled by the publishers' art department. At that time the only way of obtaining subscripts and mathematical formulae available to me was Ian Trackman's Go-Between for the Centronics 737 - a package with which I was very familiar. I had taken the precaution of sending off a sample printout to the series editor, a professor at a British university. He immediately saw the advantages of the presentation and approved it. So I went ahead with work on the translation. The first fifty pages were sent in, checked by the editor, who sent them on to the publishers. They in turn sent them on to the author in Moscow, who, after a delay of only two months expressed his approval. By which time the translation was complete and I was ready to do the print run. That was when I first heard the ominous phrase, "Now you can press the button and print it out."
This is the notion which seems to delude publishers into a false sense of ease - by cutting out the dreaded typesetting stage, the making up of expensive plates, the time-consuming mailing of galley proofs back and forth. The contract now contains a wonderful caveat - "Since no proofs will be provided, check your paper very carefully before sending it to the editor. Any undiscovered errors in your camera-copy manuscript will appear in print." This is very flattering for the author who relishes the chance to be in complete control of his work - but agreeing can add up to a costly misjudgement.
So I "pressed the button" and printed it out - but, since special single sheets had to be used this meant standing by the machine to print out what had now swelled to 230 pages (Russian expands by about one third upon translation). Tractor feed

being ruled out, and the 737's paper friction feeding characteristics being none too good, this meant a certain wastage rate. It was also difficult to link Apple Writer I files automatically and, what was worse, I could not print to the screen in order to check that my various screen print commands, page breaks etc, were correct. In practice this meant discovering errors only upon printing and this led to a further wastage - as much as one page in ten had to be redone for one reason or another.

The publishers specified that the running title on even pages should bear the title of the book, and on odd pages the name of the author. This is normal book publishing practice but neither Apple Writer I nor II can handle it. But, not to worry, I could print it all out at the end and 'strip the titles in'. Page numeration presents a much greater problem than normally assumed in word processing manuals. The main difficulty is deciding where page one is. This can only be solved by communications - by contacting the publishers' production staff, getting a decision and sticking to it. If something changes, however, as in this case shifting the table of contents from the end to the beginning, the only way to correct it is to print it all out again (no thank you), or 'strip the numbers in' once more.

Apple Writer I makes no provision for index paging - for ensuring that page references are updated for the translation. You could do it, of course by loading up each file, searching for the word, checking which 'page' and cross checking against the real pages with their stripped in words - but it isn't worth it. Easier by far to do an old fashioned paper search.

So the proud moment comes when the document, by now a hefty packet is tied up in pink ribbon, photocopied for security and mailed off to the series editor. My disks are carefully stored away, the five backup disks (which lag only slightly behind the final) are stored in a separate place and the five backups of the backups are deposited with a neighbour. My disks and I wait.

After a few weeks the series editor expresses his satisfaction again, and off the package goes to the publisher.

Samples are sent to New York. And rejected. Under no circumstances will they accept the dot matrix printout. It has to be done again on my daisywheel.

But it doesn't really matter - I "only have to press a button and change it." By now I am hooked - they haven't paid me much of an advance and if I am to protect my investment in time I have to stay with it.

Trouble is, "pressing a button" won't quite do it. I have to dust off the disks, remember which files are the best generation, load up each file in turn and remove my precious "Go-Between" printer commands. Gone are superscripts, subscripts, mathematical symbols, gone is emboldening, away with underlining. I set a full week aside now, in sacroiliac distending Olympia supervision, feeding sheets in, shaking in time with its thunderous its 14 cps. Again I cannot print to screen, so the inevitable command mistakes and other typos which creep in are only revealed too late. Pagination again. New index paging again. Discover one or two corrupt files - corrupt in all generations, of course. Have to type one or two files over again. Spot mistranslations and other mistakes and take the opportunity to fix them. After a week the pages are printed out, the new page numbers and running alternate titles stripped in, photocopies taken, manuscript delivered by hand direct to the publisher.

The initial response is fine: all perfect - something seems to be wrong with the pagination and one or two pages didn't seem to have gone in straight but still, no problem. Now the whole thing has to go to Moscow for the final approval.

Four months of nothing. Even though the contract says full payment is to be made within thirty days of delivery of manuscript no payment ensues. What is wrong, I telephone. Ah well, it's still in Moscow: payment will only be made after it is back from there. Thoughts of litigation loom large: that was not in the contract.

Finally the word comes back from the USSR, two commas have to be changed. So what about the money then.

Well, we've had a publishing conference and the manuscript is fine, we would like only one small amendment - one and a half instead of single spacing. "But you only have to press a button. Oh and while you are at it, we would like paragraphs indented five spaces..." Calamity, known only to myself. I now have six sets of five disks, and a generation gap of enormous proportions, not helped by the fact that I was not entirely meticulous

in saving every latest improved un-Gone Betweened file. The only dignified response to this is one of total up yours non-cooperation, it was their fault, after all. The series editor backed me up.

Yet beginning to think it over... the payment was in dollars, I had not yet been paid, and in the meantime the dollar had strengthened by some thirteen percent. I am also paid by the page and the number of pages would now increase by fifty percent. So I capitulated again, money being the root of all my evil.

By now I have moved to 80-column Apple Writer II and want to take advantage of the WPL to link files - and I can also print to the screen which saves much time.

I discover a curiosity. Although the old Apple Writer I binary files could be over 130 sectors in length (just!), on conversion they will not fit into Apple Writer II memory, even with a 16k Ramcard. When I tried to do this the system just collapsed. And of course, being a binary file, you cannot load up a specified segment of a file for conversion to Apple Writer II. How odd, I thought and had to fall back on excavating my Apple Writer I disk and breaking the files down into manageable units.

If only I didn't have to print single sheets! Of course I have to work through each file changing the print commands ! to a full stop, but still everything goes much faster. I nix the single sheet idea, refusing to have my brain turned to black treacle standing sentinel by the Olympia. Heavy duty fanfold will do if I get the perforations right, then with my trusty scalpel and pritstick I can strip each sheet onto the publishers' paper. Silly really, but preferable. Moreover, even my jaundiced eye runs truer than the paper in the Olympia. The numbers will come in automatically - I think fondly. Except that at page 120 I return and discover a bad page break at page 46. By error I have allowed a single word to occupy a page (before a new chapter). Fortunately I can rescue the situation because there is enough room on the previous page to strip it in. But the page numbers are now thrown out and have to be typed in by hand. I also find I have misestimated the space needed for pasting in one of the many illustrations. The solution is not to reprint, but to repaste. New unexpected problems present themselves: you can link fifty files with the WPL which works marvellously, but what about the printer carbon ribbon cartridge. Its life is finite, but how finite? Are all carbon cartridges the same length? I think not.

By this point I think I have conveyed the idea to you. It is not yet the end of the saga, but the manuscript is finished and delivered for the third time, but the lessons are learned.

I could never go back to the old typewriter. The micro gives us immense power, and enables us to easily process vast amounts of materials at great speed. But it does not turn us into Leonardo men of virtu, into Renaissance dilettantes capable of doing everything. Publishers may want us to be writer, typist, proofreader, editor and typesetter all rolled into one - because it suits their convenience and their pockets. But we should resist these pressures. The end product, if it is to be professional should be done by professionals (even if only because a typesetting machine will always outshine even the most expensive home printer). I am the greatest sinner in this regard. My recommendation: be brutal. Suppress your instincts and give the publishers a draft typescript and and tell them to get on with it. However, by far the best solution is to cut the profit margin, call up Wordsmiths (or a similar typesetting agency) and send them your Apple disks. But because I stubbornly insist on exercising absolute control right down to the last minute I must have everything done on the premises and must suffer the consequences.

In case readers believe that mine is an isolated untypical instance, let me also mention one other 80-page booklet I have had published in the meantime. This was submitted in Apple Writer II daisywheel printout and accepted by the publisher in its entirety. It was printed without even a comma being changed. However, in the meantime it had spent six months in the publishers' design department being not designed. This is about the same time needed for typesetting. So the message has to be, if speed is the essence, don't submit camera clean copy.

A Relocating Linker

by Richard Teed

This article is intended as an enticement and at the same time as instructions for a relocating linker that I have put in the software library.

WHAT IS A RELOCATING LINKER?

A relocating linker is for assembly language programs only and can cut down development times considerably. This one will work only with the DOS Tool Kit assembler and makes it a much more useful program.
If you write a large assembler program using the DOS Tool Kit assembler then your program will consist of a number of files CHAINed together and will probably take twenty minutes plus to assemble - a long time: not only does it take 20 minutes to assemble but it also takes 20 minutes to tell you that the latest piece of code you added has an undefined symbol in it. It is therefore not surprising that many assembly language programmers have been turning to faster assemblers such as Big Mac and Lisa.
With a relocating linker you write or rather structure your programs differently to how you normally would using the Tool Kit assembler. A program is split up into modules of about a hundred to a hundred and fifty lines and you tend to link the modules to each other by subroutine calls. To make a change to your program you modify the appropriate file and reassemble it leaving the others as they were. This does not take very long because the file should be very short. Next you link all the files together using the relocating linker (which I have made very fast) and in next to no time you have implemented a change.
To give you an indication of how fast assembly can be, whereas it takes eighteen minutes for the relocating linker to be assembled in the normal way using the Tool Kit assembler producing about 12k of object code from about a dozen files. Using the relocating linker (on itself) it takes about a minute at most to assemble any one of the 32 source files and about one and three quarter minutes to relocate, link and output the object code. ASSEMBLY TIME IS THEREFORE REDUCED BY ABOUT A FACTOR OF 10!
Not only does the increase in speed make the DOS Tool Kit assembler a lot faster than assemblers like BIG MAC but with the exception of the macro facility it gives considerably greater power than BIG MAC.

WHAT YOU NEED TO RUN THE RELOCATING LINKER

Apart from a disk you will need a language card for the relocating linker to sit on and the DOS Tool Kit assembler.

WRITING FILES FOR THE RELOCATING LINKER

Files for the relocating linker are different to those you have been writing up until now. First, all files must be assembled at location $0100 for if not they will be relocated incorrectly. Secondly if you assemble a file then all external labels will give an UNDEFINED SYMBOL ERROR: to get round this you must put a REL at the start of your file and use a new instruction for external symbols, which is EXTRN. Any symbols that are accessed by other files must be declared with an ENTRY. A simple program would therefore look like this:-

```
        REL
        ORG     $0100
        MSB     ON
;
        ENTRY   PRINT
;
        EXTRN   POINTER
        EXTRN   COUT
;
PRINT   STA     >POINTER
        STY     >POINTER+$01
        LDY     #$00
LOOP    LDA     (>POINTER),Y
        BEQ     DONE
        JSR     COUT
        INY
        BNE     LOOP
DONE    RTS
```

Another module to complete the program might look like this:-

```
        REL
        ORG     $0100
        MSB     ON
;
        ENTRY   BEGIN
        ENTRY   COUT
        ENTRY   POINTER
;
```

```
        EXTRN   PRINT
;
COUT    EQU     $FDED
POINTER EQU     $60
;
BEGIN   LDY     #<TEXT
        LDA     #>TEXT
        JSR     PRINT
        RTS
;
TEXT    ASC     "HERE IS SOME TEXT"
        DFB     $8D
        DFB     $00
```

There is a problem that all global symbols (those declared in EXTRN or ENTRY statements) are sixteen bit so all page zero addresses must have a ">" in front of them as in the example above to avoid assembly errors. This also rules out relative jumps between files - which is probably a good idea.

LINK INSTRUCTIONS

Link instructions can either be typed from the keyboard or placed in a text file. Text files of instructions may call other text files of instructions to as many levels as desired. There are also good facilities for controlling printers during linking.

LIBRARIES

You will find that because of the general nature of the code you write certain routines are used by many programs. It might, however, be inconvenient to work out just how many of these routines you have used. To get round this put the names of all the files you want in your library into a text file. Place the name of the text file in the link text file but instead of putting the usual @ signs round it (to show indirection rather than that it is the name of a file to be linked) put it in brackets at the end of the file. As each file in the library is loaded, a check is made to see if it can resolve any outstanding external symbols and if not the file is discarded and the next is linked.
Putting a library file at the start of a link file will, of course, result in none of the files being loaded because there will be no EXTRNs to be resolved.
A typical example of a library is the implementation af all DOS commands, one per module, but not all programs where the library is included will use all the DOS commands. Thus only the relevant ones are linked into each program.

LINKER OUTPUT

The file name of each file that is being linked is displayed together with any global entry points. At the end of linking a table may be output showing undefined symbols and finally an alphabetic global symbol table is output.

ADVANTAGES AND DISADVANTAGES OF A RELOCATING LINKER

First of all here are the advantages:-
1) Quick loading and saving of files in the editor because they tend to be short.
2) Fast assembly of programs due to size and quick awareness of assembly errors because only a small section is assembled at any one time.
3) Forces the writing of modular code which tends to be more generalised than straight assembly language programs which will probably mean that modules can be used by a variety of programs.
4) No need to alter a CHN statement in files that are used in more than one program.
5) Labels that are not declared as ENTRIEs are local to a particular file and can therefore be reused in any other file; this solves one of the big headaches of assembly language programs, that of coming up with discrete labels for what may amount to a lot of code.
6) Due to high speed assembly it is more practicable to get hard copy listings at the assembly stage than usual.
7) Very easy to locate code at any desired address, simply by changing a line in the link text file or typing it a link time.
8) The ability to have libraries of programs to ease the construction of the link text file.
9) Due to the compact nature of relocatable binary files it is possible to have a great deal of code on line for linking.

Now here are the disadvantages:-

1) Difficult to convert a normal assembly language program to link format without prior planning.
2) Very tedious converting labels to page zero by placing a > in front of them.

3) Probably the biggest problem is that it is impossible to get a fully assembled listing. However, using the load map, and disassembler I had no trouble in producing a linked version of the linker.

AN OVERVIEW OF THE STRUCTURE OF THE LINKER

The linker is a one pass program, which may sound like an unwise system to have; however, there are generally far fewer global symbols than locals, so the symbol table is small and the source code (in this case the relocatable binaries) is compact and, as you will see, the memory intensive external symbol directory section of each file is reclaimed before the next file is loaded. The big advantage of keeping the linker one pass and its code core resident is that it works at a very high speed.
The symbol table is maintained in alphabetical order and builds from the top of memory down. Object code builds from the bottom of memory up: each relocatable binary file has a list of addresses to be relocated and a list of global symbols on the end of it. Once the file has been loaded, the relocatable addresses updated and the global symbols placed in the symbol table, the next file is loaded over the additional information of the last file, which keeps things nice and compact.
Each forward reference to a symbol uses six bytes which are put on a linked chain of forward references for the particular symbol: when the symbol is resolved the chain of forward references are also resolved and then placed on a free chain so that subsequent forward references can re-use the memory.

EFFICIENT LINKING

The advice here is similar to Applesoft:-

1) Symbols are in alphabetical order, so frequently used symbols should start with the letter A or a letter low in the alphabet. (Linking is very fast and it is doubtful that this will be significant in link time but you may find it useful).
2) Forward references require more memory and take longer to resolve than backward ones (again time will be little affected but if you have a lot of forward references in a large program you may find that memory becomes tight, but the program would have to be very big for this).
3) Always keep your files small. If you must have a big file then put it at the beginning, because at the end its external symbol directory memory will not be available for subsequent re-use.

If you order the linker from the software library then you will also get the source code, relocatable binaries and an editor with it, plus some better instructions on how to use the linker.

LOCAL GROUPS

Hants & Berks Local Group
by Fran Teo

We held our first meeting on Monday 14th March which was well attended by 19 Basug members.
Our sincerest thanks go to members John Jewell and Brian Atkin for providing a room for our use at Reading University. Not only was the coffee on tap but also a room full of Apples (what a sight!) which saved us all trudging along with our own machines to a draughty hall somewhere.
Brian Atkin first demonstrated the U-Net Local Area Network which links the Apples together to facilitate teaching. As we saw, this system has not yet got all the bugs ironed out of it but when it works is very impressive. Next Brian gave us a demonstration of SCRIBE which is a spatial modeller package designed for Architects and suchlike, then we were let loose on the machines to test our own creativity in design. Glancing around the screens at this stage I saw some very interesting ideas including a house with a bay window, drawn to my request although I was not the artist! All in all this was a very good start up meeting and we hope to go from strength to strength in the future.

Meetings are held on the 2nd Monday of the month, for further details please contact me on [illegible].
Our next meeting will be held on Monday 11th April at the same venue, Reading University.

The Linguist

A review by Tony Williams

To judge from its specifications The Linguist by Synergistic Software is an attractive package offering great potential to teachers and others wishing to write BASIC programs using foreign language character sets. Unfortunately the package is so deeply flawed that it is difficult to imagine it being used in any serious way by language teachers.

The package is essentially an extension of Synergistic Software's own Higher Text font-writing package - being an assemblage of weird and wonderful fonts, combined with software on disk enabling one to put together lists of words in one language (and font) and match them for testing purposes with words in any one of a dozen languages, including Hebrew and Russian.

The conceptual origins of the package are easy to reconstruct: some Apple freaks got hold of Higher Text, said "All right, terrific, now what can we do with it? Let's get a book of crazy alphabets out of the library, do the lot and sell it as a package." Unfazed by the fact of knowing nothing about foreign languages they went ahead and did it. This must account for the schoolboy howlers on page one in Russian, where they might be forgiven, and in German ("Deutch") where they might not.

This lack of care spills over into the programming and what emerges is a strange mix of brilliance and sloppiness. For instance, if the student inputting answers in the drill mode corrects his word by using the back arrow, and why not, his answer is corrected on screen but not in memory. When he presses return he is marked wrong, and he sees that whereas he thought he had produced Madel he finds his response is registered as Mha"a"bdekl. This is a rock bottom programming lapse which makes the drill program useless. Nowhere does it say that the back arrow is a no-no.

Other glitches. When trying to match, say, Hebrew with Russian the matched pairs are supposed to appear in two columns, but don't - they march up the screen in a single column. Hebrew, of course, does not have lower case letters, so when you press Control S (the Higher Text convention for accessing lower case) hi-res garbage patterns are produced on screen. The silly computer accepts this as Hebrew lower case and quite happily treats it as a word. Why this was not trapped out is anybody's guess.

In other respects the program is quite well thought out: it has Translator, Definer and Phrasebook modules and after an hour or two's familiarization it is possible to build up matching lists, using a simple menuing system. The European character set has a quite acceptable method of putting accents over letters: various symbols are assigned to £,$,&,(,),*,> and <. When these are pressed the character will back up to the last character typed and superimpose itself. This is not messy at all and is extremely flexible. Not unreasonably the authors Robert C Clardy and Charles J Fleishmann have not included a facility for generating your own fonts (in Georgian and Armenian?) but refer you instead to the Higher Text package. Which, I'm afraid, rather leads one to the conclusion that it would be more time-effective to go to Higher Text in the first place and write your own bugless programs using the fonts.

Small Ads

Original Keyboard & display enhancer	£40
Apple 16k card	£25 ono

S. Denney [illegible]

Visicalc Corner

by Frances Teo

CREATING A TEMPLATE

There are a series of guidelines to follow when creating a template - mainly a mixture of common sense and an understanding of the limitations (and capabilities) of Visicalc. First of all, what is a template? It is a Visicalc page onto which we have written all the text and formulae needed to do a series of set routines. All the user need to do after this is enter the data every week/month/year to obtain the required results.

First, we have to set the global commands (that is commands which are constant for the whole sheet). Define the format in which you want the values to appear. If you want Pounds and Pence then use /GF$ (any local numbers you want defined as integer (ie whole) can be done when you reach the pertinent field/cell later).

Next type onto the blank sheet the largest value you expect to be working with, take into account the '+' and '-' as they take a character of the column width even if it is not visible. Now you can experiment by typing /GC8. If your value now appears as '>>>>>>>>' it means your column width is too narrow to display the full value. Keep trying with /GC9, /GC10, etc, until the value appears on the screen. Now your column width is set and you can blank out the value you were experimenting with.

When you are building the template, Visicalc can appear to be fractionally slow in taking either text or values. This is due to:-

a) Every time the bottom right hand co-ordinate is extended Visicalc has to re-build the index as the program keeps a record of where everything is stored in memory.

b) Every time you make an adjustment to the sheet or enter data, Visicalc has a vast number of calculations to perform.

To overcome the first problem, make a guess as to where the final bottom right hand co-ordinate of your template will be, extend that by a couple of columns and rows and type anything into that cell/field. Having defined for Visicalc the size of your template anything you place within the defined area will be accepted immediately without Visicalc having to rebuild its index. When you have finished creating the template, blank out that cell, save the template, clear the screen and reload the template. This action clears the unused right hand areas and gives you more memory (this is especially important to remember if you appear to be running out of memory while building the template). The second problem is even easier to overcome. Just type /GRM. Now your sheet is set to manual recalculation. This means that none of the formulae will be updated until you force a recalculation which is done by pressing the '!'. NOTE....Do remember to recalculate before printing a report!!! To reset to automatic calculation, type /GRA.

Save this formatted sheet and then sit down with paper and pencil. Work out roughly which entries are going to be entered as variables (ie the user enters the numbers), and where we want the formulae. At the entry points for variables it is a good idea to put a series of points as a prompt (example 1), the formulae show up as zero's. Example 2 shows the template after data has been entered and a recalculation done.

In this example I have reserved the first 2 columns for text and the next 6 for numeric data. If you have longer labels than me you might like to reserve the first 3 columns for the text.

The 2 Total lines simply add the monthly incomings/outgoings together.

The Surplus/Deficit line takes the total Outgoings away from the total Income.

The first opening bank balance is entered from your bank statement and the first closing bank balance uses the formula '@SUM(surplus/deficit,opening bank balance)'. By using this formula we are ensuring that if we have a surplus it is added to the opening bank balance and if we have a deficit it is taken away from our opening bank balance, or if we have an overdraft at the bank the surplus will be taken away from the overdraft and the deficit would be added onto the overdraft.

The following opening bank balances just replicate the previous closing bank balances and the following closing bank balance formulae are the same as the first formula.

Next month - using @IF, @COUNT, @NPV, @TRUE, @FALSE and @AVERAGE.

EXAMPLE 1

CASH FLOW FORECAST

	JAN	FEB	MAR	APR	MAY	JUNE
INCOME	....	....	....	....	....	....
TOTAL	0	0	0	0	0	0
OUTGOINGS						
OFFICE RENTAL	....	....	....	....	....	....
RATES	....	....	....	....	....	....
WAGES	....	....	....	....	....	....
PERSONAL EXP.	....	....	....	....	....	....
POST/STAT	....	....	....	....	....	....
TELEPHONE	....	....	....	....	....	....
CAR EXPENSES	....	....	....	....	....	....
CAPITAL EQUIP.	....	....	....	....	....	....
MISC. EXPENSES	....	....	....	....	....	....
TOTAL	0	0	0	0	0	0
SURPLUS/ DEFICIT	0	0	0	0	0	0
OPENING BANK BAL.	....	0	0	0	0	0
CLOSING BANK BAL.	0	0	0	0	0	0

EXAMPLE 2

CASH FLOW FORECAST

	JAN	FEB	MAR	APR	MAY	JUNE
INCOME	1500	1000	1000	2000	1500	1000
TOTAL	1500	1000	1000	2000	1500	1000
OUTGOINGS						
OFFICE RENTAL	250	250	250	250	250	250
RATES						350
WAGES	275	275	275	275	275	275
PERSONAL EXP.	300	300	300	300	300	300
POST/STAT	20	20	20	35	35	35
TELEPHONE	75			30		
CAR EXPENSES	20	20	150	20	20	20
CAPITAL EQUIP.				1500		
MISC. EXPENSES	30	30	30	30	30	30
TOTAL	970	895	1025	2490	910	1260
SURPLUS/ DEFICIT	530	105	-25	-490	590	-260
OPENING BANK BAL.	250	780	885	860	420	1010
CLOSING BANK BAL.	780	885	860	420	1010	750

ARE YOUR PADDLES WORN OUT?

Those of you whose paddles may be worn will have found that the Apple pots are of a non-standard value (ie 150 k ohms). These are not available from any normal component suppliers. BASUG has solved your problem ! We have had specially manufactured top quality conductive plastic pots of the right value. They will be available shortly from BASUG at £5.50 per pair including VAT and P & P.

S-C MACRO ASSEMBLER

The **S-C MACRO ASSEMBLER** provides a new level of power and performance for the beginner and professional programmer. A total of 29 commands include an EDIT command with 15 sub-options, and there are 20 Assembler Directives including conditional assembly and macro generation.

The **S-C MACRO ASSEMBLER** operates with any Apple II with at least 32K of RAM and a RAM card version is included allowing source files of over 32,000 bytes! Source files can be assembled directly in memory or can be linked from disc and object code can be stored directly to disc as binary files. The size of your program is limited only by the disc space available. Source files are stored and retrieved using a high speed technique, but standard text files from other assemblers can be read directly into memory for assembly, or you can keep a library of subroutines to load anywhere within a source file.

The **S-C MACRO ASSEMBLER** is supplied on a standard DOS 3.3 disc. Neither the disc nor the files are protected.

The **S-C MACRO ASSEMBLER** supports all DOS and monitor commands. The assembler and editor are co-resident. There are no modes and no changing between command levels; all commands are instantly available.

* **COMMANDS:-**

NEW, LOAD, SAVE, TEXT, HIDE, MERGE
LIST, FIND, EDIT, DELETE, REPLACE, COPY, RENUMBER
FAST, SLOW, PRT, ASM, MGO, VAL, SYMBOLS
AUTO, MANUAL, INCREMENT, MEMORY, MNTR, RST, USR

* **ASSEMBLER DIRECTIVES:-**

.OR origin	.EQ equate	.DO do
.TA target assembley	.DA data	.ELSE else
.TF target file	.HS hex string	.FIN finish
.IN include file	.AS ascii string	.MA macro
.EN end program	.BS block storage	.EM end macro

AVAILABLE AT: £60.00 excluding VAT, £69.00 inclusive from

ELITE SOFTWARE COMPANY
2 ALMORAH ROAD
HESTON
MIDDLESEX TW5 9AD
01-572-0453

dealer enquiries welcome

With this simple tool, and plain English, you can create your own microsoftware

IT SOUNDS EASY . . . AND IT IS!

Gone are the days when business persons needed to learn a complex computer language to write their own application programs, now . . . enter the Codewriter. Dynatech Microsoftware are specialists in the development of program generators such as Codewriter, designed to enable beginners to write their own systems in English.

Also available is Techwriter, a program generator for those with some knowledge of Basic. Techwriter can save up to 90% of program development time in technical and scientific applications.

Write or 'phone for further details.

DYNATECH MICROSOFTWARE LTD

Summerfield House, Vale, Guernsey, Channel Islands.
Telephone 0481 45934 Telex 4191130

Be An Apple Executive

by Martin Rogers.

So you want to be an Executive! An Executive is one who gives orders to his minions and then sits back and observes the action. Apple gave us all the tools to become an executive when we bought our Disk II's, and we booted up our first disk. However, one of DOS's least exploited commands is EXEC. Most users read about it in the DOS manual and try out the sample program on the System Master, but then pass on in bewilderment, never to use EXEC again.

SO WHAT DOES EXEC DO?

When you turn on an exec file with the EXEC command, DOS intercepts all Input requests, and supplies the input from the text file being EXEC'ed. The commands and responses can be Immediate commands, Program lines or DOS commands. If a program is run, the EXEC file will wait patiently until the program ends before issuing its next command.

HOW DO I SET UP AN EXEC FILE?

A text file containing commands to be EXECed is no different from a normal text file. The file can be set up using a text editor such as the editor on the DOS Toolkit or using a word processor such as Apple Writer II. If you do not possess a suitable package, a simple program such as MAKE TEXT on your System master can be used to enter the text but does not have the advantages of editing nor does it allow commas or colons, but is otherwise quite adequate for the task.

HOW MANY EXEC FILES CAN BE OPEN AT ONCE

There can only be one EXEC file open at any one time. If the text file being EXEC'ed contains an EXEC command, the current EXEC file is closed and control is handed over to the new EXEC file. The DOS command CLOSE will not close an EXEC file.

SO WHAT CAN I DO WITH AN EXEC FILE?

The list of applications for EXEC'ed commands is only limited by your imagination and inventive capacity.

The purpose of this article is to whet your appetite by telling you about some of the applications that I have found.

-- Program Modification.

For those of us not fortunate enough to own PLE or GPLE, modifying programs can be a bit of a chore. To make life a little easier, you can save your program as a text file and then use a word processor to edit the program lines. The modified program can then be EXEC'ed back into memory. To save the program as a text file, just add this line to the program -

```
60000  INPUT "YOUR PROGRAM NAME-";ZZ$:D$ = CHR$ (13) + CHR$ (4):ZZ$ = ZZ$ + ".TEXT": POKE 33,30: ? D$;"OPEN";ZZ$: ? D$;"WRITE";ZZ$: LIST 0,59999: ? D$;"CLOSE": POKE 33,40: DEL 60000,60000
```

and then in Immediate mode type

```
RUN 60000
```

and your program will be saved as a text file. Why not set up an EXEC file to do it for you!!!

This technique can be used if you want to append two programs together or to insert those favourite subroutines into the program you are working on.

--Entering Command Strings.

When you find that you are typing in the same old commands to perform a routine task - that is the time to set up an EXEC file of those commands and let DOS perform the chore for you.

e.g. Altering the screen width to let you fast type over text in print statements can be effected by a text file containing

```
TEXT:HOME:POKE33,30
```

which you EXEC when required. (Type TEXT when you have finished to get back to normal screen width!)

--To enter Machine Code routines.

When entering machine code routines from

Magazine articles I always enter them into a text file in the following format using a word processor. The main advantage is that I can easily edit and correct any typing errors if the routine does not work.

```
CALL-151
302:A0 00 A5 69 00 03 A5 6A 8D 01 03 B1
69 C9 02 D0 04 A9 08 91 69 A6 69 CA 86 6A
324:A9 08 C5 6A D0 E4 A9 01 C5 69 D0 DE
AD 00 30 85 69 AD 01 03 85 6A
```

etc.

```
3D0G
```

You can also enter machine code routines into several different memory locations in one step by entering the different start locations at the start of the file line.

(Warning - Do not enter more than 254 characters before pressing return as you will loose the excess characters. This constraint also applies when entering directly through the keyboard, but your word processor may not warn you!)

-- Modifications to DOS

There have been many desirable modifications to DOS published in the various publications available, not least Hardcore. Most advise that the modifications be entered, then a new diskette be INITialised, which will bear the modified image of DOS. This is OK until you find another desirable modification which when keyed in clashes with the memory utilisation of the original patch. The result can be unexpected and possibly downright disasterous.

Another approach is to have a HELLO program on the diskette which POKEs in the patch when the disk is booted. The problem with this is that you have to remember to boot that disk or run the HELLO program which will destroy any program in memory and the values of any variables which you may wish to preserve.

The solution is to hold the coding of the patch in a text file as described above. Then when you want to install the patch, just EXEC it in. EXEC does not affect the program in memory or its variables. (So long as you dont interfere with zero page or program/variable space.)

If you want to get rid of the patch, just boot up to regain standard DOS.

Here is one for you to try. It replaces the MAXFILES command in DOS with a new command FILEDUMP. (Very useful for examining the contents of EXEC files!)

```
CALL-151
BCDF:20 8E FD 20 A3 A2 20 8C A6 F0 05 20
ED FD D0 F6 20 FC A2 60
A8E7:46 49 4C 45 44 55 4D D0
9D48:DE BC
A933:20 30
3D0G
HOME:VTAB10:HTAB11:?"FILEDUMP
INSTALLED":VTAB15:HTAB6:?"FILEDUMP
<FILENAME>,(S6,D1)":VTAB23:HTAB11:?"AN
M.C.R. UTILITY."
```

Type it in and save in a text file called FILEDUMP LOADER. Just enter EXEC FILEDUMP LOADER to instal the routine. If you want to print the file contents, enter PR#1 to initialise the printer and then FILEDUMP <your file name>,Sn,Dn. (The last two are optional). You will then receive a print of each field on the file.

-- To control the running of a series of programs.

This is just the job for those boring maintenance runs where the APPLE is doing all the work and you only run the programs and make routine responses.

Next time you go through the procedure, make careful note of the responses you make. Then enter them into a text file and you can then EXEC the file and the APPLE will run itself while you make the coffee.

-- Entering the data from one program into an existing program.

Have you ever wished that you could enter, say, Invoice details from your Invoicing program into your Sales Ledger

package, but hesitate to modify the package because of maintenance/copyright/complexity restraints? Now you can!

This is just an extension of the preceding application. Make a careful note of all the responses that you make via the keyboard when manually entering the data. Then make the program supplying the data write a text file of these responses. The file will typically contain three sections.

1. Header information concerned with getting the 'package' program up and running.
   RUN <package program name>,S6,D1
   2
   4 (Menu responses
   27.02.83 (Enter today's date
   etc.
2. Many occurrences of actual data to be entered into the program, complete with any Y/N responses confirming correctness.
3. A trailer section which terminates the entry of data, and gets you back to the menu, possibly to run another option (such as an audit print), and finally to exit the package in an orderly fashion. This section would also contain the command to run the next program (possibly a MENU program.)

-- Control of programs requiring non file input.

The EXEC command will supply all input requests with records from the text file being EXECed. This is most inconvenient if you wish to make the occasional contribution to the action.

One way to overcome the problem is to arrange that the text file runs out of data just at the crucial moment, and then have the program EXEC a new file of data when you have had your say. Do not fall into the trap of having an EXEC command as the last entry on the old EXEC file, as the EXEC command will be read by the INPUT statement into your program and will not be 'seen' by DOS. Your program will probably object, and the command will not be executed.

A far better solution would be to turn the EXEC file on and off at suitable points in the program sequence. This is achieved by POKEing 0 or 1 into the EXEC file active flag at $AAB3 in DOS.
This must be done from within the running program with a POKE 43699,0 or POKE 43699,1. It does not seem to work when done in immediate mode. The following simple EXEC file loads and runs a program that demonstrates the technique.

```
SPEED=150:NEW
HOME:VTAB12:HTAB13:?"EXEC FILE DEMO"
10 POKE43699,0:VTAB22:HTAB10:?"<<ANY KEY
FOR MORE>>";:GETA$:POKE43699,1:HOME
RUN
?"NOW FOR A
GAME":?:FORI=1TO40:?"=";:NEXTI:POKE34,5
10 FOR I=1 TO 6
20 A$="YOUR GO":IF PEEK(43699)THENA$="MY
GO"
30 VTAB6:CALL-958:?CHR$(7):?A$
40 VTAB12:?"ENTER A NUMBER AND I":?"WILL
TELL YOU ITS SQUARE...
";:INPUT"";A:VTAB13:HTAB30:CALL-868:?A:A=I
NT(A):B=INT(A^2):VTAB18:?A;" TO THE POWER
OF 2 IS ";B
50 X=PEEK(43699):POKE43699,NOTX
60 FOR J=1 TO 3000:NEXTJ:NEXTI
GOTO10
7
5
9
POKE34,0
HOME:?"HERE IS THE PROGRAM - "
LIST
SPEED=255
?"SO WHERE DID THE INPUT COME FROM??"
```

As you will see when you EXEC the file the program takes some input from the file and some from the keyboard.

There are many more areas of EXEC to be explored yet. In future articles more useful DOS patches will be shown and further techniques for using EXEC files in an EXECutive manner. Let us know what applications you find for the EXEC command.

SNAPSHOT

NOT just another Apple bit copier

SNAPSHOT removes copy protection, and copies most programs that no bit copier can touch – including the bit copiers themselves!

Got the back-up blues? Is Locksmith letting you down? You need SNAPSHOT! SNAPSHOT will copy any memory-resident program that runs on a 48K Apple II. **For example, SNAPSHOT will copy any bit copier now sold, in less time than it takes you to read this paragraph.** SNAPSHOT is a peripheral card that uses your language card* to interrupt a running program and dump the entire contents of 48K and registers to an unprotected, copyable backup disc.

Unlike bit copiers, SNAPSHOT requires no complex parameter changes or trial-and-error tedium. SNAPSHOT is also ideal for debugging your own programs or analysing others' programs. And SNAPSHOT can be used to suspend your work with one program while you use another program. For example, you could interrupt word-processing a letter to look up an address in a database, then resume processing the letter exactly where you left off. Shooting down space invaders and the phone rings? Interrupt your game until later, or tomorrow! Save your high scores! Freeze-frame your game, print the graphics on your Epson, and resume play!

- Repeatedly interrupt and resume any program
- Faster and far easier to use than any bit copier
- Full monitor capabilities to examine, modify, trace, single-step or disassemble any interrupted program, or print graphics on Epson printer
- Copies protected programs from 13-sector to 16-sector discs
- List "unlistable" Basic programs; Make custom modifications
- Suspend work with any program; resume where you left off
- Move protected programs to hard disc or 8" disc as normal files
- Full normal use of your other hardware and software; never open the Apple's lid
- Backups run without SNAPSHOT present; most run without 16K card

And there's more! Write or ring for complete technical information.

PRICE: £95.00

from your local dealer or Dark Star Systems

Terms: Payment with order. Add 15% VAT. P&P included. VISA/Barclaycard and Am Ex accepted. Dealer enquiries invited. Foreign: No VAT. Add £2 postage to Europe, £7 elsewhere.

* One drive and Language Card/16K card required. Works with most popular cards, including Apple, Microsoft, Ramex, Digitek, MPC, RH Electronics, Orange, Franklin, etc. Other brands: specify when ordering.

Dark Star Systems

54 Robin Hood Way, Greenford, Middlesex. UB6 7QNH
Telephone: 01-900 0104

Education

GOING FORWARD WITH LOGO

by Norah Arnold

I was having great fun with Terrapin Logo as I had changed the turtle shape to a mouse. The poor animal was hurtling wildly around off one side of the screen and back on the other as I was playing about in the default mode of WRAP. What was the longest step FORWARD I could ask the mouse to take, I wondered? The command FORWARD 5000 was accomplished easily, so I tried FORWARD 10000, then 15000 and 20000. Things began to be a little unpredictable at around 25000.

I got rid of my mouse and reverted to the ordinary turtle shape. From a HOME state I tried RIGHT 91 FORWARD 10000. Would the turtle find its way HOME? A few calculations led to the discovery that a step FORWARD of 13720 was needed to do the trick. I tried RIGHT 91 FORWARD 13720 HOME LEFT 91 FORWARD 13720. The following procedure was the result.

```
TO DIAMOND :N
 IF :N < 1 THEN STOP
 PU HOME PD
 PC 1 + RANDOM 5
 RT 91 FD 13720
 HOME
 PC 1 + RANDOM 5
 LT 91 FD 13720
 DIAMOND :N - 1
END
```

This is a normal recursive procedure with a STOP rule at the beginning and a final call to the DIAMOND procedure itself. Try DIAMOND 20 and sit back and watch the colour interaction of Apple hi-res graphics. I must admit that I find the many colour shades produced by the interaction quite fascinating. The result of my experimentation with the FORWARD command was the program TAPESTRY, the parameter :N being the number of times that you wish each pattern to reproduce itself.

TAPESTRY needs the procedure DIAMOND above and those below. It could do with a colour control procedure but I rather like the unpredictable nature of the program as it stands.

```
TO TAPESTRY :N
 FULLSCREEN RANDOMIZE
 HT
 CS BG 0 T7 :N
 CS BG 0 T5 :N
 CS BG 0 T2 :N
 CS BG 0 T8 :N
 CS BG 0 T3 :N
 CS BG 0 T4 :N
 CS BG 0 T1 :N
 CS BG 0 T6 :N
 CS BG 0 DIAMOND :N
 SPLITSCREEN
END

TO T7 :N
 IF :N < 1 THEN STOP
 PU HOME PD BG RANDOM 6
 PC RANDOM 6 RT 60 FD 7920
 PU HOME PD
 PC RANDOM 6 LT 60 FD 7920
 T7 :N - 1
END

TO T5 :N
 IF :N < 1 THEN STOP
 PU HOME PD PC 1 + RANDOM 5
 RT 85 FD 13770
 PU HOME PD
 PC 1 + RANDOM 5
 LT 85 FD 13770
 T5 :N - 1
END

TO T2 :N
 IF :N < 1 THEN STOP
 PU HOME PD PC 1 + RANDOM 5
 RT 6 FD 7965
 HOME PC 1 + RANDOM 5
 LT 6 FD 7965
 T2 :N - 1
END

TO T8 :N
 IF :N < 1 THEN STOP
 PU HOME PD BG RANDOM 6
 PC RANDOM 6 RT 31 FD 6000
 PU HOME PD
 PC RANDOM 6 LT 31 FD 6000
 T8 :N - 1
END
```

```
TO T3 :N
 IF :N < 1 THEN STOP
 PU HOME PD BG RANDOM 6
 PC 1 + RANDOM 5
 RT 50 FD 9965
 PU HOME PD
 PC 1 + RANDOM 5
 LT 50 FD 9965
 T3 :N -1
END

TO T4 :N
 IF :N < 1 THEN STOP
 PU HOME PD BG RANDOM 6
 PC 1 + RANDOM 5
 RT 57 FD 5500
 PU HOME PD
 PC 1 + RANDOM 5
 LT 57 FD 5500
 T4 :N -1
END

TO T1 :N
 IF :N < 1 THEN STOP
 PU HOME PD PC 1 + RANDOM 5
 RT 5 FD 9635
 HOME PC 1 + RANDOM 5
 LT 5 FD 9635
 T1 :N -1
END

TO T6 :N
 IF :N < 1 THEN STOP
 PU HOME PD PC 1 + RANDOM 5
 RT 87 FD 13770
 PU HOME PD PC 1 + RANDOM 5
 LT 87 FD 13770
 T6 :N - 1
END
```

The Spelling Problem

THE SPELLING PROBLEM: LEARN HOW TO DO IT!

A review by Graham Davies

Computer magazine editors and readers of software and hardware documentation are subjected to a stream of verbal abuse in the shape of dramatically inventive but yet illiterate spelling. Now matter what the source - England, America, Australia or wherever - programmers and documentation writers fall victim to some mental blockage that effectively screens out the real ways in which "your" (meaning "belonging to you") differs from "you're" (meaning "you are"). Similarly "it's" (meaning "it is") is quite different from "its" (meaning "belonging to it"). Something should be done to eliminate these mental blockages. In answer to the prayers of all my fellow sufferers Glenn M Kleimann of Teaching Tools, Palo Alto, CA, has come up with the ultimate solution, which is brilliant in its simplicity: actually learn how to spell - using their Spelling Package.
To make it clear from the start: this is not a spelling checker (which goes through a WP document supposedly checking for spelling errors and giving you the option of correcting them) but a genuine teaching package employing disk software and tape recorded sound which you (or your teacher) lay down yourself.

One of the distinguishing features of all Teaching Tools educational packages is that the programs derive from a perception of real needs in the classroom situation, and are not applications dreamed up simply because the technology exists for them. In other words, the teacher not the programmer calls the tune
The package comes complete with a simple interface plugging into the games port (the makers wisely recommend the use of an E-Z port to do this and reduce wear and tear on the games socket). This connects the Apple to any off-the-shelf cassette recorder with a 2.5mm remote control socket. This does no more than drive the cassette recorder forward under software control.

cont. on p.33

Europe's top Apple event!

apple '83

Organised by Windfall

The Fulcrum Centre, Slough, June 3-5, 1983

Meet the experts at the Second National Apple Users' Convention

Find out what other people are doing with their Apples at Europe's largest three-day Apple get-together. The very extensive programme of events includes demonstrations of the latest developments in hardware and software, discussions on word processing, databases and financial spreadsheets, and case studies of the effectiveness of Apples in business and education. It's a unique opportunity to share the experience of leading Apple users and hear at first hand how Apples are expanding in new directions.

Accommodation has been made available for delegates at specially reduced rates in the luxurious Post House Hotel, Heathrow.

Included in the price is a seat on the courtesy coach to and from the Apple '83 Exhibition.

	Normal Price	Our Price
Single	£40.00	**£23.00**
Twin/Double	£47.00	**£31.00**
Triple	£52.00	**£36.50**

Prices are per room per night for bed and Continental breakfast including VAT. For only £4.00 extra you can enjoy Executive Club accommodation.

Apply for FREE exhibition ticket

Please send me:

..... free tickets to the Apple '83 Exhibition (Visitors without tickets will be charged an entrance fee of £2.), plus details of the Apple '83 Convention

Name .....

Address .....

.....

.....

POST TO: Apple '83, Europa House, 68 Chester Road, Hazel Grove, Stockport SK7 5NY.

Please enclose a stamped addressed envelope.

cont. from p.31

There is nothing startling about this or in the two pokes used to activate and deactivate the recorder. You also need a 48k Apple with one disk drive, of course.

The software provided consists of everything the teacher needs to type in a list of problematic words, and then record them with his own voice on cassette. The learner listens to a word, types in his version of the word which is then compared with the one in memory. If it is wrong the program re-presents for a second attempt the word as typed but with gaps left for all the incorrect elements. The word is "exploded" by extra spaces to makes it easier for young children to distinguish. The student then only has to type in the missing letters, which are automatically slotted into place.
At the end of the list which can consist of from 1 to 25 items at a time - the learner collects a score and is re-presented with a list of the words correctly spelled and the ones he misspelled. At the writing stage the teacher can specify whether he wants to give the learner a "pattern matching clue" and whether he should be given a second try. That is all these is to it.
The programming was obviously no great problem here, it was written in Applesoft Basic, and why not. The outer packaging is robust for mailing, and the instructions come in the shape of a 5-page A4 typed booklet - no frills but more than adequate for the job.
The great merit of this package is that it gives failsafe, step by step instructions on how to type your words in, save the list and then record them. No control combinations have to be learned and you move forward in each stage by using the RETURN key. I typed in a list of five words commonly misspelt by programmers - recorded them on the crude in-built microphone in my nasal tones . (Serious users would use an outboard microphone under studio conditions). When complete I returned to the menu - again with a simple press of RETURN - and selected the SPELLING PROGRAM. This told me to rewind the tape (a blank cassette for this purpose is included in the package) to where I began. Then - again with simple RETURN presses - the recorder played back my first word and I typed it in. It worked first time, after a familiarization session lasting at most one minute. The ease of operations means that absolutely anyone can use it, and even young childen could set up their own tests with little chance of failure or danger of breaking anything.

So far I have referred to it only as a spelling package for such it is, but in fact it has a variety of applications. The authors stress its flexibility, pointing out that it could be used for giving practice in parts of speech, math terms, foreign words, synonyms or definitions, to name just a few. (In other words, in the over-generous time allocated to each "word" on tape - as much as twenty seconds, you can record anything you like - a question, a snatch of a song, hints, clues, titles, introductions: anything that can be satisfied by a fifteen character answer (it can include spaces) which has to be spelled correctly.

The authors have planned for even greater flexibility than this, however. They describe how to activate the tape recorder through your own BASIC programs and also donate a routine for using the RETURN key for this. You are instructed to take a back-up copy of the disk for your own protection.

The screen presentation is straightforward and unfussy. The attraction of the package is immediate and the program survives very happily without resorting to any of the usual time-consuming bells and whistles and flag-raising ceremonies.
The Apple's upper case only presentation gives all programs a rather dated look, but on the other hand the chances of getting things wrong by being in the wrong case are reduced - and this program"s strength lies in its failsafe robustness .
Of course, you can do something wrong. You can, for instance, play the wrong part of the tape - no signal is laid down on tape in an attempt to synchronize the word list called up from disk with the list you hear. That would have been feasible, of course, but would certainly have led to complications. The solution adopted here requires you to carry out your own tape management, keep track of the numbers on the tape counter, or just use one test per cassette.

One warning: if computer programmers should consider using this program they should under no circumstances type in the words themselves (although, if they are fond of their own voices they could make the recordings since they are not notorious for mispronunciation too). They should employ someone who has a firm grasp of orthography, since otherwise the spelling package would only serve to compound the errors.

The Spelling Package must be considered an essential component of any teaching environment which claims to use CAL with Apple, PET or TANDY (it comes in all three implementations). It overcomes all the objections of language teachers to the use of synthesized voice production in language applications - by simply not using it - the recorded sound used is entirely human and as accurate as the teacher can make it. In essence this is a Speak n' Spell machine - with the difference that the teacher determines the contents and the voice has neither an American nor Dalek's accent. Although I cannot honestly see a really useful application in maths teaching as claimed in the literature, it takes no stretch of the imagination to see it being used in a wide variety of special schools, training establishments, hospitals, any self-study course involving the correction of typed responses to recorded inputs.

The great drawback of the package is the price. At $99.95 (i.e. comfortably over £50 including postage but not VAT) for the disk and tape recorder interface this will put it beyond the reach of all but the most well-heeled primary schools, for instance. Some schools would also have to buy a new cassette recorder with a 2.5mm remote control socket.

/Graham Davies is the author of "Computers, language and language learning", published by the Centre for Information on Language Teaching and Research, 1982. If anyone is interested in more details of the package please contact the editor/

readers' letters

Sunbury on Thames

Dear Sirs,

I have a little problem. I recently retired and acquired the company's old (pre-revision 7) Apple with various hard- and software, including a printer card. This, I remember, was ordered as an Apple Parallel printer card, but we ended up with what we were told was "an ITT card - just the same!" We made do with a photocopy of the "Apple II Parallel Printer Interface Card (A2B002X) Installation and operating manual". This worked well enough driving an Anadex DP 8000 for simple listing.

I now have an Epson MX100 III and want to 'get into' the printer card firmware.

I am not even certain it is an ITT card - the only lettering on it says P.I. PC 8 4, the 4 being inside a circle. Does this mean anything to an ITT buff? Comparing the card with a photo of the Apple "Parallel Printer interface card" in the "Apple II User's Guide" by Poole, McNiff and Cook shows it has the same five chips in the same positions plus an additional 74S474 in one of the positions marked "Spare" on the Apple card.

I would be grateful for any information that is available to explain what this card really is. In particular I will gladly pay for a photocopy of any ITT manual that may be relevant, or at least of the firmware if the commands (CTRL-I etc) are really the same as the Apple's. ITT don't have anything, apparently. Alternatively, if there is a commercial source you might prefer to let me have the address.

Yours &
Ron Teale

(John Sharp replies: For ITT manuals you can contact Arthur Grimwood on [illegible] Controls to the EPSON are in the printer itself and basically any card will send them through, e.g. if you wish to switch on enhanced mode. What matters is getting the character to the printer.)

London SW12

At the moment I am undertaking some research into the dominance of BASIC as the main high level language on microcomputers. I would be interested in corresponding with other BASUG members on this topic. I am particularly concerned to gather evidence on how the present situation has arisen, given that languages like FORTRAN and COBOL have long been available, and how programming languages for micros may develop over the remainder of the decade. I have noted the effect that XEROX's Smalltalk is having both on Visicorp and Apple with interest, but there is also the influence of bodies such as the BBC and the Open University, which would appear to promote the retention of BASIC albeit in a developed form. I would also be interested in any views on whether user groups can or should attempt to influence the policies of manufacturers and software implementers to make changes in the programming environment and, if so, what these should be and how. My address may be released for the purposes of correspondence.

Finally I am slowly building up a library of software including standard Apple products for use on the Videx Videoterm 80 column card. Would you be interested in including them in the software library?

I look forward to hearing from you and, hopefully, from some correspondents on my research project.

Yours sincerely,

M Farlie

[illegible]

(Ed. As for the software library, we include contributions made by members and friendly groups, but of course for the most part not of proprietary software, which might exclude most of what you have in mind - but why not try us?)

Prudhoe, Northumberland

Dear Tony,

With regard to the Menu program on disk 46 I have corrected the free sector count for DOS 3.3.

BLOAD MENU, CALL -151

43E7: 96, BSAVE MENU, A$4000,L$600

Yours faithfully,

M Long

(Ed. Or, if you are afraid of machine code,
BLOAD MENU
POKE 17383,150 (RETURN)
BSAVE MENU, A$4000,L$600)

London NW3

Dear Ray (Harris)

I was very interested to read your article in Hardcore (February 1983) on extending the table of user-defined printer commands in Co-Between.

First of all I was pleased that you were able to buy a copy of the program. I know that a number of BASUG members have had difficulties in convincing the distributors - Apple (UK) - that they should have the program in stock.

Of course, you were at a disadvantage when producing your patches in that you had to work from the object code rather than the source. It may therefore be of interest to you to examine the enclosed two extracts from the source code listings. The first (marked 'Original') shows the unamended code. The second ('Revised') shows how the original can be altered with minor modifications in order to handle multiple-page tables.

There is one further change that needs to be made if you re-allocate the top section of memory, which I should have pointed out in my original article. At $A09 I store $60 in $958E after loading a new file. This ensures that when returning to Teditor, new end-of-file pointers will be set up instead of those relating to any file previously in memory. The address must be changed to $8F9E.

If there is sufficient demand from other readers I will take up your challenge and amend Change constants to allow multiple-page editing of the user's tables.

In the meantime I hope that the above will be of interest.

Yours sincerely

Ian L Trackman

see over

```
                                        ORIGINAL
                  1700
111D: A9 80       1701          LDA   #<USERTBL
111F: 85 77       1702          STA   TEMPLO
1121: A9 17       1703          LDA   #>USERTBL
1123: 85 78       1704          STA   TEMPHI
1125: 20 98 11    1705          JSR   COMPARE      ;ATTEMPT MATCH IN USER'S TABLE
1128: C9 00       1706          CMP   #0           ;NO MATCH AT ALL IF A= 0
112A: F0 25       1707          BEQ   USER6        ;SO EXIT
                  1708                             ;
112C: 86 7A       1709          STX   XSAVE2       ;SAVE AS INDEX >( X = COMMAND
112E: A2 00       1710          LDX   #0                                NO INDEX)
1130: A0 FF       1711          LDY   #$FF
1132: A9 10       1712          LDA   #<USERTXT
1134: 85 77       1713          STA   TEMPLO
1136: A9 18       1714          LDA   #>USERTXT
1138: 85 78       1715          STA   TEMPHI
113A: E4 7A       1716 USER2    CPX   XSAVE2       ;COUNTED ENOUGH ?
113C: F0 08       1717          BEQ   USER4
                  1718                             ;
113E: C8          1719 USER3    INY
113F: B1 77       1720          LDA   (TEMPLO),Y
1141: D0 FB       1721          BNE   USER3         NOT END OF STRING
                  1722                             ;
1143: E8          1723          INX
1144: D0 F4       1724          BNE   USER2        ;ALWAYS
                  1725                             ;
1146: C8          1726 USER4    INY                ;PRINT USER'S CODE
1147: B1 77       1727 USER5    LDA   (TEMPLO),Y
1149: F0 06       1728          BEQ   USER6        ;UNTIL 0 EOT
                  1729                             ;
114B: 20 5C 0E    1730          JSR   PRINT
114E: C8          1731          INY
114F: D0 F6       1732          BNE   USER5        ;ALWAYS
                  1733                             ;
1151: A4 7C       1734 USER6    LDY   YSAVE2       ;SAVED IN COMPARE
1153: 60          1735          RTS                 ( Y = POINTER TO MAIN TEXT
                  1736 *                                  PLACE)
```

Higher Text Extended

Peter Trinder

Higher Text II marketed by Synergistic Software is for those Apple Owners who like to create TEXT displays and is a very comprehensive package. It includes 13 different chartacter fonts containing both normal and large sized characters. The normal fonts can be displayed in TALL or WIDE form. An identical programme is available from Call A.P.P.L.E (Higher Text Plus) for members.(*)

This programme is very comprehensive but because it has a lot of Control (key) sequences it isn't the 'friendliest' to use in its published 'STANDARD' form without reference to the manual. I have found a programme called HIGHER TEXT EXTENDED (H.T.E.) which patches a machine language programme onto Higher Text to update it. Full and fool proof instructions are given for doing this so novices should find no difficulty in updating their Higher Text discs.

```
:ASM                                    REVISED
                                                        ;
111D: A9 80                     LDA   #<USERTBL
111F: 85 77                     STA   TEMPLO
1121: A9 17                     LDA   #>USERTBL
1123: 85 78                     STA   TEMPHI
1125: 20 A2 11                  JSR   COMPARE
1128: C9 00                     CMP   #0              ;NO MATCH AT ALL
112A: F0 2F                     BEQ   USER6           ;SO EXIT
                                                        ;
112C: 86 7A                     STX   XSAVE2          ;SAVE AS INDEX
112E: A2 00                     LDX   #0
1130: A0 00                     LDY   #0
1132: A9 10                     LDA   #<USERTXT
1134: 85 77                     STA   TEMPLO
1136: A9 18                     LDA   #>USERTXT
1138: 85 78                     STA   TEMPHI
113A: E4 7A           USER2     CPX   XSAVE2          ;COUNTED ENOUGH ?
113C: F0 0E                     BEQ   USER4
                                                        ;
113E: B1 77           USER3     LDA   (TEMPLO),Y
1140: F0 07                     BEQ   USER3A          ;EOT ?
                                                        ;
1142: C8                        INY
1143: D0 F9                     BNE   USER3
1145: E6 78                     INC   TEMPHI
1147: D0 F5                     BNE   USER3           ;ALWAYS   AS THERE IS ROOM FOR
                                                        ;         THE EXTRA CODE, TAKE
1149: E8              USER3A    INX
114A: D0 EE                     BNE   USER2           ;ALWAYS   THIS OUT TO, SAY, $1600.
                                                        ;
114C: B1 77           USER4     LDA   (TEMPLO),Y                  BJQ USER_6 BECOMES BEQ
114E: F0 0B                     BEQ   USER6           ;EOT ?    USER4A AND ADD
                                                        ;         USER4A RTS.
1150: C8                        INY
1151: D0 02                     BNE   USER5                     114F/50 ARE REDUNDANT.
1153: E6 78                     INC   TEMPHI
                                                        ;         THEREFORE:-
1155: 20 5C 0E        USER5     JSR   PRINT                     114C JSR $1600
1158: 4C 4C 11                  JMP   USER4                     114F NOP
                                                        ;         1150 NOP
115B: A4 7C           USER6     LDY   YSAVE2          ;SAVED IN COMPARE
115D: 60                        RTS
                      *
```

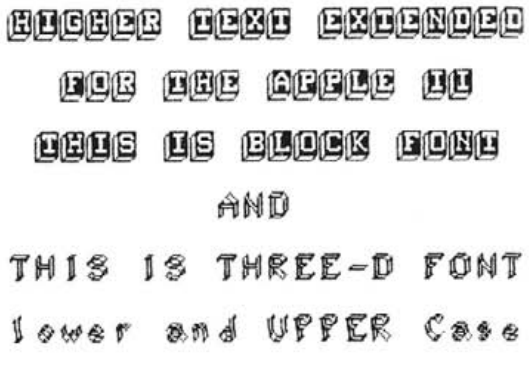
H.T.E. provides help menu's so that you don't have to rummage around the manual. It provides right, centre, and left justification of text on a line-by-line basis. It prevents accidental scrolling of the text off the screen. H.T.E. has an easy load/save/catalog menu which allows for change of disc drive with one keystroke. When creating a picture a Control Shift M will save the current screen (not to disc) and you can then mess about with your drawing, if you don't like it a Control Shift N will restore the original screen.

The H.T.E disc also has a number of frames to provide pretty borders around pictures.

It is sold by C & H Video, 110 W. Caracas Avenue, Hershey, Penn 17033 U.S.A. H.T.E. costs $22.50. I have included a sample screen produced using Higher Text.

* (A disc called 'HIGHER FONTS' is available from CALL A.P.P.L.E has a number of additional fonts to add to the one's on the Higher Text Disc, there may be a commercial version from Synergistic Software but I can't confirm this myself although I am sure I have seen it advertised in American Magazines.)

(see page 38)



The WIDE and EXPANDED characters can be printed in the normal six colors.

WHITE GREEN
VIOLET ORANGE
BLUE BLACK

The EXPANDED characters can also be printed in these ADDITIONAL COLORS.

YELLOW BLUE VIOLET
PINK AQUA

HIGHER TEXT HAS SIX CHARACTER TYPES.

REGULAR
TALL
BOLDFACE
WIDE
EXPANDED
LARGE FONT

BASUG Disk No.46 Enhanced

by D.J. Bullar

The BASUG disk No 46 is a super collection of useful bits and pieces, but I found one mod desirable. If I have not got it wrong, others may benefit from it. The particular set of programs concerned are:

APPLESOFT TO TEXT
APPLEWRITER TO TEXT
INTEGER TO TEXT
TEXT TO APPLEWRITER

These programs enable one to edit BASIC programs with Apple Writer and to transfer INTEGER programs into APPLESOFT (provided the language mods are made).
Now APPLEWRITER TO TEXT is only supplied on this disk in INTEGER but I only have Applesoft (or rather Palsoft) in firmware so I have to BRUN some copy of Integer to use this program. Apart from being a bit of a pain to do each time, I am always suspicious of memory clashes when doing so. sure enough it worked on some files but not on others! So the obvious answer is to use the facility to transfer the Integer version of "APPLEWRITER TO TEXT" into Applesoft by saving it as a text file and "exec"ing it back into Applesoft. (I have included precise instructions for this in case members have not met this technique before.)

The various language differences were sorted out, and are detailed, and the program run. Still it "bombed out", and the reason seemed to be that the "Applewriter" text was being loaded across the space required by the basic program. How and why this does not matter in the Integer version I do not know. Nevertheless by loading the "Applewriter" text just a bit further up the memory and modifying the machine code bit of the program it now seems to work well.
Of course loading "Applewriter" text further up means that slightly less text can be accommodated, but it is better to work satisfactorily with room for 10k of text than not to work at all but to have room for 11k!!

TO CONSTRUCT "APPLESOFT" VERSION OF "APPLEWRITER TO TEXT" (BASUG DISK 46)

1. BRUN some version of INTEGER
2. LOAD "APPLEWRITER TO TEXT"
3. EXEC "INTEGER TO TEXT"

You should now finish up at the end of this with a text file on your disk called "LISTING"

4. Get rid of the INTEGER BASIC by re-booting the disk.

5. EXEC "LISTING"

You should now have a copy of the integer program "APPLEWRITER TO TEXT" in memory. Applesoft will find that it is full of "SYNTAX ERRORS" but these can be corrected as below:

a) Delete "TAB8:" in line 5
b) Change ",OP" to ";OP" in line 95
c) Delete "TAB8:" in line 125
d) Change ";FI$" to ";FI$" in line 125
e) Change "A$900" to "A$D00" in line 130
f) Delete line 110 and replace it with 110 POKE 773,13

Be careful about D$... I prefer to change line 0 to say D$=CHR$(4). Then it "screen copies" correctly.

see over for modified listing

MODIFIED LISTING OF APPLEWRITER TO TEXT

MODS EMPHASISED

```
0  TEXT : CALL  - 936:D$ = CHR$ (4): PRINT :
   PRINT D$;"BLOAD APPLEWRITER TO TEXT.OBJ"
5  PRINT "APPLEWRITER FILE TO TEXT"
10  PRINT : PRINT
15  PRINT "THIS PROGRAM WILL TAKE A FILE CREATED"
20  PRINT "BY APPLEWRITER AND CONVERT IT INTO A"
25  PRINT "A TEXT FILE. THIS ALLOWS YOU TO DO THE"
30  PRINT "THE FOLLOWING WITH APPLEWRITER:"
35  PRINT "A.  WRITE 'EXEC' PROGRAMS."
40  PRINT "B.  WRITE PROGRAMS AND 'EXEC' THEM IN."
45  PRINT "C.  TAKE A PROGRAM IN APPLERWRITER AND"
50  PRINT "    VERIFY THEM BEFORE PUBLICATION."
55  PRINT "D.  PREPARE DOCUMENTS TO BE TRANSMITTED"
65  PRINT "    FORTRAN PROGRAM  ECT."
70  PRINT : PRINT "THE OPTIONS ARE"
75  PRINT "1.   CONVERT TO UPPER CASE ONLY"
80  PRINT "2.   NO CONVERSTION"
85  PRINT "3.   CONVERT TO UPPER AND CHANGE"
90  PRINT "     &&<CHR> TO A CONTROL CHARACTER."
95  PRINT : INPUT "INPUT OPTION NO. ";OP
100  IF OP > 4 OR OP < 1 THEN 70
105  POKE 6,OP - 1
110  POKE 773,13
115  PRINT D$;"CATALOG"
120  PRINT "THE NAME OF THE FILE TO BE CONVERTED?"
125  INPUT "TEXT.";FI$: IF  NOT  LEN (FI$) THEN 115
130  PRINT D$;"BLOAD TEXT.";FI$;",A$0D00"
135  PRINT D$;"OPEN";FI$
140  PRINT D$;"DELETE";FI$
145  PRINT D$;"OPEN";FI$
150  PRINT D$;"WRITE";FI$
155  CALL 768: REM  CALL BLOADED PGM
160  CALL  - 958: REM  CLEAR TO END OF PAGE
165  PRINT : PRINT D$;"CLOSE"
170  PRINT "FINISHED": PRINT "TEXT FILE NAME IS ";FI$
175  END
```

Workshops

COMMUNICATIONS WORKSHOP

by John Sharp

The twenty five members who met at Cambridge on March 14 had an enjoyable and informative day out.

Mike Gardner of OWL Communications gave an overview of the terms used and the principles involved in sending information from one computer to another, particularly over a telephone line. The talk was so popular that only the threat of missing lunch was able to break up the session. The problem that came over strongly was that British Telecom's monopoly has held back the use of modems for communication, compared with the USA, with the result that the choice is limited to expensive modems from BT. With MICRONET and PRESTEL seeing the potential this looks set to change, and judging by the enthusiasm of those at the meeting, there could be as many micros talking to one another as there are people nattering in the evenings before very long.

Leon Heller of the TRS 80 user group gave a demonstration and talk on Bulletin boards. The use of micros to send mail and pass and store information is being given a lead by such enthusiasts as him. In the same way as Steve Wozniak and Steve Jobs paved the way for the personal computer with the first APPLE so these bulletin boards are the heralds of something bigger around the corner.

John Sharp gave a short talk on MICRONET and the impact of PRESTEL on the micro, followed by an on line demonstration. This covered a tour of some of the information pages, and then downloading of a telesoftware program.
After a short demonstration of the OWL editing package for PRESTEL, and other gatherings round the APPLES, the group convened to discuss what BASUG members want. The problem seems to be the possibilities are so enormous it is difficult to know where to start. If the wrong decisions are made too hastily the mess will be hard to sort out, so another session of those interested in this particular area will be called in the near future.

Beginners' page

by John Sharp

CONCLUDING DEBUGGING?
This follows on from previous articles and is a conclusion to the set, but not an end. Everyone has their own private tricks. I no longer make all the mistakes I did, so some vital problems may be missing.

Try the following program:-

```
70 FOR N = 1 TO 8
80 Q(N) = N
90 NEXT N
```

and RUN it.
Since there is nothing output to the screen then there is no visible evidence that the program has done anything. The array Q() has however been filled. This happens even when other parts of the program are writing to the screen. It is possible to find out if variables are what you think they are by inspecting them in immediate mode just by typing in from the keyboard, so that for example if you type:-

```
PRINT Q(3)  <RETURN>
```

3 is the result.

We can also use variables so

```
PRINT Q(N)
```

is just as valid. Try it and see that the following is the result

```
PRINT Q(N)
 0
```

So what is this? Surely since the loop has gone around from 1 to 8, N has become 8 why do we not get the value 8?

```
PRINT Q(8)
 8
```

is OK. So try

```
PRINT N
 9
```

NOTE !!! When you come out of a loop the value of the variable you are changing has always gone one more step than you specified. When the program reaches the NEXT in line 750 the value of N is increased and then it goes back to the beginning of the loop at line 700 and tests if the value of N is greater than 8. Only if it is not will it go through the loop again, otherwise it will try to find the line following the NEXT statement, i.e. a line numbered greater than 750.

It is sometimes the case that when running a program, you realise it is not going the way you thought. In this case you can break the program by pressing <CTRL-C> and try to sort it out from there.

Add line 75 as follows:-

```
75 FOR DLY = 1 TO 500: NEXT
```

This is a very useful general line. It is a time wasting statement, simply sending the program round and round 500 times so that time is used up. In this case it is there to make the program go slower so that we can break the program before it has finished. You can see the delay as follows :-

```
TRACE
RUN
```

Lots of #75s are generated.
Having switched off the trace function with NOTRACE, RUN the program again and when it has gone a few rounds of the outer loop, but before it has finished (you will have to guess the timing) press <CTRL-c> The message:-

```
BREAK IN 75
```

comes up.
This is a very important aspect of debugging. A program may appear to have gone away - technically known as 'hanging'. If you press <CTRL-C> then it will tell you that you have stopped at a

particular line. If you now type

```
CONT
```

it will CONTinue where it left off, without destroying the value of any variables as RUN would.

In practice you would probably print out some of the variables in immediate mode in the meantime. This would in this case for example yield the following information:-

```
RUN
BREAK IN 75
?N
 3
CONT
```

and you know you have broken into the program when the outer loop had a value of N=3. The value of N was preserved and it was able to carry on even though you had printed the value of N. You can even alter the value of N and still allow it to continue.

```
RUN

BREAK IN 75
?N
 3
N=6
CONT
```

This means that you would not have filled the array when N was equal to 4,5 and 6. To show this type the following in immediate mode:-

```
FOR N = 1 TO 8: ? Q(N);:NEXT

 1  2  3  0  0  0  7  8
```

The line containing the FOR ... NEXT loop should be on one line without pressing <RETURN> until the end of the line. Do not forget the ";" so that the values are printed in one line, or you will possibly scroll off the screen.

Now alter line 80 to

```
80 Q(N) = M
```

and add

```
60 M = 49
```

and run the program again, without breaking and repeat the last exercise :-

```
FOR N = 1 TO 8 : ? Q(N);:NEXT

 49  49  49  49  49  49  49  49
```

This time since M is fixed by the previous line to 49, all the values of the array are the same.

If this was an error of typing, missing the N key and hitting the M instead, you would have a clue instantly especially if you then printed M and found it to match.

One other way to pick up this type of error, i.e. of getting all the values wrong without at first knowing why would be to go straight to the appropriate loop directly. This can be done in two ways:-

```
RUN 70 or GOTO 70
```

They are both valuable tools. Suppose we carry out the same exercise and print the value of the array variables as before, using RUN 70, the result would be as follows:-

```
RUN70

FOR N = 1 TO 8 : ? Q(N);:NEXT

 0  0  0  0  0  0  0  0
```

The value of M has automatically been set to zero by the RUN command.

The command GOTO does not zero all variables. Try the following using <CTRL-C> key to halt the program after the value of M has been set, and see that the array variables are all set to 49 again

```
RUN
BREAK IN 75
GOTO 70

FOR N = 1 TO 8: ? Q(N);:NEXT
 49  49  49  49  49  49  49  49
```

In fact, just to prove the point, the alteration of the variable first could be done as follows:-

```
M = 99
RUN 70
```

```
FOR N = 1 TO 8: ? Q(N);: NEXT
 0  0  0  0  0  0  0  0
```

or as follows

```
M = 99
GOTO 70
```

```
FOR N = 1 TO 8 : ? Q(N) ; : NEXT
 99  99  99  99  99  99  99  99
```

The same techniques can be used with strings. If the string variable has not been assigned, it will not of course print anything. So the following holds

```
? A$ : ? B$

FRED
```

means that A$ was empty (a blank line was printed) and B$ had been set to FRED.

CONCLUSION: This is not the final word in debugging, but a summary of some of the "tricks of the trade". Most you learn yourself, the hard way. Any more ideas would be welcome, even by some of the older hands, especially since, as I said at the beginning, I have forgotten some mistakes I used to make.

Seedlings

BASUG members seeking fame (but not fortune!) in the national press could try sending listings to Personal Computer News. Games, utilities, you know the kind of thing. The reason we include this is because Nigel Cross, the programs editor of the new glossy weekly asked to pass the message on to our members. So here goes: if you have anything for the world on the Apple or any other machine, send it to:

Nigel Cross
Program Editor
Personal Computer News
62 Oxford Street
London W1
Or try telephoning 01-636 6890 (Ext 268).

(Be sure to to get the magazine's name right, but if you don't, it's not your fault.)

Some unused cerebral whorl in your editor's skull led him to single out an item in Wayne Green's ego trip publication Selling Micros, concerning a new product by Scott Instruments of Denton, Texas. In response to his letter Scott have sent him a pack of information on their Shadow/Vet, a voice entry terminal for home, education, business, finance and industry. This claims to be able to use most existing software under voice control without modification. "Solve Visicalc forecasting problems more efficiently, draw Graphs with Apple Plot, obtain information from The source," etc. On the education front, the blurb says, "Questions and answers are easily entered into the system. The student reads a question on the computer screen and speaks the answer into a microphone. The computer immediately informs the student if the answer is correct... Computer game lovers "haven't lived" until they have played the Adventure game series by voice. A severely handicapped person can control his or her environment or program a computer by voice. For example, lights can be turned on by saying "Lights on." The system which comes in a little box allows you to build up a 40 word set 'with overlays' for access to additional sets of 40-word vocabularies.

Your editor comments: in the pioneering days of BASUG Frank Kay demonstrated speech inputting devices with a similar word storage capacity but using synthesized or digitized speech. Speech and voice recognition at that time was in its infancy and the general opinion was that if, say, you programmed a security system to recognise your voice saying "open the door, sesame", if next week you came down with laryngitis you would stay locked out. Scott now claim that the VBLS (Voice Based Learning System) can understand any language, including utterances resulting from speech impediments. Only 'voice-on' experience will say whether Scott have really succeeded in overcoming that hurdle. If so this would constitute a major breakthrough and indeed "make it ideal for bilingual and foreign language study or special education." Included in the wodge of explanatory material provided by Scott is the candid information that their system uses existing technology and operates on individual words only.

"Continuous speech recognition" making the "automatic secretary" possible lies somewhere in the 1990s. This frankness about the realities of their technology suggests to me that Scott are the kind of outfit likely to provide good customer support.
Your intrepid editor has written to Scott asking for the whole package to be lent for evaluation by BASUG members, but that really is a bit of a tall order in view of the fact that the Shadow/Vet (complete with voice operational software, manual and noise-cancelling microphone) retails at $995.00 in the States. Stayed tuned in to this column.

What is MICRONET?

by John Sharp

The PRESTEL service has been of limited general success. The two most acessed ares are financial information and travel. For the average user there is a great deal of miscellaneous information, but it is not always easy to find. The pages on PRESTEL are accessed in a menu driven way by sometimes long and tortuous routes which unless the system is used proerly can be seen once never to be found again. If you have not seen PRESTEL, it is available in some libraries and other public areas. There is a similar service being sent along with the television pictures on both BBC and ITV. The information is sent at the top and bottom of the frame. These information frames are only available with a special decoder in the set. The BBC often shows a selection of the frames from the CEEFAX services, before transmission starts as normal pictures, so you can gain an idea of the type of information and display. The PRESTEL and CEEFAX/ORACLE systems are different, in that PRESTEL is interactive whereas the TV systems are not.

The marketing of PRESTEL has been given a boost by the concept of telesoftware. This is the carrot to persuade micro users to subscribe to the service. Programs are coded in computer readable form and can be 'downloaded' from PRESTEL then run. The whole of PRESTEL is also available for the micro user as a bonus. A special company has been set up as a consortium, called MICRONET.

MICRONET is an area on PRESTEL which is a club to which you have to subscribe in order to see all the information, including the software. Information may be enclosed with your HARDCORE -we were still negotiating at the time of writing. Some members have said that they have had difficulty in getting information from MICRONET. I have asked them why and they have had over 12000 requests recently and so have had a little backlog. One person at the meeting said he asked for specific information on CP/M and MICRONET but only got the standard brochure. With 12000 applications would you answer every one specifically?

MICRONET have given all the user groups a number of pages free. BASUG has been involved because John Sharp and David Bolton have been working with PRESTEL and MICRONET in a private capacity. Some information pages have been put up by them, but there has not been the time to keep them up to date. More help will be required and the members who have written in will be contacted to arrange a meeting. Please be patient as the direction that is going to be taken needs to be mapped a little first. Any suggestions would be appreciated.

The areas that MICRONET/PRESTEL can be used for are software, and information exchange. This could be VISCALC files or text files, or straight hints and tips type information or electronic mail. At the moment there is no search facility, but there are plans to achieve this by going through a 'GATEWAY' to an external computer, and then coming back with a list of pages to look at. What sort of DATABASE would you like to see? Does anyone have experience of the SOURCE which is the US micro-compatible system?

It is difficult to be more forthcoming in such a short space, a demonstration is a better way to go.

The latest news is that the latest APPLE connection of an accoustic modem is due sometime in APRIL. It may be better to wait. MICRONET are producing a modem which will have not just the PRESTEL 1200

AUTOCRAT COMPUTERS

SPECIAL OFFERS

Disk storage box: capacity 60 5.25" disks. Lockable. Perspex top £19.95

Paper Tiger 445 printer (shop-soiled). Below cost £375 + VAT

LAST MINUTE SPECIAL:
ANOTHER AUTOCRAT FIRST
Epson's Latest Printer
FX80 Friction/Tractor
160 cps + Special Quiet Mode
Guaranteed lowest price
£389.00.

Vlasak Apple workstation. Takes Apple, two disks and Hitachi 9" Monitor. Apple coloured all-steel construction. Single power input for all needs. Single key switch.

Usual £130
One only price £79

AUTOCRAT COMPUTERS,
264 Preston Road,
Harrow, Middx., HA3 0PY.
Tel: 01 908 3636

VAT EXTRA

receive/ 75 send, but a 1200/1200 switchable system so that two micros can talk as well- with suitable software. This was a necessity as far as the members at the meeting were concerned, so don't rush to join before the full facts are available. We will keep you posted.

ADVERTISERS

2 Lux Computers Services Ltd.
Systemics Ltd.
Meekrose Ltd.
6 Wida Software
Mass Micros
21 Small Ads
24 Elite Software Company
25 Dynatech Microsoftware Ltd.
29 Dark Star Systems
32 Windfall
45 Autocrat Computers
47 Pynwon Software
48 Pete & Pam Computers

Advertising rates

Full page	£50.00
Half page	£27.50
1/4 page	£17.50
Preparation of artwork	£ 5.00
Flysheets	£75.00

Please send complete camera ready artwork in monochrome. If in A4 size then typeface must tolerate photographic reduction to A5. Hard Core can undertake minor alterations to copy.

Editorial and advertising copy to 2, Nicholas Gardens, London W5 5HY.

DIARY

April

5th Herts Group Forth.
9th BASUG Graphics Workshop. Central Institute, Longford St, London NW1
9th BASUG Visicalc course.
11th BASUG Hants & Berks Local group, Reading University
11th BASUG Croydon Apple User Group
12th BASUG East Kent Group
14-16th 4th London Computer Fair. Westminster Hall
14th South London Group Visicalc & AGM
21st British Computer Society "Hype and the Micro-Computer" Ian McNaught-Davis (Charing Cross Hotel)
28-30th Midland Computer Show Birmingham

May

3rd BASUG Herts group (Complete Graphics System)
9th BASUG Hants & Berks Local group, Reading University
9th British Computer Society AGM & "Creating a Total Information Technology Environment" David Firnberg, Deputy President, BCS (BCS Headquarters)
10th East Kent group
12th South London Group External Interfacing with process control

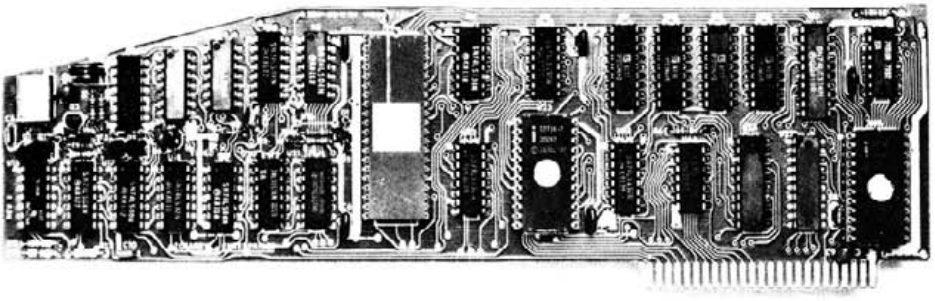
June

3-5th Apple 83 Fulcrum Centre, Slough
7th Herts group Plotters
9th South London group - Graphics
13th Croydon Apple User Group
16-19th IPC show Earls Court, London

80 COLUMNS FOR IIe —

CHEAPER THAN APPLE ?

This card is fully compatible with Apple's latest 80 column board. However, for the same price as Apple's, this board has 8 spare sockets ready to accept 64K of RAM. Works with all software that works with the new IIe 80 column display. A sure winner at £80 RRP.



VISION-80 80x24 Video Display Card

The VISION-80 stands apart from all other cards as the "ROLLS ROYCE" of 80 column cards, as confirmed by every magazine review. Our Rev 3 and Rev 5 boards are fully compatible with the Apple IIe and II, giving a superb 9 x 11 dot matrix, full lower case descenders and inverse lower case as well as upper case. This is the only 80 column board that supports Apple's text window and text manipulation commands. It works with Pascal and CP/M, and the following software: WORDSTAR 3.00, APPLEWRITER II, LETTER PERFECT, FORMAT-80, ZARDAX, EXECUTIVE SECRETARY, MAGIC WINDOW, VISICALC, MULTIPLAN (CP/M or DOS).

VISION-80 card.....£195
VISION-80 AWII (AppleWriterII pre-boot.....£18
ROSETTA Pascal to DOS utility.....£35
Screenmaster 80 column card.....£185
VISION-80 VC pre-boot 80 col Visicalc.....£29.50
VISION 80 VC EXPANSION 80 col Visicalc and 143K using all popular 128K RAM cards..£39

Please contact your local Apple Dealer, or PYNWON for the address of one of our Dealers closest you

---

UK Dealers please contact: PYNWON SOFTWARE 17 Watermill Lane Edmonton N18 1SU London 884 0879

SCOTLAND	N.IRELAND	SWITZERLAND	NETHERLANDS	HONG KONG	AUSTRALIA	USA
Scotbyte	IC Computing	Lemane CompSA	Aramco	Delta Comms	Imagineering	Coex
031-3431005	0265-53155	021-291516	070-761716	3-699241	02-3583011	714-5583972

TRIPLE YOUR DISK ACCESS SPEED

with

FastDOS

FAST DISK OPERATING SYSTEM FOR APPLE II COMPUTERS

Completely compatible with DOS disks

Loads and saves standard DOS files

Completely compatible with all DOS/APPLESOFT programs that access DOS through standard hooks, including FID and MUFFIN

Executes all standard DOS commands

Comparative Timings:	DOS	FDOS
Loading integer basic	13 sec	3 sec
Cataloging a 12 file disk	2 sec	1 sec
Saving a 10 sector program	6 sec	2 sec
Saving a100 sector program	34 sec	7 sec
Loading a 100 sector program	24 sec	7 sec

Requires 48K

Retail Price £19.95

AVAILABLE FROM

Head Office:
New Hall Hey Road,
Rossendale, BB4 6JG
Phone: (0706) 227011
Telex: 635740 Petpam G

London Office:
103-5 Blegborough Road,
London, SW16 6DL
Phones: 01-769 1022/3/4 & 01-677 2052
Telex: 923070 PPCOMP G