FEBRUARY 1992 £1.95

SINCLAIR QL WORLD

Software File:

FIVE LIVE GAMES

Five game disk

Hardware User Report

CQV1 Digitizer

JUST FOR STARTERS

SuperBasic database for new programmers

M.C.M.
QUALITY
EDITORIAL

02
9 770951 933016

DJC Dilwyn Jones Computing

41 Bro Emrys, Tal-y-Bont,
Bangor, Gwynedd LL57 3YT
Tel: Bangor (0248) 354023

SOFTWARE

ADDRESS LABELLER(flp, mdv, 384k, R)	£15.00
BASIC REPORTER (flp, 128k)	£10.00
BASIC REPORTER (mdv, 128k)	£12.00
BIBLE TEXT DISKS (flp, Quill format)	£20.00
BIBLE TEXT DISKS (flp, Editor format)	£20.00
COCKTAILS WAITER (flp, 256k)	£10.00
COCKTAILS WAITER (mdv, 256k)	£12.00
Cocktails waiter recipes sets:	
MIX 2 (flp, mdv)	£5.00
MIX 3 (flp, mdv)	£5.00
DATA DESIGN, by PROGS V2.03 (flp, 512k)	£50.00
DISCOVER (fllp, 256k, dual drives)	£20.00
MULTI DISCOVER (flp, 256k, dual drives)	£30.00
TEXTIDY (flp, 256k)	£15.00
FLASHBACK (flp, mdv, 256k)	£25.00
FLASHBACK SPECIAL EDITION (flp, 256k)	£40.00
Upgrade FLASHBACK to FLASHBACK SE (return master copy)	£20.00
HOME BUDGET (flp, 128k)	£20.00
HOME BUDGET (mdv, 128k)	£20.00
FILEMASTER (flp, mdv, 512k, R)	£12.00
FILES 2 (flp, mdv, 128k)	£12.00
IMAGE PROCESSOR V2 (flp, mdv, 256k)	£15.00
PAGE DESIGNER 2 PLUS (flp, 512k)	£40.00
Upgrade PD2, return master copy	£20.00
PD2 CLIPART, Screen Format (flp, 128k)	£10.00
PICTUREMASTER (flp, 256k)	£15.00
PICTUREMASTER PLUS (flp, 256k)	£20.00
Upgrade PICTUREMASTER to PLUS version (quote serial number)	£5.00
QL GENEALOGIST 2ND EDITION (flp, 384k)	£30.00
QL GENEALOGIST 1ST EDITION (flp, 384k)	£19.50
Upgrade QLG1 to QLG2 (return master)	£12.00
BUDGET QL GENEALOGIST (flp, mdv, 128k)	£12.00
Upgrade Budget QLG to QLG 2nd Edition (return master)	£24.00
QLIBERATOR COMPILER (flp, 256k)	£50.00
BUDGET QLIBERATOR (flp, mdv, 128k)	£25.00
QLOAD/QREF (flp, mdv, 128k)	£15.00
QUESTION MASTER (flp, 128k)	£10.00
QUESTION MASTER (mdv, 128k)	£12.00
QUIZ SETS for Question Master. Each	£5.00
1. General Knowledge 2. Classical Music 3. QLestions (Questions about the QL)	
QUICK POSTERS (flp, 128k)	£10.00
QUICK POSTERS (mdv, 128k)	£12.00
REMIND-ME (flp, mdv, 128k)	£12.00
ROB ROY BARGAIN PACK (flp, 128k)	£10.00
ROB ROY BARGAIN PACK (mdv, 128k)	£14.00
SCREEN SNATCHER (flp, mdv, 128k)	£12.00
SCREEN ECONOMISER (flp, 128k)	£10.00
SCREEN ECONOMISER (mdv, 128k)	£12.00
SLOWGOLD 2 (flp, mdv, 128k)	£5.00
SPELLBOUND (flp, mdv, 256k)	£30.00
SPELLBOUND SPECIAL EDITION (flp, 512k)	£50.00
Upgrade Spellbound to SE (return master)	£30.00
SUPER DISK LABELLER (flp, 256k)	£10.00
SUPER DISK INDEXER (flp, mdv, 256k)	£12.00
TASKMASTER (flp, mdv, 384k)	£25.00
Upgrade TASKMASTER for Hard Disks (return master copy)	£10.00
THE CAT (flp, 128k)	£5.00
THE CAT (mdv, 128k)	£5.00
THE CLIPART, by PROGS (flp, mdv, 128k)	£12.00
THE PAINTER V4.03, by PROGS (flp, 512k)	£25.00
THE GOPHER (flp, mdv, 128k)	£12.00
TRANS24 (flp, 128k)	£10.00
TRANS24 (mdv, 128k)	£12.00
VISION MIXER (flp, 384k)	£10.00
VISION MIXER (mdv, 384k)	£12.00
VISION MIXER PLUS (flp, 256k)	£22.50
Upgrade VISION MIXER to VISION MIXER PLUS (quote serial number)	£14.50
WINBACK (flp, 256k)	£25.00

NB 'flp' = disk, 'mdv' = microdrive, 'R' = RAMdisk reqd. 128k/512k etc. = minimum memory reqd.

LABELS

MICRODRIVE LABELS roll of 100	£2.00
DISK LABELS roll of 100	£2.00
DISK LABELS roll of 100 on printer pin feed backing	£2.50
ADDRESS LABELS printer roll of 100 3.5" wide, 1.5" high	£2.00

Please add 50p postage per roll of labels (max. postage £2.50) if only orderning labels.

FLOPPY DISKS, ETC.

3.5" unbranded DSDD disks	£0.45
20 or more disks	£0.40
Pack of 10 SONY branded 3.5"	£9.00
5.25" DSDD unbranded disks	£0.35
Microdrive cartridges (new)	£2.50

N.B. PLEASE ADD POSTAGE SPECIFIED BELOW. NO NEED TO ADD POSTAGE ON MICRODRIVE CARTRIDGES IN UK.

DISK STORAGE BOXES

Posso boxes, drawer style, holds 150 3.5" disks, stackable, storage system	£17.00
Box to hold 80 3.5" disks, with lockable, clear lid	£7.00
Box to hold 40 3.5" disks, with lockable, clear lid	£5.00
Slimpak case, small plastic box, holds 10 3.5" disks	£1.20
Set of 20 headed disk box subject dividers, "dry wipe" laminated, assorted headings one side, blank other side	£3.00

POSTAGE RATES – SEE BELOW

CATALOGUE

PLEASE ASK FOR A COPY OF OUR QL SOFTWARE CATALOGUE (SENT FREE ON REQUEST WITH ORDERS). WE NOW SELL SO MANY ITEMS THAT WE CANNOT PROVIDE MUCH DETAIL IN THE ADVERT, THE CATALOGUE CONTAINS MORE DETAILS OF ALL OUR PROGRAMS

SOFTWARE DISCOUNTS

BUY 2 PROGRAMS TOGETHER, CLAIM 5% OFF

BUY 3 OR MORE PROGRAMS, CLAIM 10% OFF

STANDS

Monitor stand, tilt & swivel, holds up to 14" screen monitors	£15.50
Printer stand, plastic, fits most 80 column printers	£7.50

POSTAGE RATES – SEE BELOW

VISA

Access

SOFTWARE POSTAGE: Software sent post-free to UK addresses. Abroad add £1.00 per program for postage and packing. SUPPLIES POSTAGE: Add £2.50 per order to UK addresses (except labels, see above). Abroad, add 10% of order value (minimum £3.50) for airmail delivery where possible.
PAYMENT:We can accept payment by cash, cheque (in Pounds Sterling drawn on UK branch of bank or building society), Postal Order, International Postal/ Money order, Eurocheque, Payments in UK currency (Pounds Sterling), please, in cases of difficulty, contact us first to arrange payment method.
We can accept payment by credit card (VISA, ACCESS, MASTERCARD, EUROCARD). Please state your card type, number, expiry date, address (goods must be sent to cardholder's address) and remember to sign your order. Goods remain property of DJC until paid for in full. Our telephone number has an answering machine for when we are unable to answer in person – don't be shy!

EUROCARD

Master Card

SINCLAIR QL WORLD

Editor
Helen Armstrong
Production Controller
Jayne Penfold
Designer
Jeff Gurney
Advertising Manager
Jason Newman
Magazine Services
Yvonne Taylor
Advertising Production
Michelle Evans
Group Advertising Manager
Lynda Elliott
Group Editor
John Taylor
Publishing Director
Wendy Palmer
Group Deputy Managing Director
Ray Lewis
Group Managing Director
Peter Welham

Sinclair QL World
Panini House
116-120 Goswell Road
London EC1V 7QD
Telephone 071-490 7161
ISSN 026806X

Unfortunately, we are no longer able to answer enquiries made by telephone. If you have any comments or difficulties, please write the The Editor, Open Channel, Trouble Shooter, or Psion Solutions. We will do our best to deal with your problem in the magazine, though we cannot guarantee individual replies.
Back issues are available from the publisher price £2 U.K., £2.75 Europe. Overseas rates on request.
Published by Maxwell Consumer Magazines, A Division of MCPC Ltd., Sinclair QL World is distributed by IPC Market force, King's Reach Tower, Stamford Street, London SE1 9LS.
Subscription information from: MCM Subscription Dept, Lazahold Ltd., PO Box 10, Roper St, Pallion Ind. Est., Sunderland SR4 4SN. Tel: 091 510 2290
UK: £21.00, Europe: £32.00,
Rest of the World: £38.00.
Please include your subscriber number with any queries.

Typesetting by Ford Graphics, 8-10 Whitsbury Road, Fordingbridge, Hampshire. SP6 1BR. Tel: (0425) 655657
Printed and bound by BPCC, Colchester. Covers printed by Spottiswoode Ballantyne, Colchester.
Sinclair QL World is published on the third Thursday preceding cover date.

CONTENTS

FEBRUARY 1992

4 TROUBLESHOOTER ● New disk drives from Miracle
13 QL SCENE ● Disk interface from Falkenburg
14 OPEN CHANNEL ● DiskOver and TexTidy
16 CQV1 QL DIGITIZER ● A user report
18 SOFTWARE FILE ● QFlick
22 NOTES ON DISK STORAGE ● Some general comparisons
25 THE NEW USER GUIDE ● Part 12 CLOCK to DATA
29 JUST FOR STARTERS ● SuperBasic programming tutorial
32 SOFTWARE FILE ● Five Game Pack
34 DIY TOOLKIT ● Front ends and name tables, part 2
38 SOFTWARE FILE ● Prospero Pascal
42 SYSTEMATIC MACHINE CODE PROGRAMMING ● Part 4
49 SUBSCRIPTION INFORMATION

NEXT MONTH

We have on line a comparison of three well-known wordprocessors from a historic and user's perspective, the second part of Howard Clase's New Uses for Abacus, some more games and utility reviews, and some programming.

T ROUBLE

A PROBLE

Bryan Davies looks at Spellchecker and a box of Miracle hardware.

For the past month or so, my system has been back to its basic configuration, without hard disk, but now it has a new Miracle hard disk unit (the old one went astray somewhere after being serviced). There is a quite considerable improvement in working comfort with the new one; it is *much* quieter than the old, mainly through having only one fan running. It is still mounted on a slab of foam rubber, but there is now little reason to complain; all hard drive units are likely to be somewhat noisy, because the drive needs a fan for cooling. Recent versions of the *Gold Card* have included modified software that provides better handling of hard disk subdirectories; for users who run several programs from subdirectories, this feature makes a big improvement in the usefulness of the hard disk, and it will be useful with the higher-density floppy drives too. The rom version in the GC should be 2.22 or later for proper operation of ED (extra-high density) drives.

All Formats

The All-Formats Computer Fair in London on 3 November was as well patronised as any of the previous fairs, and the QL suppliers looked to be doing reasonably well. DJC, EEC, TF Services, Miracle Systems, Quanta and Digital Precision occupied most of one corner, as usual. It somehow does one good to see new QLs appearing out of the familiar silver boxes after all these years. Digital Precision had the *Perfection Spellchecker* program on show, and the speed of it obviously surprised onlookers. Regardless of file or dictionary size, most spelling checks take seconds rather than minutes to complete. Incorporation of 'new' words into a dictionary is a separate process, but it doesn't take long. The Gold Card now has a red pcb and incorporates some minor changes. From about version 2.21, the rom has included software changes from Tony Tebby, such as the DEV function, which allows eight subdirectories to be treated as individual devices; this makes it possible for programs which accept only the basic five characters (eg flp1_) when defaults are configured to utilise sub-directories with longer names. TF Services was doing a good trade in ZX8301 chips and had its small QL-controlled robot on display. The latter has a simple interface and seems a good device for teaching children the elements of 'robotics'.

In case you haven't been reading the Digital Precision ads recently, note that some of the better programs previously sold by PDQL have resurfaced. These programs have mostly been updated since they were last supplied by PDQL. The pride of the collection has to be either *CPort* or *Cash Trader,* depending upon your orientation. The former is a relatively painless transformer of SuperBasic program code into C-language code. The latter is a home- and small-business accounts-keeping system. Both are working programs; that is, they appeal primarily to users who have work to do, rather than those who are looking for amusement. Neither demand any high level of technical knowledge from the user.

The program which has occupied most of DP's time during late summer is the Spellchecker portion of the Perfection WP program. Spell-checking tends to be an add-on facility, and this one started out that way, but it became a major undertaking in its own right. The problem was not simply writing the necessary program code – although that was a fairly big task – but to decide upon the 'smarts' which had to go into it to enable checking to be *very fast* with a *very large* word list. You are unlikely to come across a larger word list than the one labelled 'Mega' by DP; it presently holds about 360,000 words. In the course of developing the various word lists on offer with Spellchecker, several competitive or comparable offerings were looked at, the largest claiming to have going on half a million words. It was found that some of the lists contained many 'words' which would have baffled dictionary experts, and the effective totals were much lower than the stated ones. The ordinary mortal will still find a large number of the words in the Spellchecker lists unusual, to say the least, but we are assured that they are genuine, having been thoroughly cross-checked against several reference works. Here are a couple of statistics sufficient to make most of us choke – the supplied dictionaries have limits of 32,767 words *for any one letter* of the alphabet and *about* 34 letters for any word. If these limitations cause you problems, you may well have a sizeable problem yourself!

The largest of the Spellchecker word lists occupies over 1 MB in a *compressed form,* the percentage compression from full size being 60-70%. In terms of numeric sizes 1990 has seen the QL world take a large step forward – 12 MB of ram, 16 MHz cpu speed, 3.2 MB floppy disk capacity (all courtesy of Miracle Systems). One thing that has stayed pointedly unchanged is the screen resolution of 512 x 256 pixels. The QL emulator which runs on the Atari ST has gone well past that, with support for 768 x 280 (320?). In the PC world, 1024 x 768 is becoming relatively common. Presumably, a significant increase in resolution means a new display unit, and that means a fair amount of money, but users have shown that they are willing to pay for more memory, a faster processor, hard disk, and higher-capacity floppy drives. Can the QL be made to drive any of the host of higher-resolution displays which are available at reasonable prices for PCs?

Large box

Just as this article was being completed, a large box arrived from Miracle Systems. It contained dual ED (extra-high density) floppy disk drives and a 4-drive adapter. Things rarely go as smoothly as one would like, but *this* installation went without a hitch. The ED drives were plugged-into the upwards-pointing connector on the adapter, the existing dual DD drives were plugged-into the straight-out connector, and a 3-pin plug was put on the mains supply cable. When the system was switched on, all drives worked fine; the ED ones are flp1_ and flp2_, the DD ones flp3_ and flp4_. Note that no changes whatsoever were made to jumper settings in the drives, or to software. The system already had *Gold Card* and hard disk connected; there are a few things to sort out concerning them – eg neither *Professional Publisher* nor *Conqueror* worked from hard disk initially – but these problems should be sorted out in time for further comment on ED drives in next month's issue.

Readers' letters

First, another request – don't send blank cheques to suppliers! Yes, some people do this, and it may be a sign of their faith in the supplier, but it would be better to insert wording to the effect 'valid up to a maximum of £x' (insert the highest figure you think the goods will cost), if you are not sure of the exact amount. Think what *might* happen if a blank cheque got stolen. . . .

Eros Forenzi of the QItaly Club (see INFORMATION below) mildly takes me to task for being less than enthusiastic about Qpac, and suggests I give it another try. This I *have* done, several times, each time deciding to go back to the simpler way I normally do things. The same thing is currently happening to me on the PC, where *Windows* (which has quite a bit in common with Qpac) has convinced me to stick to the text-only (that is, non-graphical) interface. It is not a question of being a computer luddite in the matter – I have long thought that a better, more intuitive user interface was needed and that it would have to be (in some way) 'graphical', but I had not realised until using these two utilities the price the user has to pay in terms of additional actions, memory and time, on both QL and PC, to gain the advantages of the graphical interface.

From what I read, it looks possible that the Macintosh with System 7 and the Next computer provide a much better way of handling interaction between computer and user, but they are in a different price league. Some years ago, I also rejected *Taskmaster* as my permanent 'supervisor' utility, largely because it is a memory hog (as is Qpac), but the advent of the Gold Card means that one can be rather profligate in allocations of ram so, maybe, something like Taskmaster could have a future.

After a few years of battling with a mutitude of new programs, each with their own idiosyncrasies in the methods of actuating functions, one tends to get rather tired of having to 'go back to school' for each new arrival. It may sound like heresy to programmers, but some users actually want programs to be usable without requiring the reading of a heavyweight instruction book, the writing of an extensive boot program, and the burning of midnight oil for several weeks. By and large, Taskmaster is straightforward going, the keying for programs is simple and memorable, and operation is snappy; being a few years old, it could do with some face-lifting, and I must admit to still liking the desktop approach of *Ice,* so a choice of keyboard or mouse control would be desirable to me. What I am, perhaps, saying is that there seems a need for something with the *essence* of Qpac, but with a more 'human' face to it.

We are presently rejoicing in the way the Gold Card has made operation of existing programs so much better, but bear in mind that Miracle have already demonstrated a speed-up board with a 68020 cpu (a step up from the 68000 in the GC), and are almost certainly considering the possibilities of using the 68030 or 68040; boosting the QL with these cpus would put the QL in a state where the software was lagging badly behind the hardware.

Eros sent me two disks containing two issues of the *QItaly Club* magazine and said that 19 issues have been produced since formation of the club in February 1989. That's a good record, and he says they expect to keep it up. The two issues he sent me were very well produced and displayed a professional attitude, for example in the drawing of the opening screens; the disks have a boot file, so the user needs do no more than pop them into flp1_ and boot up. More comment on this form of magazine next month, when I've had time to go through these two issues. A forthcoming issue is planned with just Qpac as the subject, and written in English; send me a copy of that one, too, please.

Four drives

H.R. Banks has had some difficulty getting four floppy drives working together, on his QL(s) with Trump Card and Gold Card. There may be some problem mixing DD and HD drive pairs. This has to be checked but Miracle have, in the meantime, replaced a pair of Banks' DD drives with HD ones. As noted above, my own system now has ED and DD drives working happily together off a GC.

The mention of cut-and-paste operations in a previous issue prompted **Peter Hamill** to mention that he has transferred data from *Archive* to *Perfection* without too much bother, but my impression is that the process is too difficult for relative novices. You can always copy part of a loaded file from one program into temporary storage in a ram disk, then import it into another program; provided the first program can put the file into an Ascii text form when sending it to the ram disk, or the second program has a suitable text-only import function, you have a form of cut-and-paste.

What I was visualising is a much simpler process (for the user, but not for the programmer); a supervisor program – eg TaskMaster, Q_Switch, Qpac – that could take a snapshot (like a screen dump) of all or part of the information on the screen of the first program, put the information in memory and hold it *automatically,* allow you to switch to the second program, put the cursor where the information needs to be inserted, then 'paste' the information there.

There are various drawbacks to the implementations of this idea that I have used, but none of them render the operation useless. One version will 'grab' nothing less than the whole screen – menus, status lines and all – but the surplus can be deleted with the normal editing keys when the information is dumped at its destination. As it is screen information that is being grabbed, it is hardly surprising that functions which were used to produce the screen display in the first place, such as tabs and codes for bold, changing fount etc, get lost in the transfer; typically, this also happens when text-only transfers of files are made via disk files. The best method I've seen succeeds in transferring text or graphics, and (some of) the formatting codes, with only simple key combinations being needed for the cut and paste.

Problems with printer ribbons must be widespread, and **Stanley Hurwitz** was not telling me anything new when he related the problems he has with ribbons for his Panasonic KX-P1124 DMP printer. If you find close on £10 too pricey for a 'branded' ribbon, the only choice seems to be to pay about £3 and get a 'pattern' ribbon of much lower quality. The usual trouble with such ribbons is that they wear badly, and quickly, at the point the ends of the ribbon are joined together (usually at a roughly 45° angle). A hole appears at the join, and characters are unprinted if the action of the print pins coincides with arrival of the hole. Presumably, there could be damage to the print pins in unfavourable circumstances, such as when the hole gets particularly bad. My own attempt to solve this problem, by asking the suppliers for an intermediate-quality ribbon, got me nowhere. They didn't have such a thing, and supplied me a branded ribbon, which hasn't exhibited this problem, certainly, but prints rather fainter than the 'cheapies' do. It wasn't worth the money. In earlier years, the cheap ribbons for my Kaga-Taxan (Canon) DMP were quite good and not too prone to producing holes; my practice was to re-ink these and branded ribbons, but there was a tendency for print density to vary quite considerably. Has anyone found a source of decent-quality ribbons for the KX-P1124, at between £3 and £6?

Good news from **A. Ingrey,** who got both a replacement keyboard and interface from EEC Ltd. *and* a refund from his credit card company! Hopefully, the money side has been sorted out by now. He has an unusual problem with disks, that has appeared since he installed a Gold Card, although that may be pure coincidence. The disks in question hold letters which have been written over almost a year, and they now appear to be unusable as a result of some form of directory corruption. There are over 100 files on the disks, and Ingrey wonders if that is too many. The basic answer to that is 'no', but the size of the files does matter.

Experiment

As an experiment, I've copied close to 500 files onto one 720 KB floppy; the experiment was terminated by the 'drive full' message. There was no sign of corruption, but the files were all founts, occupying less than 1 KB each, and couldn't easily be checked for correctness. The possible technical explanation of what happened on Ingrey's disks is something that has been put to me in the past, but some of the detail has got lost in my memory banks. Essentially, the story was that it is possible to have a certain number of files sitting on a disk, occupying almost all the space but giving no trouble, until one of the files is modified and increased in size. The directory space for that file will already have been allocated, at a time when directory allocation space and file space did not exceed the available space on the disk; then, the file is increased in size so that there is insufficient room for it. Presumably, what *ought* to happen is that something – Qdos – recognises the potential danger and refuses to start the save process which will end up with the drive full and the file incomplete on disk. The user should be warned to save to another disk, before the existing file or – much worse – the directory, get corrupted.

As for a way of avoiding such a problem in future, the only obvious point is that the process of regularly saving files with the same names, but increasing sizes, is what leads to the trouble. You don't get it by simply copying a lot of files at the same time. Maybe he should use one disk for saving current documents (not taking up most of the disk), and transfer these to archive disks at intervals, when the files are in their 'final' state.

Data file

Apropos of saving SuperBasic program files from Perfection, **Martin Wheatley** suggested creating a special printer_data file for the purpose, rather than switching the line wrap function off. The file would have no entries other than LF for the EOL (end of line) code. DP do supply such a file with Perfection, under the name Null_data, and it is no doubt successful for saving Ascii text files for export to other programs, but it doesn't appear to solve the problem I spoke of a few issues back. If you edit an SB program file in Perfection, and use the ENTER key in doing so, that key puts a character – shown as a right-angled arrow when the CTR/-H command is used to display codes – and that appears as a 'splodge' when the file is subsequently LOAD-ed from the SB command line; that program line also has MIStake inserted after the line number. This is even though the file is saved by using the Print command and specifying a drive rather than the printer port as the destination. I'll stick with switching line wrap off and using the Export command to save the file. Martin gave the address (see INFORMATION) of his local Quanta sub group meeting place and recommended it for informal, social gatherings of QL users.

INFORMATION

QItaly Club (for Italian QL users):
Eros Forenzi
Via Valeriana 44
23010 Berbenna (SO)
Italy
Tel: +39 (0342)-492323

Quanta (London group):
Meetings are held from 1900 hours on the second Wednesday of each month at the Prince George of Cumberland, 195 Albany Street, London NW1.

PERFECTION
PERFECTION PLUS

Perfection is the finest word processor available for any computer. We have received dozens of letters from happy users saying just this... and all of these letters were unsolicited. "Superb" was used most often.

Perfection manages to achieve all the sophistication of the most complex PC word processors while still using a user interface as friendly as Quill's. Perfection has a dual system of user control: menus while you are familiarising yourself with the program, and direct commands for the time when you feel ready for more adventurous things. The two systems can be used interchangeably and even simultaneously. Even more exciting – both systems are iterative. In case you don't understand what this means, let us give you an example: suppose you wished to move a block of text using the menus. You would choose Block Move (yes, it is right in the first menu) and the screen would then tell you to move your cursor to the start of the block. On most word processors you would have to navigate manually to this position: indeed, on many of them (Quill included) only a subset of the normal navigation commands would be available. On Perfection, not only can you use all the manual navigation commands (viz all 28 permutations of CTRL, ALT, SHIFT and the arrow keys!) but in addition you can use direct commands like GoTo Line or Page or any of eight markers. Even more amazingly, you can use Search (either as a direct command or from the menus) *even though you are already 'within' a menu option.*

Perfection has about 200 commands, but the layout of menus and the choice of keys for the direct commands makes it very easy to master. Though a 100+ page manual is provided (with all the important bits right at the front), you should only need to consult it for specialised operations like macros.

Even if speed is not particularly important to you, we assure you that Perfection's lightning performance will enable you to use the word processor in sensible ways that you would not have dreamed possible before. For example, scrolling 100 pages or so is accomplished so quickly using the normal navigation commands that you do not need to bother using a menu option to do the move. Spellchecking, assuming you have Perfection Plus, is accomplished virtually instantly: to spellcheck this whole ad (all the pages) would take under 1.5 seconds... Searching (you can switch case sensitivity, as well as equivalences between tabs, soft spaces and hard spaces) is at the rate of about 100 A4 pages per second.

Moving from one word processor to another is usually very traumatic. With Perfection, this will not be the case. Not only can Perfection read in Quill _doc and _exp files directly (you do not even need to tell it they are Quill files!) but it can make direct and immediate use of your existing Quill printer driver. File re-export is also possible.

Perfection is truly WYSIWYG: this means that bold appears bold on screen, italics appear as italics, underlined as underlined, and so on. Of course, your printer may have functions we do not know about (upside down?). To deal with these, Perfection provides a number of on-screen shaded strips: these can be attached to any printer function you wish, and will not upset justification as a translate would. Of course, translates are provided as well!

A variety of statistics on the document being processed are available: some of them are on view all the time, the rest can be toggled to instantly. Not only is there a word count, but also page, line, character and special character (like Superscript Off) counts. There are also a dozen status indicators, letting you know whether you are in Insert or Overwrite mode, whether a block is defined, whether interactive spellchecking is enabled etc. Current line (from top as well as within page) and column positions and character codes are also available.

A terrific feature of Perfection is the dual screen mode. You can view one part of the document while editing another. The sizes of the two windows are themselves adjustable, both in real-time or via the configurator. We should devote more space to the configurator: however, it must suffice to say that everything that could be dynamically set within Perfection may also be preset with the configurator. The configurator can, for example, allow you to select any of 256 colours for any of a dozen parameters (like paper colour, border colour, status window ink and paper colour etc).

Perfection is fully multitasking without need for any external accessory: however, if you already use QPAC or Taskmaster or similar and are happy, you may go on doing so.

There is absolutely no way that we can prepare you for the quality 'feel' of Perfection. We have a great deal of experience using PC word processors costing many hundreds of pounds: with absolutely no exception, Perfection is far easier to use and master.

So if you thought Perfection was unattainable, you have a very pleasant surprise coming to you!

LIGHTNING SPECIAL EDITION
LIGHTNING

These programs accelerate QL operation by up to 10x (2x –4x is typical) without having any adverse effect whatsoever on compatibility or anything else. Lightning SE is typically 40% faster than the standard version. This acceleration is totally independent of, and in addition to, any speed-up obtained by hardware means. So if you have Gold Card, your need for Lightning SE is just the same as if you had only an unexpanded QL – Lightning SE will accelerate both by the same ratio.

The Lightning programs achieve their acceleration by automatically paging out sections of the QL's operating system and replacing these with optimal, concise code written by us.

Lightning installation is a completely automatic and one-off: no knowledge of computing or programming is required. Once installed, Lightning can be completely forgotten about – you will soon get used to the superb speed! Knob twiddlers are catered for too.

Lightning technology is not built in to any of our other programs. Perfection users (as well as users of all other QL software) should therefore use Lightning all the time.

In summary: if you do not have Lightning, you are wrong. Buy this one **FIRST OF ALL!**

PROFESSIONAL PUBLISHER

The Professional in Professional Publisher refers to the quality of output from that program, and is not meant to suggest any complexity of operation. Few programs are as easy to use as this one: > 99% of users will be able to do without using a manual!

Professional Publisher is by far the best DTP program for the QL. It is fully compatible with Perfection, Editor, Quill, Eye-Q & the ASCII editors. It allows you to *both* create and import *both* text and graphics. Text can be 'poured' into boxes of any shape , size and number, automatically maintaining justification and hyphenation settings. So flowing text around graphics is a doddle.

Professional Publisher is supplied with a generous selection of fonts of various sizes, as well as clip Justification is by pixel, not by character. This gives a much smoother effect.

It is pointless for us to try to list all of Professional Publisher's features – we would end up filling half the magazine! We will concentrate on just a few 'points': Professional Publisher is extremely precise, performing all its computations accurate to a small fraction of a millimetre. All its features can be preset by you using its configurator, ruling out the need for repetitive key strokes.

The program is extraordinarily versatile while remaining intuitive in its user interface. Buy it!

DIGITAL PRECISION LTD

PROFESSIONAL PUBLISHER
TOOLBOXES

Toolbox I is an excellent collection of high definition fonts, clip art and utility programs for Professional Publisher. While the fonts supplied with Professional Publisher are excellent, many users will feel the need for a wider range of typefaces and styles.

Toolbox II starts where Toolbox I leaves off, providing an even better – and different – font collection.

The two Toolboxes complement each other and are available together at a special price.

FONT ENLARGER
GRAFIX

Font Enlarger does exactly what you would expect it to from its name. While Professional Publisher is also capable of enlarging fonts, it does them 'on the fly' and consequently is not able to remove the jaggedness caused by magnification. Font Enlarger is much cleverer, and enhances detail without any step effect.

While the built-in printer driver for Professional Publisher is excellent with 9-pin printers, it is not optimal with 24-pin or laser printers. Grafix is.

DIGITAL PRECISION LTD

EYE-Q
ULTRAPRINT

Eye-Q is the finest graphics program for the QL. While there may be other graphics programs with a few more features, no other program comes anywhere close to Eye-Q in sheer enjoyability. Eye-Q develops a pleasurable tactile relationship with you, and makes you feel like an artist (even if you aren't). Eye-Q graphics can be read in by Professional Publisher, and the latter's pages can be exported to Eye-Q (using Toolbox I). Everything in Eye-Q is menu-driven and there is context-sensitive help.

While Eye-Q has its own printer driver, Ultraprint allows you 22 distinct styles/sizes of printer output. The reasoning is that the scale of gradation suitable for pictures is probably unsuitable for text or line drawings.

PC CONQUEROR
SOLUTION

PC Conqueror makes your QL into a PC-compatible machine,automatically. It does this by software means only, so there are no screws to undo or wires to fiddle with. Your QL stays a QL too.

Why, might you ask, should you wish to make your QL into a PC-compatible? The reason is simple: you may wish to run the same programs at home as you do at work. Alternatively, you may wish to tap into the vast storehouse of PC software of every type and description you could imagine.

Using PC Conqueror could not be easier. Just boot up your machine with the PC Conqueror disk in floppy 1 and within 10 seconds your QL will be transformed into a PC that is just waiting to be switched on. From this point on you will do exactly the same as you would if you were running a 'real' PC – this means putting a DOS disk (any version) into one of your drives and pressing a key. If you do not already have legal access to a copy of DOS, we can provide you with one at reasonable cost (see our price list).

PC Conqueror runs as fast as it is possible for a PC emulator to run: we have used all our skills to make it work quickly. Of course, you can make the emulation must faster by using Gold Card and Lightning SE. With this combination, you should get speed noticeably better than that of a PC XT...

PC Conqueror allows you to fine-tune the operating environment of the PC in order to improve performance. If you get a hard disk or other high capacity floppy system, you can utilise part or all of it as a PC hard disk.

PC Conqueror occupies under 80K and leaves 667K free for DOS when run on a Trump Card. This is more than you will get on a 'real' PC.

Solution does what Conqueror does but is about half as fast and is not quite as compatible.

SPELLCHECKER
MEGA DICTIONARY

Spellchecker is what makes Perfection into Perfection Plus. We have made it available as a separate item for two reasons: (a) to allow Perfection owners to add it later (b) to allow users of other word processors to benefit from the very best in spellchecking technology.

Spellchecker is supplied complete with three dictionaries of differing sizes as well as a system for building, reviewing and maintaining user dictionaries.

Spellchecker's ultimate accessory is the Mega Dictionary, which gives the user a vocabulary of over 350,000 words!

3D PRECISION CAD SYSTEM

This program allows you to manipulate shapes and figures in 2D and 3D at a speed that will leave you breathless. Irrespective of whether your interest is in CAD, in animation or in just having fun, this program should not be missed. You can output to plotters directly from it, or alternatively create graphics screens to be manipulated and output by Eye-Q, Ultraprint or Professional Publisher.

SUPER SPRITE GENERATOR

SSG moves things about the screen very fast and very smoothly, without flicker. Sprites can have up to 16 frames.

DIGITAL PRECISION

DIGITAL PRECISION LTD

DIGIT**A** LP REC ISION

A **I**GITD L**P** REC ISION

A P**G**ITD LI **R**EC ISION

A PRI**TD** LI GEC ISION

A PRID**T** LI G**E**C ISION

A PRIDE **LI** GTC ISION

A PRIDE I**L** GTC ISIO**N**

A PRIDE IN **G**TC **I**SIOL

A PRIDE IN IT**C** G**S**IOL

A PRIDE IN ITS **G**CIO**L**

A PRIDE IN ITS L**C**I**O**G

A PRIDE IN ITS LO**I**C**G**

A PRIDE IN ITS LOG**CI**

A **PRIDE** IN ITS **LOGIC**

A LOGIC IN ITS PRIDE

MEDIA MANAGER SPECIAL EDITION
MEDIA MANAGER

Media Manager Special Edition (MMSE) is a program to be used both when things have gone wrong as well as when things are perfectly OK. It allows for automatic, semi-automatic and manual correction of a huge variety of disk and tape problems. It allows you to explore disks and tapes to your heart's content, producing all sorts of different diagnostic reports. MMSE is very simple to operate, being menu-driven and assuming no degree of computer knowledge whatsoever.

MMSE also allows you to tidy, catalogue, sort and order your disks and cartridges.

The standard Media Manager is both less powerful and less user-friendly, but manages to work on an unexpanded QL.

Both programs allow for data transfer between PC and QL. With MMSE, this transfer is at file and directory level, is bi-directional and is completely automatic.

SPECIAL DESKTOP PUBLISHER
DESKTOP PUBLISHER

These programs are very primitive compared to Professional Publisher. However, if you have not experienced that program as yet, you will find both of these very competent. Both are capable of producing excellent results. The cheaper one has fewer features but is able to run on smaller systems.

EDITOR SPECIAL EDITION
THE EDITOR

With the sole exception of Perfection, this is the best word handling system on the QL. Editor's features include an unrivalled degree of programmability and the ability to cope with the entire 256 character ASCII set. The Special Edition has enhanced document-type facilities, including column blocks and on-screen page break displays. Neither program is suitable for computing novices. Until Perfection, Editor Special Edition would have been our 'Desert Island Program'.

Editor SE can do a few things that Perfection can't, so the ideal combination is to have both (they are compatible at file level and can multitask). If you order Editor SE at the same time as Perfection, you can have Editor SE at half price.

PROFESSIONAL ASTROLOGER
PROFESSIONAL ASTRONOMER

The Astrologer program teaches you Astrology from scratch and enables you to automatically produce text narrative on personality delineation, year-to-year and minute-to-minute life predictions, compatibility interpretations and so on. Whether or not you believe in astrology – indeed, especially if you do not – this program is one that you cannot afford to have. You can tailor the readouts (both in terms of quantity and what is said) to your own particular requirements. The amount of fun you can have with this program is endless. Do not blame us if you start believing in astrology, though!

Astronomer is an extremely fast and accurate solar system calculator, with planetarium views, planet faces, eclipses, cinerama display etc..

TURBO BASIC COMPILER
SUPERCHARGE BASIC COMPILER

Turbo is the finest BASIC compiler for the QL and arguably the finest BASIC compiler for any computer!

Turbo automatically converts working BASIC programs into optimised machine code, with no need for human intervention. The benefits of this conversion are vastly enhanced running speed (as well as much faster loading, encryption and automatic bug fixing for a variety of QL interpreter oddities). Typical speed-up is 20x – 100x.

Turbo is provided with a 200 command toolkit, adding many useful commands to BASIC. Most of these commands will be of immediate use to the programmer, whether he is a novice or an expert. There are commands to load strings and floats into RAM, and to extract them automatically; to search memory and to move its contents; to control jobs and change their priorities...

Supercharge has a few of Turbo's facilities and half its speed.

TOOLKIT III

Toolkit III starts where Toolkit II stopped, adding about 60 new commands and enhancing many existing dual functions. Toolkit III is available either on disk or on ROM, and works whether or not you have Toolkit II.

Toolkit III commands can, with only a couple of exceptions, be compiled using Turbo.

QFLICK CARD INDEX

All QL owners have a copy of Archive, supplied free with the QL. While Archive is competent, it is very hard to get to grips with and is not particularly fast. QFlick presents a very convenient alternative – a snappy , simple-to-use, pointer-controlled card file database. You can move data between QFlick and Archive in either direction.

QFlick is not itself programmable but we document its data structure and give guidance on how to program it using Turbo.

ARCHDEV + RTM
DATABASE ANALYSER
ARCHIVE TUTORIAL
NAMES + ADDRESSES
MAILMERGE
DAT–APPOINT
SEDIT
SCREENPRINT
RECOVER

This suite of utilities will greatly enhance your use of the Archive database system.

Archdev + RTM is a straight replacement for Archive: it gives enhanced speed, greater workspace and a much cleaner boot-up. All your existing applications will work.

Database Analyser provides very fast and comprehensive statistics about your Archive databases.

Archive Tutorial proceeds systematically through the whole philosophy and grammar of Archive, providing you with expert and patient guidance.

Names + addresses, Mailmerge and Dat-Appoint are ready-to-run, off-the-shelf Archive applications, providing an address database, mailmerging and appointment diary respectively. You now have no excuse not to use Archive.

SEdit allows you to create and edit screen format files in Archive. Screenprint allows you to print them out.

Recover allows you to get back lost Archive databases, created when you switched off the computer without properly exiting from Archive.

XREF
SUPERBASIC MONITOR
BETTERBASIC EXPERT SYSTEM

XRef analyses the structure of a BASIC program, providing detailed reports on things like variable usage, what calls what, dynamic call hierarchy of procedures and functions, and so on.

SuperBasic monitor actually monitors and reports on the performance of BASIC programs as they run under the interpreter.

BetterBasic analyses and automatically corrects structural flaws in your programs and allows you to customise things like indentation, number of statements per line, filtering out of noise words, etc.

The three programs together provide a matchless diagnostic and auto-correcting facility for BASIC programs.

TRANSFER UTILITY

This program copies files at high speed between devices, performing translates as it goes along. Ideal for all sorts of applications, including transfers from microdrive to disk.

QMATHS SYSTEM

This is an incredible mathematical compendium for the QL. Pride of place goes to the symbolic problem solver: this can solve equations, simplify expressions, factorise, expand, etc, all symbolically. If you could sneak this one into a maths examination, you would have a formidable ally. QMaths knows about all the algebraic operators, powers, roots, brackets, trigonometry, matrices, determinants, vectors, factorials, permutations, combinations, binomials, exponentials, logarithms, hyperbolics, inverse functions, infinite series including Taylor & Maclaurin expansions, complex numbers, conversions, Fourier series, and lots of calculus: both differential and integral, including integration by parts and definite integrals. QMaths optionally displays its workings and comes with a superb interactive tutorial.

The package also contains an interpretive, fractal, image-generating language with loads of beautiful fractal programs supplied for you to use and edit – no programming skill is required.

There is also a multiple precision floating point maths package, giving calculations at precisions up to over 600 decimal digits of accuracy.

There is even more to this system, but we think we have told you enough.

QMON MACHINE CODE MONITOR

The latest version of Tony Tebby's superb monitor: an absolute must for those who really want to know what is going on in the QL. No other machine code monitor even comes close.

Do not confuse this program with SuperBasic monitor, which monitors SuperBasic, not machine code.

COMPARE

This program compares files – data or program – at colossal speed. Where a mismatch is detected, the relevant areas are highlighted and you can shuffle, displace and align very easily.

CASH TRADER WITH ANALYSER
PAYROLL

Cash trader with Analyser is an accounts system designed by businessmen and not by wretched accountants! Consequently, it has excellent reporting and management facilities, and is very flexible. It is aimed primarily at the layman, probably a sole trader running a small or medium sized business. All the features you would expect – including audit trail – are present.

Payroll is a reasonably flexible system designed to automate the payroll function in small businesses.

Both programs are configurable, with editable defaults letting you adapt the programs from year to year.

HARDBACK WITH FINDER

This is the ultimate hard disk backup and management utility, with all the sophisticated features you could want. User dialogue is via overlapping pop-up windows – the whole program just feels right. It is possible to scan the disk at great speed, too.

DISKTOOL WITH QUICKDISK

This permits you to add password protection to disks, to optionally increase disk storage capacity on DSDD drives by 36K and to increase speed of access by as much as 30%. All this is done while maintaining full compatibility. Automatic file management is also provided.

DIGITAL C SPECIAL EDITION
DIGITAL C

These are extremely fast and efficient C compilers, complying with and surpassing the Small C definition. The Special Edition goes much further, including support for structures, pointers, long pointers, >64K code size, direct access to QDOS traps, etc. The Special Edition C generates code that runs about twice as fast as the other.

SPECIAL DEALS

5% off total if you buy 2 programs/upgrades;
10% off 3; 15% off 4; 20% off 5; 25% off 6+
Upgrades cost difference in price + £10
Non-UK Europe add 5%, rest of world 10%

For full terms and conditions, please refer to any of our QL World ads from Jan–Nov 1990, or write in including a SAE

CPORT BASIC TO C CONVERTER

This program translates SuperBasic programs directly into C source code, automatically. This C source code may then be edited or compiled. If you want to move programs to C for migration to other hardware, or want to accelerate your programs, or just want to learn C the easy way (chuck BASIC in one end and examine the C that spews out of the other), CPort is the system for you.

CPort is friendly and tolerant of poorly written BASIC. There is even a method of dealing with unusual BASIC keywords. The generated C, which can be switched between the ANSI and Lattice industry standards, is very readable and is often optimal. CPort's user interface is extremely friendly. CPort is available with or without the C68 compiler.

SUPERFORTH COMPILER WITH REVERSI

Forth is the most logical computer language. This compiler produces multitasking code. The manual teaches you Forth-83 from scratch.

IDIS SPECIAL EDITION IDIS

These intelligent disassemblers make the otherwise terrifyingly complex task of understanding other people's machine code programs absurdly easy. The SE version, which has a higher hardware requirement, sorts out some routines, replaces addresses with names, untangles data from code and much more.

QKICK FRONT END SYSTEM

This is a simple, easy-to-master, pull-down menu controlled multitasking front end. QKick runs in the background and can be called up at any time. It provides you with notepads, sophisticated file/sector/RAM handling, backing up facilities, a clock, diary, calculator, mini-database and so on.

ADVENTURE CREATION TOOL SPECIAL EDITION

ACT is a must for every programmer. The name of the program is misleading, insofar as it has capabilities far beyond the 'mere' creation of adventures. ACT has utilities providing animated graphics, data compression, language design, parsing, maps, object-oriented control etc. If all you want to do is generate adventures, though, you do not need to be a programmer to use it. This is a purchase you will never regret.

PEDIT

A fast, modern and capable printer driver for the programs bundled with the QL.

MICROBRIDGE

Superb contract bridge bidder (ACOL etc) and player, using millions of random but reconstructable hands. Microbridge also includes a state of the art interactive bidding tutor and a clear instruction manual. There is nothing like this anywhere else!

SUPER ASTROLOGER

A very cut-down version of Professional Astrologer – still great fun, though!

SUCCESS CP/M EMULATOR

Allows your QL to run CP/M programs at great speed.

£ 169.95	eT	PC CONQUEROR WITH DR-DOS v5.0
129.95	dT	PERFECTION PLUS WITH SPELL CHECKER
119.95	eT	THE SOLUTION WITH DR-DOS v5.0
99.95	cT	CPORT WITH C68K COMPILER
99.95	cT	CASH TRADER v3.0 + ANALYSER
89.95	dT	PERFECTION
89.95	eT	PC CONQUEROR
89.95	cT	PROFESSIONAL PUBLISHER
89.95	cT	CPORT BASIC TO C TRANSLATION SYSTEM
79.95	aT	TURBO BASIC COMPILER + TOOLKIT
69.95	aT	PROFESSIONAL ASTROLOGER WITH ASTRONOMER
59.95	cT	QMATHS MATHEMATICAL SYSTEM
59.95	aT	PROFESSIONAL ASTROLOGER
49.95	+aT	LIGHTNING SPECIAL EDITION
49.95	eT	PROFESSIONAL PUBLISHER TOOLBOXES (PAIR)
49.95	eT	ACT SPECIAL EDITION
49.95	dT	THE EDITOR SPECIAL EDITION
49.95	dT	SPELLCHECKER
49.95	dT	MEDIA MANAGER SPECIAL EDITION
49.95	dT	3-D PRECISION CAD SYSTEM
49.95	cT	PAYROLL
49.95	cT	HARDBACK + FINDER
49.95	bT	SUCCESS
49.95	aT	DIGITAL C SPECIAL EDITION
44.95	+a	TOOLKIT III WITH ROM
39.95	eT	THE SOLUTION
39.95	dT	IDIS SPECIAL EDITION
39.95	ct	SPECIAL DESKTOP PUBLISHER
39.95	at	QMON MACHINE CODE MONITOR v2.05
39.95	aT	SUPERFORTH COMPILER WITH REVERSI
39.95	aT	EYE-Q
34.95	d	MICROBRIDGE
29.95	gT	MEGA DICTIONARY
29.95	fT	PROFESSIONAL ASTRONOMER
29.95	eT	PROFESSIONAL PUBLISHER TOOLBOX
29.95	eT	PROFESSIONAL PUBLISHER TOOLBOX PART TWO
29.95	a	TOOLKIT III
29.95	a	SUPER SPRITE GENERATOR
29.95	a	SUPERCHARGE SPECIAL EDITION
29.95	a	QFLICK CARD INDEX SYSTEM
29.95	a	PERFECT POINTER TOOLS
29.95	a	GAMES COMPENDIUM (ALL FIVE GAMES)
29.95	at	THE EDITOR
29.95	at	DIGITAL C COMPILER
29.95	aT	XREF v2.0
29.95	aT	SEDIT
29.95	aT	ARCHDEV + RUN-TIME MODULE
24.95	f	SUPER ASTROLOGER
24.95	d	DESKTOP PUBLISHER
24.95	a	QKICK MULTITASKING SYSTEM
24.95	a	MEDIA MANAGER
24.95	at	IDIS INTELLIGENT DISASSEMBLER
24.95	aT	SUPERBASIC MONITOR
24.95	aT	LIGHTNING
24.95	aT	BETTER BASIC EXPERT SYSTEM
19.95	eT	FONT ENLARGER
19.95	dT	NAMES AND ADDRESSES
19.95	dT	DAT-APPOINT
19.95	b	DISKTOOL WITH QUICKDISK
19.95	aT	ULTRAPRINT
19.95	aT	SCREENPRINT
19.95	aT	RECOVER
19.95	aT	PEDIT
19.95	aT	MAILMERGE
19.95	aT	DATABASE ANALYSER
19.95	aT	COMPARE
19.95	aT	ARCHIVE TUTORIAL
9.95	f	SUPER BACKGAMMON GAME
9.95	f	REVERSI GAME
9.95	f	DROIDZONE GAME
9.95	f	BLOCKLANDS GAME
9.95	f	ARCADIA GAME
9.95	eT	GRAFIX
9.95	b	TRANSFER UTILITY
* KEY:	+	As well as mdv disk, you get a ROM
	a	Available either on mdv or disk
	b	Available only on disk
	c	Minimum 512K RAM:only available on disk
	d	Minimum 256K RAM:either mdv or disk
	e	Minimum 256K RAM:only available on disk
	f	Available only on mdv
	g	Minimum 1.5M RAM:only available on disk
	T	Compatible with THORs including XVI
	t	Compatible with all THORs except XVI

NEW!

✂

Send to: DIGITAL PRECISION LTD, 222 THE AVENUE, CHINGFORD, LONDON E4 9SE or phone 081-527 5493
Please rush me: ....
Name: .... Address: ....
.... Postcode: ....
Encl: CHEQUE/VISA/ACCESS/MASTERCARD/PO/MO/CASH for £ ....
Card No: .... Expiry: .... Signed: ....

DIGITAL PRECISION LTD

Delete as appropriate: (QL/THOR/STQL)(MDV/3.5"/5.25")(Disk=360/720/1440/2880/6400 sectors; nb 1 sector=0.5K)(If 720 sectors: SSDD/DSSD)

MICRO ADS
MICRO ADS
MICRO ADS
MICRO ADS
MICRO ADS

JS QL, Trumpcard, 2 x 3½ DD, Microvitec 102 Keyboard, manuals, journals, Psion suite, Pro Publisher, SE Lightning Taskmaster, Pagedesigner, EyeQ, TurboQuill, SE Conqueror, MsDos 4.0, £400. Tel: 0225 332812.

MICRO ADS
MICRO ADS
MICRO ADS
MICRO ADS
MICRO ADS

QL ROMS AND SPARES

ROMS, MGE (Spanish, MGF (French), MGG (German), JS (English), £20 each. JM (English), £10

MDVS. Complete QL MDV Subunit £20, head and chassis £10, Microswitch £2, Motor £6. Rubber & Plastic Roller £2. MDV2 Stand-off spacer + srew £2. ULA £5. Tr 4,5,6,7 set £2. MDV pcb inc, ULA £11. Prices include UK postage. Send SAE for price list of all QL spares.

**Joe Atkinson, 36 Ranelagh Road, Ealing,
London W5 5RJ (Mail Order Only).**

MICRO ADS
MICRO ADS
MICRO ADS
MICRO ADS
MICRO ADS

FOR SALE, Conqueror, Taskmaster, Spellbound, Filebound, Chess, QDraw, Serial 8056, 2 Uncased TEAC 3.5's, buyer pays postage, offers. Tel: 0303 872238 evenings.

MICRO ADS
MICRO ADS
MICRO ADS
MICRO ADS
MICRO ADS

CLASSIFIED

Please send coupon to: Advertising Department Sinclair/QL World, Panini House, 116/120 Goswell Road, London EC1. Pleae place this Classified Advertisement in the next issue of Sinclair/QL World.

NAME ....

ADDRESS ....

....

DAYTIME TELEPHONE NO ....

I enclose a cheque/P.O. to the value of £17.65 (inc. VAT) for a lineage advertisement of no more than 30 words. Please make cheques payable to MCPC Ltd

Z 88

EaZy Money	£71.00	Centronics Interface	£22.25
Z-Base	£71.00	128K Eprom Pak	£28.00
Z-Tape	£51.50	128K RAM Pak	£38.75
Z-Term	£51.50	Z88 Computer	£223.00
ThinkZ	£51.50	512K Ram Pack	£103.50
Superchip 4	£154.00	256K Eprom Pack	£68.25
Golf	£25.50	1Mgb Ram Pack	£220.00
Spellmaster	£60.50	Squeeze 88	£47.00
Topper	£10.50	Superchip 3	£133.50
		Z88 QL Link	£27.50

PSION SERIES 3

128k Computer	£200.00	256k Computer	£250.00
128k Flash card	£40.00	256k Flash card	£70.00
512k Flash card	£120.00	1MgB Flash card	£200.00
2MgB Flash card	£300.00	128k Ram card	£80.00
512k Ram card	£200.00	1MgB Ram card	£300.00
Parallel interface	£30.00	Mains adaptor	£15.00
PC link	£70.00	Mac link	£70.00

TK COMPUTERWARE

the QL stockist

Tel: 0303-81-2801

PSION ORGANISER 2

Model LZ	£100.00
Model LZ 64	£130.00
Comms Link	£60.00
Comms Link – Mac	£73.00
Travel Pak	£40.00
Harvester Superchip	£100.00
Letter Organiser	£40.00
Informed Organiser	£40.00
Diary Link	£13.00
128k Datapak	£100.00
64k Datapak	£60.00
32k Rampak	£60.00

SPARES

Keyboard Membrane	£8.50
JS ROM set	£31.00
ZX 8301	£15.50
ULA ZX8302	£10.50
MAB 8049	£7.50
QL Power Supply	£31.00
MC 68008	£26.50
M/drive assembly-(complete)	28.00
Microdrive Assembly (exchange)	£21.50
Voltage regulator (high output)	£5.00
QPower Regulator	£25.50
QL Test kit	£14.50
Repairs	phone

SUNDRIES

VDU cleaning kit	£7.50
Disk Cleaning Kit 3.5"	£15.50
10 Microdrive cartridges	£30.00
100 Microdrive labels	£4.00
Discs 3.5" 10 DSDD (branded)	£26.00
Power Clean Mains Filter	£24.50
Joystick + adaptor	£18.50
Psion 2.35 Set	£23.50
Printer Ribbons	Phone
Keyboard-90 Interface	£77.00
Tandata QCOM set	£90.00
QL Scanner	£143.50
QUIMI I/F with mouse	£101.50

SOFTWARE

BUSINESS

Project Planner	£30.00
Sales Invoice	£20.50
Small Traders Pack	£30.00
Stock Accounting System	£41.00

TEXT

Master Spy	£31.00
Page Designer 2	£36.00
Perfection	£80.00
Perfection +	£125.00
Professional Publisher	£90.00
QTYP 2	£31.00
SPY	£15.50
Turbo Quill Plus	£14.00
The Editor	£30.00
The Editor Special Edition	£50.00

FINANCE

Home Finance [Buzz]	£22.50
Investment Monitor	20.50

CAD

Eye Q	£40.00
GraphicQL Plus	£25.50
PCB 1	£102.50
PCB 2	£51.50
TechniQL	£51.50
Technikit	£25.50
The Painter	£51.50
3-D Designer	£39.00
3D Precision	£50.00

PROGRAMMING

Assembly Workbench	£25.50
Basic ally	£20.50
Better Basic	£25.00
Digital C	£30.00
Digital C Special Edition	£50.00
Forth	£31.00

IDIS	£25.00
IDIS Special Edition	£40.00
Macro Assembler - GST	£39.00
Macro Assembler - Talent +	£20.50
Pro Fortran - 77	£101.50
Pro Pascal	£101.50
QMON 2	£20.50
Lightning Special Edition	£60.00
Locksmith	£15.50
Mailfile	£15.50
Media Manager	£25.00
Media Manager Special Edn	£50.00
QFlash Ramdisc + t'kit	£19.50
QFlash Ram Eprom	£25.50
QFlash Ram Utilites	£10.50
QPAC 1	£23.50
QPAC 2	£51.50
Q Top	£31.00
RPM	£15.50
Sidewinder Deluxe	£20.50
Super Toolkit 2 (Eprom)	£30.00
Super Toolkit 2 – Reconfigurable	£30.00
Success	£40.00
Terminal	£31.00
Touch Typist	£12.50
Ultraprint	£20.00
XRef 2.00	£20.50
4 Matter	£15.50

LEISURE

Bridge Player 2	£15.50
Cosmos+	£15.50
Flight Simulator	£19.50
Matchpoint	£15.50
Micro Bridge	£40.00
Prof Astrologer + Astronomer	£80.00

Prof. Astronomer	£40.00
Snooker	£13.50
Strip Poker	£15.50
Super Astrologer	£25.00
Super Backgammon 3	£13.50
Type 22	£18.50

GAMES

Aquanaught 471	£19.50
Arcanoid	£15.50
BJ in 3D land	£12.50
Brain Smasher	£18.50
Crazy Painter	£13.50
Crown of Sorkinen	£18.00
Deathstrike	£15.50
Firebirds	£15.50
Hopper	£13.50
Hoverzone	£15.50
Jungle Eddy	£15.50
Karate	£15.50
King	£13.50
Land of Havoc	£19.50
Lost Pharoah	£15.50
Mortville Manor	£15.50
Stone Raider	£13.50
Super Arcadia	£10.50
Vroom	£13.00

UTILITIES

Archivist	£31.00
Archivist MP	£41.00
ArchRTM+ArchDEV	£20.50
Cartridge Doctor	£18.50
Conqueror	£100.00
Conqueror MS-DOS 4.01	£125.00
Conqueror with DR-DOS 5.00	£180.00
Discover	£31.00
Lightening	£30.00
Super Forth + Reversi	£40.00
Super Sprite Generator	£31.00
Supercharge Special Edition	£25.00
Turbo	£80.00

• Please telephone for details of other products not listed here •

BOOKS

Assembly Language Prog	£9.00	Using Your Z88	£10.00
Z88 Dabhand Guide	£16.00	Z88 Computing	£10.00
Z88 Ref Manual	£16.00	Z88 Software DLV Guide	£40.00
Z88 Magic	£16.00	Using Psion Organiser	£10.00
Z88 Pipedream	£16.00	Psion Handling Book	£15.00

ALL PRICES INCLUDE VAT. U.K. POSTAGE £2.50 PER ORDER. - WE EXPORT WORLDWIDE
Phone for details of other QL products OR send SAE for our latest catalogue.
Credit card phone orders accepted, or send cheques, postal orders, Eurocheques, to:

**TK COMPUTERWARE, STONE STREET,
NORTH STANFORD, ASHFORD, KENT TN25 6DF**

TELEX
966676 PMFAB G

VISA Access MasterCard AMERICAN EXPRESS EUROCARD

FAX
0303 812892

QL SCENE

New hard disk interface from Falkenburg

QL Scene (News) QL World Jurgen Falkenburg has produced a new interface board for hard disk connection to QLs via PC hard-disk controllers. Dubbed the QL-HDD-Card, the interface was shown in a system running a reliable 20 MB hard disk in London in November.

Port

The QL-HDD-Card uses the QL expansion port to communicate with a standard PC hard disk controller (which can be ordered as an optional extra), which allows connection of an MFM– (controller type OMTI 5520 or WD 1002A-WX1) or RLL- hard disk (controller type OMTI 5527, WD 1002-27X or 1002A-27X) to the QL. The driver, win_1, is included in an on-board eprom, and is independent of the hard disk being used. Any hard disk, says Falkenburg, is supported, even with more than 100 MB.

The main advantages of hard disks are much greater storage capacity and much faster disk access.

The QL-HDD-Card can be supplied alone, with a controller and hard disk, or as a complete system with the controller, hard disk and power supply. Being modular, the system is very flexible for QL users who have re-cased their QL systems.

For further flexibility, the Card can be adjusted by switches to any rom address in the QL memory, including one of the 16 upper rom slots from $C0000 upwards. The manual describes the addressing process in detail, and it is designed to be done without a deep knowledge of Qdos.

The Card also supports real subdirectories. Switches may also be added to file names to allow subdirectory operations straight from the Psion programs, using short paths with level-2 drivers.

Dubious

The Card does not support through-porting of the expansion support, because this is regarded by many designers as dubious electronic practice. If more than one peripheral is to be used, Falkenburg recommends a bus expander, or his own QL-BUS-Driver with five card sockets as standard. However, Falkenburg also notes that the QL-HDD-Card is 125 mm in width and does NOT fit directly into the expansion slot of a QL in its original case. It will fit directly to the through-port of a ram expansion board, or in a QL-BUS-Driver socket supplied by JFC.

The QL-HDD-Card special adapter for the rom port for Gold Card users will be available shortly. Currently the QL-Bus-Driver cannot be used with a Gold Card.

The QL-HDD-Card export price (for customers outside Germany) is DM 219, or DM 351 complete with MFM- or controller. Information leaflets are available, and both information and products can be obtained directly from **Jurgen Falkenburg (JFC), Thanweg 36, D-7539 Ersingen, W. Germany.**
or from the UK distributor **EEC Ltd., 18-21 Misbourne House, Chiltern Hill, Chalfont St. Peter, Bucks SL9 9UE. Tel. 0753 888866.**

Quanta questions traders' role

Quanta is to hold an Extraordinary General Meeting (EGM) at 2pm on 16 February 1992 at Rayne Village Hall, Gore Road, Rayne, Essex. The subject under discussion will be the position of traders serving on the Quanta committee and the possibility of conflict of interest, after a series of proposals submitted by QView's Jonathan Oakley.

Proposals

Ron Dunnett, Quanta's general secretary, who runs Qubbesoft PD public domain service, said "I think that some of the proposals are probably unworkable but some of them are very sensible. When I joined the committee, I declared all my interests to Quanta, and if anything was handed to me that I thought would cause a conflict of interest, I would pass it straight over to Phil."

To call an EGM, the proposer must present a petition with at least 25 signatures, which the Committee will then consider, and call an EGM if they believe there is a suitable case to do so.

Long-time committee member Dennis Briggs of Adman Services resigned from the committee at the Nottingham Workshop last October.

For more information about the EGM, contact Ron Dunnett, 38 Brunwin Rd., Rayne, Braintree, Essex CM7 5BU, or Phil Borman, 1 Newton Rd., Raunds, Northants. NN9 6LX.

Amiga Emulator

Qubbesoft PD can now offer the software QL emulator for the Amiga. For more information contact Qubbesoft PD, 38 Brunwin Rd., Braintree, Essex CM7 6BU. Tel. 0376 47852.

Another PD service launched

A new public domain software service has contacted *QL World*. SJPD, run by Stephen Jacobson, is offering around 30 disks of public domain software at a copying price of £1 per disk. Blank disks can also be supplied at 50p each. A disk of samples is available to anyone sending a blank disk, self-addressed label or envelope, and return postage, which is a very fair offer. Writers to contribute new software are also sought.

Interested parties should contact **Stephen Johnson at 36 Eldwick St., Burnley, Lancs BB10 3DZ.**

OPEN CHANNEL

Open Channel is where you have the opportunity to voice your opinions in *Sinclair QL World*. Whether you want to ask for help with a technical problem, provide somebody with the answer, or just sound off about something which bothers you, write to: Open Channel, Sinclair QL World, 116/120 Goswell Road, London EC1V 7QD.

DiskOver

I read with interest C A Bland's letter in the July *QL World*. I agree that *Discover* should solve the problem.

I run two QLs, one at home and one for my general work in the office. I am a public relations consultant and compose press release drafts on my QLs. Final copies of everything are produced on an IBM-compatible, and I find *MultiDiscover* indispensible for converting my *Quill* files to the IBM format.

I have been using Textidy and Multi since October 1989, initially to transfer Quill files to a CPM format. At the time our main office system was CPM-based, hence the use of Multi. Also I write material for our local church magazine, which is put together using *Fleet Street DTP*. Converting files to this have been no problem.

Recently I changed our office system to an AST Bravo 286. Again, using Multi presented no problems. The 286 will shortly be changed to a 386, and I am anticipating no problems with that either.

I do not use *Easel*, but I have converted *Abacus* files to both CPM and IBM formats.

I mentioned *Textidy* earlier. I find that I need this to strip out some of the hidden Quill commands to minimise editing on the IBM. Textidy and Discover are easy to use and have similar formats, both being written by Dave Walker. I needed to talk to him in the early days because our CPM was an 'oddity', in that it was an Olympia Olytext system developed from Olympia's electronic typewriters. Dave sorted things out for me and altered Multi's formatting procedure to cope with the unusual. As a result this is now a standard option in Multi. Dave Walker's software is now available from Dilwyn Jones Software.

Keith Tearle
Anderson Associates Ltd
Sidcup

Editor's notebook

First , thank you to those who responded to the appeal for reviewers in the December issue. More good news: Bruce Everiss, founder of the All Formats Computer Fairs, has kindly allowed us to print a £1-off ticket for QL World readers attending a fair in the spring. That should appear in QL Scene next month. Check with Miracle, DP or EEC, or any other favourite supplier, to see which fairs they will be attending.

Not such good news: we have had to postpone the 'post-Christmas-present' article through pressure of time and low response. It seems some people are dubious about plugging other people's products! Come on, let's have some co-operation!

Better news: Dave Walker has been in touch to say that release 2 of the C68 Public Domain C Compiler will be issued just after Christmas. In other words, just about the time you read this.

Happy New Year!

COPY

Surely Peter Tomlin is wrong to suggest that, in the Abacus interest table he designed (*Open Channel* October 1991), it is possible to COPY rows 7 to 15 from the March row (6) and then edit them. In doing so, every row will be edited.

If the intention is to copy the whole block, each row must be copied in its entirety. If only one block is required, the simplest way is to precede D3, H4 and J4 with the dollar sign ($) in both long formulae in row 5, changing the second one to:

($D3+sum($J4:J5 *F5/1200))

Row 5 can then be copied to row 6, rows 5 and 6 to row 7, rows 5 to 8 copied to row 9 and rows 10 to 12 copied to row 13. The only editing required then will be the text in column C. Even that will be unnecessary if the formula:

month(row( )-3)(to3)

is typed in cell C4 and copied to C5.

In designing spreadsheets, it is important to realised that cell references in formulae are translated into relative positional references (eg two up, one to the left) unless, in the case of Abacus, they are preceded by $ or the index function is used.

P H Warne
Newcastle Upon Tyne

Numbers

Despite Brian Storey's contribution to *Open Channel* in the April issue of QL World I stand by my remark that it takes about 30 lines of SuperBasic to avoid a Qdos error message when INPUTing numbers.

My ideal criteria are: 1. No keyboard entry, however unlikely, should be able to cause the Program to crash. 2. Any numerical entry acceptable to the INPUT statement should pass, eg " -3,45e +456 " (with leading and trailing spaces).

I think everyone would agree that the first must be satisfied, but the second is almost impossible, and that is where Mr Storey and I are at cross purposes – he is satisfied with mere positive decimals, while I would like to get closer to the ideal.

Unfortunately, Mr Storey's routine does not even satisfy criteria 1 – try pressing ENTER without any digits or a number with more than one decimal place (eg 2,3,4), and both will crash with a Qdos error. The listing supplied is the most compact uncrashable FuNction I could come up with that satisfies his definition of a number. (Line number counts are not a really useful measure of excellence, especially when multi-statement lines are used!)

I think a FuNction is more appropriate than a PROCedure since it 'returns a value' and that is just what we want here. Lines 80 and 90 illustrate how the FuNction might be called from the main part of a program. The parameter is the channel number of the window used by the INPUT statement. While the use of INSTR is one way to go, the SELECT ON structure combined with the use of CODE is easier to follow than a cascade of IF . . . THEN . . . ELSEs when a sequence of tests has to be made – and about twice as quick, too.

The error flag, er, is set whenever an error is made and there is also another flag, d, which only allows one decimal point (200). You can only get out of the loop at line 230 if no error has ben made. Coersion is implicit at line 250 when a string is RETurned by a *floating point* FuNction. The SELect ON structure can be expanded with more flags to deal with signs, exponential notation, spaces etc., but there is not room in Open Channel to give a full explanation. I have rewritten my original routine along these lines, reduced it to 29 lines, and it now accepts positive and negative numbers from 1E-600 to 9,999E+600, with leading but not trailing spaces.

Howard Clase
Newfoundland

Mouse

I have a QL with a Sandy SQ board and mouse port. I can't find a mouse that works with it. Does anybody know the kind of mouse I need? Does anybody sell it?

Julian Colomina Gonzalez
Madrid
Spain

Editor's comment: We're keeping Julian's letter on file and would like to hear straight from current users/vendors on this one.

More C

As a professional C programmer, what I need is a compiler that saves me time and confusion, especially when I come back to look at code after six months. C has now developed as far as ANSI C, with function prototyping, enums, and standard libraries, as well as C++ with encapsulation, inheritance, etc. I don't mention these because I think I need a more powerful complicated compiler, but because these make it easier for the compiler to catch errors before the program is run, making it easier and quicker for me to produce and bug-free program. How about a SmallC++?

Commenting on *Troubleshooter*, it seems to me that what is really needed for the QL is a connection standard; in other words, an IBM-compatible PC/AT bus, of which there are millions in the world. This makes no difference to the QL Qdos environment or software, but would allow plugging in IBM standard hard disks, fax cards, modems, VGA cards, and so on (although QL rather than PC software drivers would be required). Monitors are more (or less) standard, mice are cheaply available for serial ports, so this would enable everyone to concentrate on software on the only multi-tasking budget machine around. Not everyone can make a pcb in their basement, but everyone can have a go at a world-beating program.

Does nobody make an add-on box for the QL with an AT bus?

Philip Kerrigan
Milan
Italy

Editor's comment: there are a number of private projects going on quietly behind the scenes at the moment, which, quite rightly, the owners won't reveal until they are within sight of completion. Will one of these turn out to provide the type of link-up desired by Peter Kerrigan?

Pink Vikings

Having tried several times to read the article *Valhalla Revisited* in QL World March 1991, I have had to give up each time due to pink Vikings wielding axes, jumping across the page. Is there any possibility you can let me have a copy of the article without the embellishments?

I do hope you do not repeat this underprinting exercise. It may look pretty, but it is totally aggravating if one wishes to read the article. Your comments would be appreciated.

A C Hill
Newton Aycliffe
Co. Durham

Editor's comment: I assure you, my comments weren't appreciated at the time. We issue standing orders that back-prints are to be no more than 10% density. The Vikings should have been the same density as the pale blue strip on page 3 of the March issue, or the screen dump on page 20, but they are, as you can see, much darker. So somebody downstream of our office 'improved' on our design.

And I still catch 20% and 30% back-prints trying to sneak past me.

I will ask Simon Goodwin if he can do a printout of the article which we can send to anyone who sends us an SAE – and, of course, to Mr Hill for his efforts, rivalling those of Alfred the Great, in stopping the Vikings.

Proliferation

I recently purchased *Perfection* from Digital Precision, and after managing with *Quill* for so long I can only praise the DP program.

Regarding the Justify functions (Capt. Eric Starling, *QL World* December 1991), you could type your address (or anything else) on the left of the page, then place the cursor at the top of the page, go to Menu 2, press J, then Menu 2, press E (for reformat hEreon) and magically, your address will now be centred.

Remember to alter the justification for the rest of your document.

What I particularly like about the program is, it seems that you can do anything with the document and still get back to your original version. In Quill, the cursor and line formatting seem to have minds of their own, leaving the document in a mess. You then have to experiment to try and recover the situation to avoid re-typing whole sections.

The handbook is large and comprehensive. I like the extra snippets of information that are dotted throughout for the likes of us who are not steeped in knowledge of computer terms.

I was so impressed that I obtained the spelling-checker and *Lightning*, and have now ordered *Professional Publisher* and *Eye-Q* for easier production of notes with diagrams and drawings.

Bryan Orgar
Ashford
Kent

Editor's comment: One or two people have remarked that we seem to print a lot of letters praising Perfection. It's true: this is because we get a lot of letters praising Perfection. It has reversed the normal trend where a magazine only receives a run of letters if something goes wrong, and critical/quizzical letters outnumber letters of praise. The last time we had a jot of this enthusiasm was after Lightning was released. However, I remember the first time an employer of mine sat me down with Wordstar and said, learn this. After half a day my emotions were a volatile mix of astonishment at its speed and power, and rage at its vast, labyrinthine, unending complexity.

Perfection is comparable with office-standard PC systems like Wordstar, with an ease-of-use factor nearer to Quill. Joy and astonishment would be a natural reaction. For those who have unexpectedly encountered Rage – and we have had all of two, maybe three (there was a borderline case) vexed letters about some aspect of Perfection – I can offer the comforting thought that the answer is usually very close at hand, and that if you were a PC user you might have to use WordPerfect. If in doubt, try the HELP document which is both comprehensive and compact.

In conclusion, I would like to quote from a letter about the Spellchecker, which Freddy himself waved at me after I commented that I had only seen it running on large QL systems.

"I am only using a fairly early and modest internal 512K expansion board, which I know is much slower than Trumpcard, but the speed of spellchecking is phenomenal, and makes the concept of spellchecking on completion of a document the right one, as it is really quick and easy to use." Perfection Spellchecker does not have an auto-replace option for individual checked words, like large hard-disk-based systems – this would make it untenably large – but it is so fast that checking an average document is probably faster.

Now I'm doing it. . .

Les Binns has been using his CQV1 for several months and has found plenty of use for it.

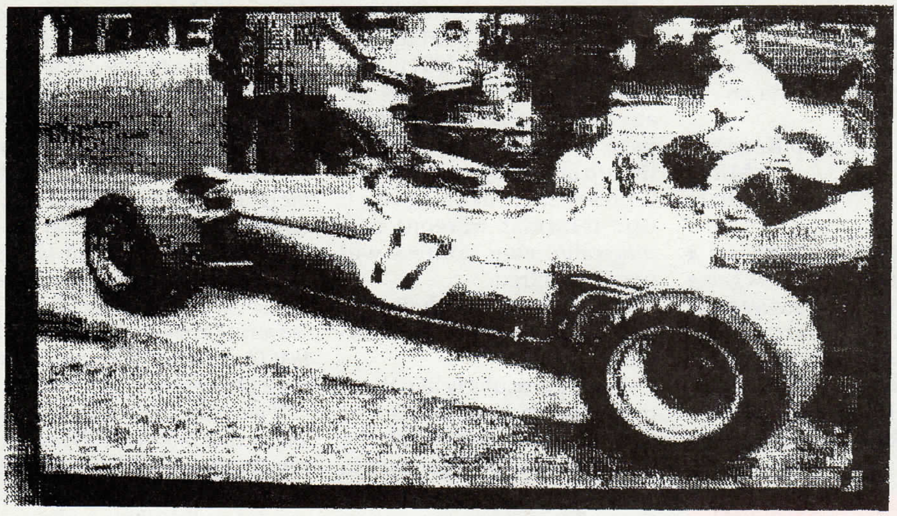
User Report-CQV1 Digitiser

Filling a gap in the UK QL market, the CQV1 is a reasonably priced digitiser at £120, and will appeal to QL users interested in desktop publishing or similar packages. It is an advantage if you already use a video camera but for those without access to one, images may be taken direct from a vcr or other PAL video source, provided that due care and attention is paid to copyright law! Capable of running on an unexpanded QL, the saved images can be treated in the same manner as QL screens, loaded into art or dtp software and manipulated or printed.

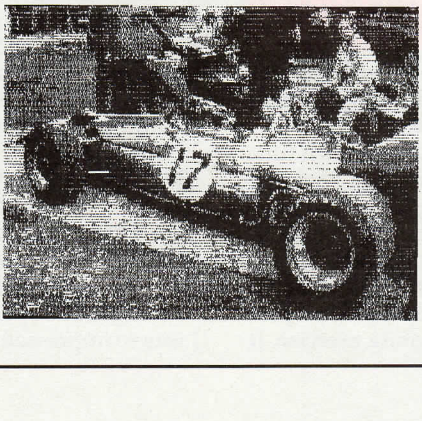
The Digitiser is housed in a glossy black ABS plastic box 180 mm by 110 mm by 60 mm high. Protruding from the lower edge of one long side is a double-sided fibre glass pcb edge-connector to fit into the QL rom socket. On the rear is a phono socket marked I/P, for the PAL video signal source input, and a small diameter red adjustment knob marked 'B.LEVEL' for black level setting. It is guaranteed for 90 days against faulty components and workmanship, and offers a choice of 3.5 or 5.25 in disks or mdv for the controlling software, and an 8-page A5 manual.

My first thought on examining the digitiser was that some people, myself included, who have built the QL into a KBL casing with drives and psu, would need to build a jumper lead to connect the rom socket to the unit. I had brought the rom socket to the rear of the casing but positioned it too high to allow direct fitting of the digitiser without placing the entire weight on the pcb connection. Those using the QL as purchased should have no trouble fitting the unit as the edge connector is at the correct height, provided the QL remains flat without its ridiculous plastic feet! As the manual stresses, the CQV1 should be raised to the QL's level if a different typing angle is used. In my case, the temporary solution has been to remove the rom edge — connector socket from the rear casing and use it as a flying lead, sitting the CQV1 on the top of the computer casing, which conveniently brings the PAL input socket the front of the unit. A market obviously exists for some enterprising soul to make a suitable jumper lead, because the convenience would be enhanced if the CQV1 wasn't yet another physical parasite of the QL motherboard.

The manual describes the characteristics of the CQV1, the installation of its hard and software, its use and, finally, gives application examples with basic programming and some sample prints of digitised pictures. Safety measures are covered sensibly during installation and any non-technical person should find the instructions easy to follow. Backing up the master software is also straightforward, but although it is capable of multi-tasking no assistance is given to those wishing to incorporate the boot program of extension codes into a front-end, or a different program boot like *Eye-Q*, that you may wish to multitask it with.



The picture above was printed from the digitiser screen dump. The same picture on the right has been loaded into DP's Pro Publisher through the Mode 8 texture selection and printed after the addition of this text. The image was digitised from a Black & White negative using a video camera set to macro and tungsten lighting, diffused through white paper. The "negative" command was used to reverse the image to positive prior to saving as a screen image.



Sample display

Having installed the extension codes and executed the main program, 'IMAGED', the screen quickly displays the program title and a sample of digitising above a command menu and status box. The commands are fully explained in the manual and are straightforward in their use as can be seen from the following list:

F1 opens Menu window; F2 saves screen image in buffer to file; F3 loads screen image from storage medium; F4 samples video from input. This clears the screen of menu/status box but a flashing cursor remains in view at the bottom right of screen to signify that sampling is active; F5 clears screen; shift/F2 stores the last video sample in temporary buffer; shift/F3 recalls stores image from buffer to screen; control/F3 recalls buffer image and 'ORs' with screen image; control/shift/F3 recalls buffer image and 'XORs' with screen image (Note: This command is not shown on the screen menu); D toggles between three display types (quarter screen (128 multiply 128 pixels), positioned on screen with cursor keys; four quarter screens, each being refreshed sequentially; full screen of 256 x 256 pixels); S toggles between video sampling modes, continuous or keyed; N changes display to negative image; I inverts displayed video; P prints the displayed image as a 1 : 1 dump.

All commands except F2, F3 and P are usable during sampling without the screen menu being called up (F1) but this only became apparent on reading the full command explanation for F4.

Camera to monitor

My first trial with the CQV1 was using a video camera set to monitor (in the same manner as you sometimes see in shop windows) which, with the controlling software running, gave a picture on screen in the blink of an eye when a sample was taken using the space bar. Switching the mode to continuous gave the sampling iteration rate of 20 msec (data from manual) and toggling the display to quarter screen gave five frames per second, similar to an old handcranked movie! This partially substantiated the claim to monitor moving pictures on one corner of the screen while editing text on another. Could this be the answer for computer-game-playing security guards!

All aspects of the software and hardware ran smoothly and quickly but file in/out commands do not allow for directory searches, so the absent-minded could be faced with the need to exit temporarily to SuperBasic. This in my opinion could have been easily remedied but is sometimes overlooked as unnecessary due to the QL's multitasking capability. A saving grace is a warning given that a file already exists followed by the option to overwrite it.

The black level adjustment allows the user to set the 'blackness' of the picture, giving control over the range of tones in the finished print. The effect can be equated to photographic print exposure: too much will be too dark, losing shadow detail, while too little will give a whitewashed effect in highlights. I found it easiest to determine the correct setting using quarter screen and continuous sampling where the time lag between pot adjustment and screen effect was minimal. The print command is limited to a 1 : 1 dot matrix dump for Epson-compatible 9-pin printers, using codes ESC L and ESC J to produce 120 dots per inch with an n/216 line feed.

Device name

The manual and menu are a trifle confusing when using the P print command. It states that the device name for printing 'is not limited to SER1H or SER2H, it could be any valid file name'. Does this mean that printer files can be spooled or does it mean that saved files can be printed. Either way I must be doing it incorrectly as I have been unable to use flp2_nnnn to spool or print saved files. At first I had difficulty in printing files with the P command from screen. The printer driver cannot be configured by the bundled software so if your printer is not 100% Epson compatible you should request assistance from CL Systems, who have shown willing when I approached them with my difficulty. They were very prompt in sorting out this problem, which entailed a small charge in the boot program.

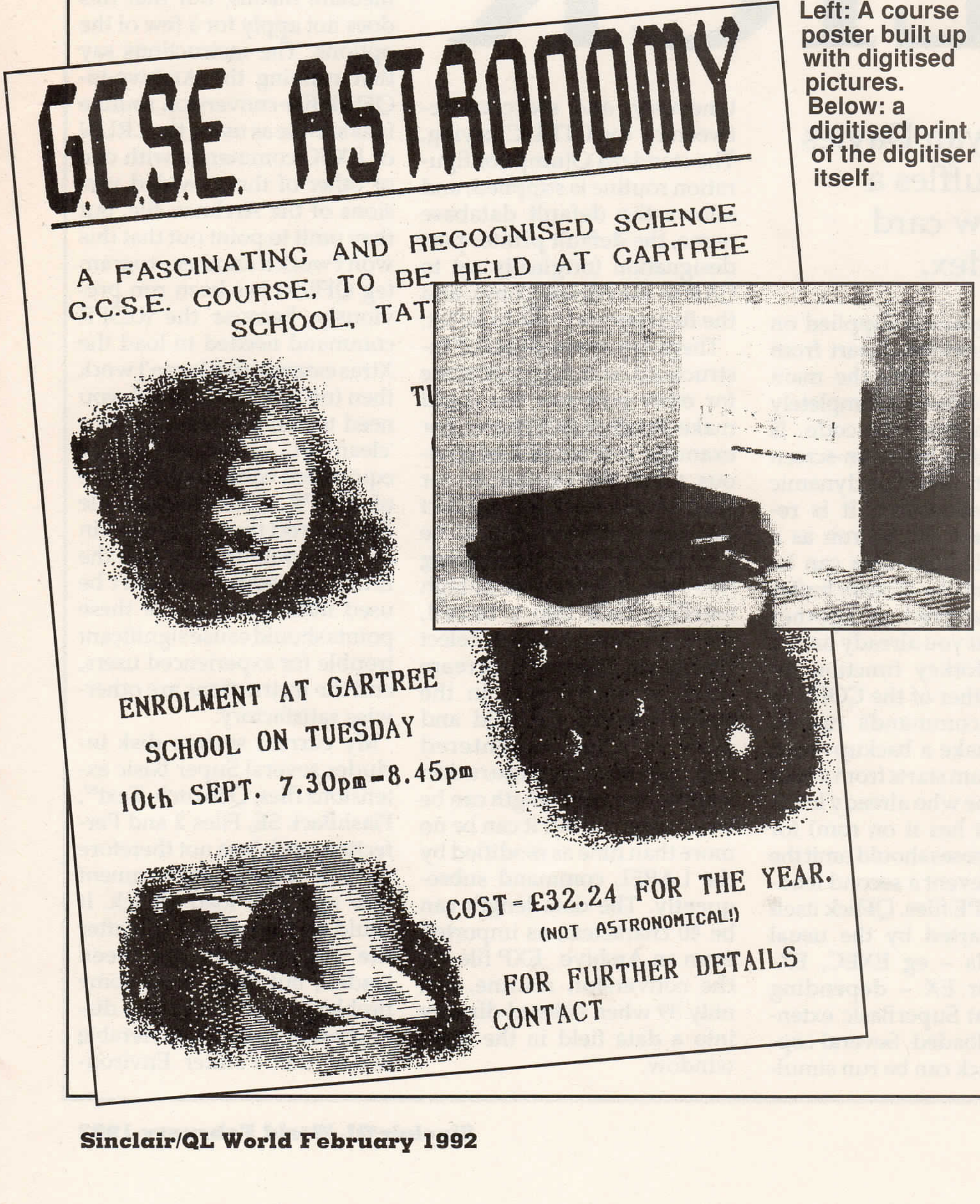
My printer is a Smith Corona D200 that works well with dot matrix dumps from *Easel*, *Eye-Q*, *GraphicQL* and DP *Pro Publisher*, yet selecting P and entering SER1 gives jumbled characters, printing only intermittently after recycling the printer driver file may require alteration in my case. The prints I have obtained are of good quality with a dense dot pattern giving a good range from white to black. Of course, some users are more likely to use other programs than dump direct from the CQV1. Not however, that when saving screens for later use in Eye-Q it is advisable to use the Negative function to invert the screen image prior to saving, as Eye-Q prints black as white and vice versa.

What users can the digitiser be put to? Your imagination will probably be the limit. The manual suggests slow frame recording (for biological growth?), gives a Basic listing and supplies the file to do it. Written for use with a 720 K disk, the program will store 21 full-screen images at time intervals set by the user. The given example can be altered to save only part screens thereby giving room for more screen on each disk. MDV users can also use the program with some minor changes.

From negatives

One possible use not yet mentioned is to 'print' direct from photographic negatives or positives. I have experimented with colour slides, and both black and white and colour negatives with moderate success. Naturally the results are not as good as the real thing but you have another medium for gathering clip art of dtp pictures.

I welcomed the CQV1, especially for the numerous times I am asked to produce posters or to dtp for local clubs. Where one of my amateurish drawings took me several hours and pots of tea, I can now expect professional polish in a fraction of the time. My main difficulty, that it could have been more flexible if a jumper lead had been used to interface it to the rom socket, is one that will apply to only a few users.



G.C.S.E ASTRONOMY

A FASCINATING AND RECOGNISED SCIENCE G.C.S.E. COURSE, TO BE HELD AT GARTREE SCHOOL, TAT

TU

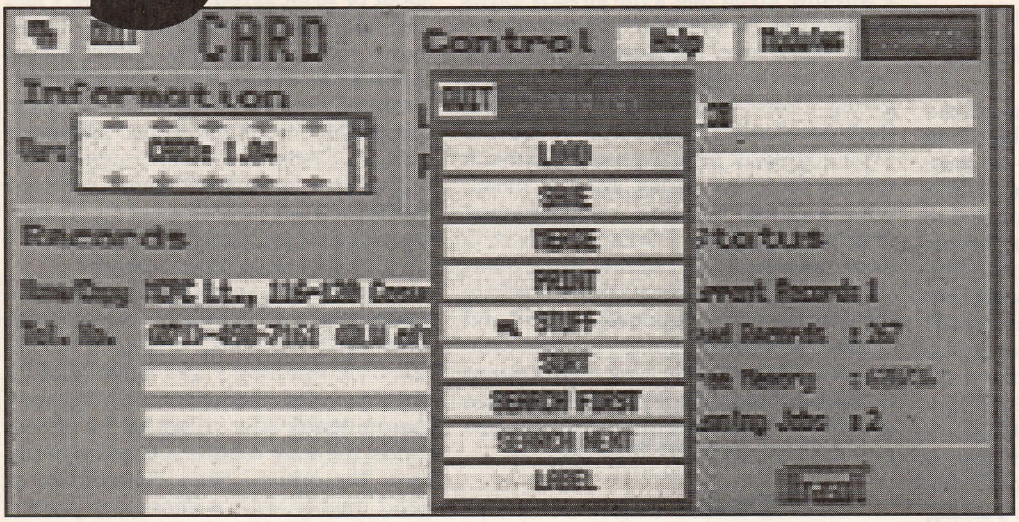
ENROLMENT AT GARTREE SCHOOL ON TUESDAY 10th SEPT. 7.30pm-8.45pm

COST = £32.24 FOR THE YEAR. (NOT ASTRONOMICAL!)

FOR FURTHER DETAILS CONTACT

Left: A course poster built up with digitised pictures. Below: a digitised print of the digitiser itself.

SOFTWARE FILE



QFLICK

Bryan Davies shuffles a new card index.

QFlick is described in the instructions as a small user-friendly card index database utility to handle simple database applications, which seems a fair description. It is an alternative to *Archive* or *FlashBack*, for small jobs. It does not have the programming language of Archive, or the flexibility of FlashBack, but it is easy to use. Originally sold by Ultrasoft from Germany, the program has been taken over by Digital Precision; QL users who have bought programs from DP in the past will no doubt feel more confident buying from them, rather than from less-well-known companies. The origins of the program are confirmed by its use of the QJump Pointer Environment, which is included on the program disk. The Pointer Environment seems to have been taken up with more enthusiasm by programmers in continental Europe than by those in the UK. Anyone familiar with Qram or Qpac will find the QFlick user interface essentially the same as with those program suites.

The program is supplied on disk or cartridge. Apart from the use of the PE, the main features are that it is completely written in machine code, is multi-tasking, has an on-screen help facility, and has dynamic memory allocation. It is re-entrant, and can be run as a Hotkey file, and data can be 'stuffed' into the Hotkey buffer and transferred to another program, if you already have a QJump Hotkey function installed. Either of the COPY or WCOPY commands can be used to make a backup copy. The program starts from a boot file; anyone who already loads the PE (or has it on rom) for other purposes should omit the boot to prevent a second loading of the PE files. QFlick itself can be started by the usual commands – eg EXEC, EXECUTE or EX – depending upon what SuperBasic extensions are loaded. Several copies of QFlick can be run simultaneously, and switched-between by the CTRL-C keying. The standard QJump configuration routine is supplied, and allows the default database name, the default printer port designation (originally set to PAR in the review copy), and the field names (1-8), to be set.

There are seven pages of instructions, which are adequate for most purposes but could make certain points clearer. For example, it is not at first obvious what the procedure for searching is: you have to select the data field which is to be searched, type the search string into that data field as though entering data into a record, press ENTER, and then select the Search command. There are small inconsistencies in the stated maxima for field and data lengths; as entered through the configuration routine, the field length can be ten characters, but it can be no more than nine as modified by the LABEL command subsequently. The data length can be 40 characters, as imported from an Archive _EXP file via the conversion routine, but only 39 when entered directly into a data field in the main window.

INFORMATION

Program: *QFlick Card Index System V1.04*

Price: £29.95 (plus 5 or 10% for overseas buyers)

Supplier: Digital Precision Ltd
222 The Avenue
Chingford
London E4 9SE.
Tel: (081) 527 5493.

The listing of key combinations for 'shortcut' ways of invoking commands does not make it fully clear that the use of CTRL-SHIFT before the listed single keys will activate Modules and Commands options directly, without the user having to go through the intermediate menus, but that this does not apply for a few of the options. The instructions say that running the Archive-to-QFlick file-conversion routine is as simple as using the LRUN or EXEC commands with one or other of the provided versions of the Archnav file, but they omit to point out that this won't work if another program (eg QFlick) has been run previously, because the RESPR command needed to load the Xtras extensions file can't work then (on a JM QL at least); you need to run the program on a 'clean' QL, or use ALCHP or equivalent command in place of RESPR. (The Updates file does correct the instructions in so far as pointing out that the EXEC command shouldn't be used initially). None of these points should cause significant trouble for experienced users, and the instructions are otherwise satisfactory.

My current system disk includes several Super Basic extensions files, Q Switch, Text^{87}, FlashBack SE, Files 2 and Perfection, and does not therefore provide a good environment into which to load QFlick. It could be loaded and run after the system disk had been loaded, but there were some problems with programs disappearing, and it is preferable not to mix Pointer Environment-based programs with others loaded 'outside' the PE. The ARCHNAV routine provided to convert Archive _EXP files to QFlick format, or QFlick _DB files to _DBF format, did not seem to be able to multitask with the other jobs, despite being EXEC-ed and having a cursor. It just beeped when any attempt was made to get out of it. The QFlick program file is 15KB in size, and it (and the required extensions) took 111KB when running; loading an 88KB database raised the total space used to 207KB.

Pointer

The main program window is shown in the first illustration. The diamond-shaped pointer can be moved by the cursor keys or a suitable mouse (presumably, the Qimi one). As with other PE applications, the shape of the cursor changes with the area of the window it is over; for instance, a padlock shape indicates that the cursor has been moved outside the usable area. Movement of the cursors is very smooth. The QJump conventions for use of the cursor/pointer are followed, and there is a fairly wide range of them, which needs a bit of study if you want to streamline your use of the program. There are several ways to perform most functions. Data is entered into a record by either:

1) moving the cursor (by cursor keys or mouse) to the field data area, so that a box appears around it, pressing the Space Bar, typing-in the data, then pressing ENTER, or

2) selecting the data area by means of the 1-8 keys when in the main window, typing in the data, and pressing ENTER.

To have a new record actually inserted into the database, you use the INSERT command:

1) cursor to Modules, Space Bar, cursor to INSERT and Space Bar or just press I, or

2) simply press CTRL-SHIFT-I.

Selection of commands is similar:

1) Move the cursor to the Modules or Commands panel (as appropriate), so that a box is displayed around the panel, press the Space Bar or left mouse button, cursor to the required command and press the Space bar or just press the appropriate letter for the command. It is unfortunate that the standard convention of highlighting the letter of a command on a menu, to select that command, is not followed; unfortunately, it would not be possible to do this with some of the chosen letters anyway – eg the letter is V for LABEL.

2) press f2 (for the Modules menu) or f3 (Commands menu), cursor to the required command. For example, CTRL-SHIFT-L will load the default database. Most of the letters used are mnemonics (as with the L for Load), but some are not (eg Y for Clear Display). Which method you choose depends to some extent on whether or not you have a mouse in the system; it may also depend upon how good your memory is for key combinations, but the key shortcuts are undoubtedly the fastest way. The ways of selecting functions may cause some confusion at first, but there should be no difficulty once a particular method of operation has been chosen and the required keypresses or mouse movements memorised.

The Information panel gives the version numbers of QFlick, the Qptr interface, the Qman windows, and Qdos. The Control panel has small windows for Help, Modules and Commands. A sample Help window is shown in the second illustration; other pages of help can be obtained via the NEXT and BACK commands, from the Help window itself. The Modules panel allows access to the commands relating to individual records in the current database, as shown in the third illustration. The Commands panel gives access to commands which are concerned with the current database file as a whole (fourth illustration). You need to be a bit careful when 'backing up' from one menu level to another, because what might seem like the obvious key for doing this – ESC – will kill the program if pressed when only the main window is displayed, whereas it does move you back to the previous menu if used from the Modules or Commands menu. The function of the Status panel is fairly self-evident; the 'running jobs' figure will be the total of all jobs running in the QL at the time, as is also recorded by the SuperBasic Jobs command. The free memory figure can go negative – if you get no response from the Help, Modules or Commands panels, have a look at this figure.

The important panel for data entry and manipulation is the area 'Records'. There are eight slots here, for field data. To the left of each slot is a space for the name of that field. You can't get into the field name areas using the cursor – you have to go via the Commands menu and the Label command. A maximum of 10 characters can be used for field names; there is no requirement for the $ symbol (a feature of Archive) to be used.

Conversion

Users will wish to be able to make use of existing Archive database files, and that is possible. The supplied Archnav conversion routine is a bit sparse – eg it takes no account of available space when starting on a conversion, and just dies if it runs out of space to store the converted file – but it does the job. It would normally be used only occasionally, perhaps just when files are transferred initially from Archive to QFlick. The fields and their names from the Archive file are put into a _DB file, within the restrictions imposed by the latter's structure; field names can be up to 10 characters in length, and data in fields will be truncated if it exceeds 40 characters in length. The field names can subsequently be changed from their Archive form to whatever is chosen in QFlick by using the Label command; there is no requirement to have the $ ending. The RU command of any version of *The Editor* can be used to load _DB files, or the Archnav-converted versions (for transfer to Psion programs), for editing.

The 267-record file containing the record on display in the main window (first illustration) was created by the Archnav routine from an Archive name and address file, in a few seconds. The original Archive _DBF file was 25KB, the _EXP file 21KB, and the converted, _DB, file 88KB. A couple of attempts to convert the resulting _DB file back to an Archive _EXP file appeared to be successful, but Archive would not Import the resulting files. A quick check of the file structure didn't reveal anything obviously wrong. There is another conversion routine, supplied for users of version 1.00 of the program, to make their database files suitable for use in 1.03 (and, presumably, the supplied 1.04 too), although this is not mentioned in the instructions.

The program works smoothly and reasonably quickly, with no untoward behaviour. A mouse is a highly desirable accessory when using any program which is 'fronted' by the Pointer Environment. At first try, the new user might well find this program rather cumbersome in operation, as using the cursor keys to move the mouse pointer emphasises the rather pedantic nature of this aspect of the interface, with lots of key action being needed to accomplish most tasks. However, careful study of the instructions, and some experimentation, will show that the provided keyboard shortcuts allow you to bypass most of the keyboard actions. The program then becomes very much pleasanter to use. Harking back to comments made in the review of another database program not too long ago, it is important to allow for the knowledge level of the majority of likely buyers when writing the instructions, and this seems particularly important when the Pointer Interface fronts the actual application program. Fortunately, Digital Precision have realised this and provided instructions which give sufficient guidance to allow the most to be got from the database facilities. No confirmation request is issued before a Load or Save is carried out, and this could cause trouble if you have omitted to make a desired change to the file or device name in the Load/Save slot on the main program window. Similarly, the Delete record function does not request confirmation. While omitting a confirmation request streamlines operation, it is perhaps safer to have one, for most users.

This is not one of DP's major epics, but it is a capable program, at a moderate price, and particularly-suited to use under the Pointer Environment.

MIRACLE SYSTEMS

MASTERCARD VISA

QL 3.2 Megabyte Dual Disk Drive

New standard for the QL

£175 inclusive (£155 export)

• 2 year warranty • 14 day money back guarantee •

The 3.2M byte DUAL DISK DRIVE together with the GOLD CARD brings the QL system right up to date. The drive contains 2 mechanisms each able to store 3.2Mb on an ED diskette. ED actually stands for 'Extra High Density' and refers to the rate at which data bits are read from and written to the disk. ED disks transfer data at 1 Megabit per second whereas HD (High Density) transfers at 500K bit/s and the old QL standard DD (Double Density) at 250K bit/s. Please note that these are instantaneous rates and that to work out file transfer times other factors must be considered.

The drive is fully backward compatible in that it can read and write both DD and HD diskettes with any type in either drive. The DD, HD and ED diskettes are all the same physical size known as the 3.5" form factor. The HD and ED diskettes have an extra hole in their casing, in different places, so that the drive can detect which type of diskette has been inserted. This means that there is no need for the user to tell the QL of the diskette type in use. For example, if you format a DD it will format to 720K, whereas an ED would format to 3.2M.

The drive is fully cased with cable and integral 220/240V power supply and comes with 10 ED diskettes.

We are exhibiting in:

LONDON:

18th January 1992
Royal Horticultural Hall
Greycoat Street,
Victoria

OTHER PRODUCTS:

(more details about our products may be found in the QUANTA Magazine)

ITEM	PRICE	
QL CENTRONICS	£29.00	(£28.00 export)
QL 40M HARD DISK	£449.00	(£405.00 export)
* TRUMP CARD 768K	£100.00	(£89.00 export)
* TRUMP CARD 2	£150.00	(£130.00 export)
* EXPANDERAM 512K	£50.00	(£46.00 export)
5.25" DISK DRIVE	£95.00	(£88.00 export)
DISK ADAPTER	£15.00	(£15.00 export)

* reconditioned product

MIRACLE SYSTEMS Ltd, 25 Broughton Way, Osbaldwick, York, YO1 3BG, U.K. Tel: (0904) 423986

SOME THOUGHTS ON DISK STORAGE

As alternatives appear, Bryan Davies reconsiders his approach to mass storage.

Back in 1990, most of us were still managing to store our mass of files on cartridges or floppies, but some were getting increasingly frustrated by the lack of space on single media. Cartridges had always been too expensive and unreliable. Some computers still used 360 KB floppy disks as the basic storage medium, so we were not the worst-off; our 720 KB disks stored a reasonable amount of the program. If you were in the habit of loading several programs, and switching between them though, 720 KB was a pretty small pot to get all the required system files into. The question was, was it worth spending over £400 for a hard disk?

Lots of money

That was, and is, a lot of money for most of us, and relatively little was known about hard disk on the QL, making it rather a blind decision to make. Those of us who had experience of hard disks on other computers were at least familiar with some of the advantages and problems. The decision was made easier for me by the arrival of a Miracle hard disk unit on my workbench, brought to me by the proud new owner, with the instructions to 'set it up'. I could find out the cheap way what hard disk on the QL was like.

There was an obvious interest in hard disk within the QL community, and the setting-up process would generate enough material for an article. That article sits before me on the screen now; after a struggle, it got completed, but somehow never got into printer at the time. Looking at it now, it is rather ridiculous to read.

So what changed? One user's requirements are not identical with those of another, but there are almost certain to be common items on the 'wants' list of most new hard disk owners. High on the list is the ability to put files into separate compartments, in order to be able to identify particular groups from amongst what may amount to thousands. A sub-directory structure is needed, and the Miracle hard disk is not well-provided in this area. There *is* a sub-directory structure, but it is short on commands and hard to deal with, and it does not allow convenient running of several programs from their own sub-directories.

In a sense, my wheel has turned full circle. The article on hard disk arrived back for updating at the same time as I decided to remove from my friend's hard disk the sub-directories already created. That is, I'd decided he was going to have to use the 40 MB storage space like one enormous floppy disk, with all the files going into the root directory. No doubt some other hard disk users have been doing that all along.

That isn't the end of the story, however. There has been a new development recently. Miracle have come out with what looks like a major boost for the QL, in the form of the *Gold Card*. That in itself will add new life to the QL scene, by allowing existing programs to run much faster, and thereby rekindling the interest of users who are fed up with the snail-like progress of work done on Quill and its associates. More to the point in the present context, the Gold Card comes complete with the ability to handle high-density 3$^1/_2$in disk drives, giving the possibility of storing 1.44 MB on each disk. Even better, there is the prospect of 3.2 MB drives being available for the Gold Card in the not-too-distant future. Maybe it is time to rethink what we really need, to get around our storage problems?

Boots growing

The initial reason for my getting itchy about lack of storage space was not the need for masses of floppies to keep files on, but the fact that my boot system files were growing in number and size and could not comfortably be held on one disk. Each time a program revision arrived, there would be a time-wasting session, trying to delete yet another file to allow for the (inevitable) extra space the revised version took compared to its predecessor.

Quite simply, the one, vital disk in my collection was no longer coping with the strain. Incidentally, it really is one disk, as I still use the actual disk bought from Eidersoft way back in 1985 and set up as a boot disk; it is numbered eight in a collection now running to several hundred and must have done many, many miles.

That's one reason for wanting a larger disk. Undoubtedly, some other users will be much more concerned about the increasing space taken by their disk collection, as the number of files increases overall. These users may not even load more than one program at any given time; there will surely still be some who run only Quill most of the time but, nevertheless, create many document files. Disks and their boxes take up much space, and cost a lot of money. Careful buying at computer fairs can keep the costs in check, but cannot do anything about the space problem.

There is also the retrieval problem. Unless you keep detailed and up-to-date records of what is on each disk, you find yourself having to check directories of several disks to find the file you want. This situation arose for me quite early on; within a few months of buying my first QL in 1984, in fact. All my work was done with Quill, and there were plenty of individual documents and quite a few very large ones. One of the documents created in microdrive days ran to 288 KB when assembled (later) onto disk, from about 25 separate files. Of course, many of those documents are long gone, because they could not survive the dubious storage capability of cartridges; they were destroyed by the system, in other words.

No leap

Disk drives became readily available, at a reasonable price, about a year after the QL entered my life. The storage problem was ended, overnight. Not only was there a lot more space on each disk, but the files were actually still usable months or years later. No more 'bad or changed medium' messages (well, very few anyway). Things were quite good for a year or so, but the situation deteriorated once again. This time, there was no answer for the QL; the number of files was into the hundreds, but there was no 'quantum leap' in storage method available. For this, and purely commercial reasons, a PC with hard disk took its place alongside the QL. There was just no way the QL could cope sensibly with 50-100 MB of files, then or now. Leastways, that was the way it looked until the Miracle hard disk arrived on the scene. But I've already said that this doesn't really provide what I need, because I want files **and programs** in separate areas. Could the new high-density disk drives, possibly with hard disk too, provide a viable solution?

Accepting the impracticability of running several programs from separate hard disk sub-directories, there is still the option to use high-density floppies for system files, and the hard disk for data files. As data files can be identified fairly easily by their extensions – DOC,_T87, ABA etc – there is less of a problem finding particular ones when they are all in the root directory of the hard disk.

Given some serious thought and work, it would be possible to set up a relatively simple sub-directory structure, for data files only, and use it with those programs that can cope with device names longer than about five characters. Hopefully, some kind programmer will eventually produce a decent, cheap utility program to permit simple and rapid housekeeping operations in a sub-directory environment. Disk operations should not be slow with the Gold Card; it is claimed that floppy operations will be at a comparable speed to hard disk ones on the basic QL. As program loading may take less time overall than the loading and saving of data files, continuing to load programs from floppies may not be a serious drawback.

More floppies

There is an additional operational factor to consider. A hard disk does not make copies unnecessary; you still need backup copies. With the growing collection of data files on your hard disk, you can be in queer street if any access difficulty occurs, and that is not an uncommon occurrence with hard disk. A backup set of files must be kept on floppies. This brings up an unsatisfactory area even on a computer with 50 MB or more of hard disk storage space; making backups to 720 KB floppies can soon get you used to 'no room at the inn' messages. 1.44 MB floppies ease this situation considerably, but not for ever. It is good to be able to contemplate the next step mentioned in the Miracle literature – to 3.2 MB – in advance of the need for it. At the moment, I'm not aware of 3.2 MB drives being offered for PCs in this country, so we may be one-up on the PC world there (but these drives are standard on some computers sold in the Far East). For the record, there are *floppy* drives which give as much as 25 MB capacity per disk, with a quoted price of about $30 per disk (£15 or so) but it'll be quite a while before such capabilities become readily available.

It may be that some readers will not be familiar with 1.44 MB drives, let alone 3.2 MB ones, so a few details are appropriate at this point. Physically, $3^1/_2$in 1.44 MB drives look just like your existing 720 KB ones. There's no difficulty installing them. The disks themselves look essentially the same. The one difference you may notice on HD disks is that there is a matching square cutout on the opposite corner of the disk from the write-protect slide. This extra cutout is used by some drive mechanisms to detect a HD disk. While a standard 720 KB drive will not read 1.44 MB disks, a 1.44 drive will read and write both 1.44 MB and 720 KB disks.

In the case of the drives being offered by Miracle, it looks as though the driver software, rather than the drive mechanism itself, detects what the capacity of an inserted disk is. This allows the user to format DD (720 KB) disks to 1.44 MB and use them in the HD drives. It remains to be seen whether or not this practice is satisfactory on the QL, but it works on *some* PCs.

The identifying cutout is not made for nothing – some drives will not format disks to 1.44 MB unless they have this cutout – but it looks as though we don't have to bother too much about it.

Quoted capacities can be a bit misleading, because both unformatted and formatted figures can be given, and different operating systems produce different formatted capacities. Our standard 720 KB formatted-capacity disks can be referred to be as 1 MB *unformatted capacity*, or 800 KB formatted on some computers. For our purposes, the actual usable storage space is 717 KB, because there is a small amount of information put onto the disk at format time for the benefit of Qdos. The normal 1.44 MB formatted-capacity disks are sometimes referred to as being of 2 MB unformatted capacity. (Likewise, $5^1/_4$" high-density disks can be called 1.6 MB unformatted or 1.2 MB formatted).

Just what the available space for QL users will be, I don't know, but presumably something over 1.42 MB. The sector size should still be the same, at 512 bytes, so that a format will show 2,880 sectors (1.44 x 1000 x 1024/512).

Backups

The much-greater capacity of hard disk soon brings you face to face with a major problem; backing up files. If you don't do that regularly, you are either a bit daft (to put it mildly) or are not doing anything important enough to justify hard disk anyway. My own time-scale for backing-up word-processing files (like this article) is every 5-10 minutes, the larger figure only being used on very large files, when more-frequent backing-up would begin to interfere with the input of words. When the data you enter is already on paper, you are in a position to retype it without straining the grey matter; all you lose is time and temper. Should the material be 'off the top of your head' (maybe you write novels?), the idea might not be recaptured if retyping becomes necessary through loss of the original file. Losing 720 KB from a floppy can be a pain, but not anywhere near as much of one as losing 40 MB from hard disk.

What is needed is a means of backing up large quantities of files, quickly, onto small quantities of floppy disks. Not surprisingly, that facility was not included with the hard disk software; such things are normally left to software houses to provide. One software supplier did advertise a program for the job – *HardBack* – but the supplier was PDQL, and we all (I hope) know by now that this has not proved the best source to get software from these days. The program should be good, being written by Chas Dillon, but it is an open question whether or not it will become available from other suppliers in the future.

Dilwyn Jones Computing has introduced the program *WinBack*, which should provide essentially the same facility as *HardBack*. In its initial version, *WinBack* does not permit backup files to be split between floppy disks, that is, if the next file to be backed-up is larger than the remaining space on the current backup disk, another disk has to be inserted. This can create a lot of wasted disk space, if files tend to be large. A later version of the program should allow the splitting of files, to avoid the 'waste' of disks.

Storage space appears again on my list of reasons for buying the hard disk, but this time in a somewhat different context – use of the *PC Conqueror* emulator, to run MS-DOS programs. One thing current MS-DOS programs aren't is slim and trim; put simply, some of them are grossly overweight. A PC with only floppy drives is close to being unusable in practice, because major programs do not fit onto one disk and you are forever swapping disks. On old PCs, which were slow anyway (even with hard disk, in some cases), being distracted by demands to swap disks can be enough to completely put you off whatever work you are trying to do. Although any emulator is likely to run slowly compared to what it is emulating, a QL with Conqueror and a hard disk may be a better bet than an old PC with floppies. You may well not use more than 5-10 MB of the hard disk for Qdos files, so giving (say) 10-15 MB to Conqueror for MS-DOS should present no problem to most users. Of course, with the Gold Card, it should even be acceptably speedy for regular use. . .

The hard disk sub-system makes use of about 55 KB of QL ram memory, making it quite pointless to consider adding a hard disk to a system which is not fitted with expansion memory. 128 KB less 32 KB for the display and 55 KB for the hard disk leaves little memory for running programs. If you use *Ice* or *Files 2*, or some similar housekeeping program, you would end up with maybe 20 KB to run an application program in Quill for one would soon tell you what to do with that measly allowance . . . The hard disk and Gold Card are compatible, as are hard disk and Trump Card. Going either the hard disk or the Gold Card route leaves the rom port free for whatever programs you have on Eprom, whether it be Ice or a collection of utilities (maybe program-development tools) specially blown onto Eprom.

The Gold Card projects only a small amount from the QL, whereas the Trump Card extends the length considerably; there is a big difference in the space taken by a system with Trump Card and hard disk, compared to one with just the Gold Card (assuming both systems would have the same number of floppy drives). In the hard disk's favour, there is no obvious reason why the size of the drive unit should not be increased, possibly with the existing interfaces. Don't take my word for that, though; check with Miracle before starting on an upgrade. The drive inside the Miracle box is almost certainly 'PC-compatible'.

The well-heeled user may opt to make Miracle happy by buying hard disk, Gold Card, *and* high-density floppy drives. The rest of us, being budget-conscious, have to make a decision. The reason for buying the hard disk is, primarily, the desire to have large numbers of files quickly and easily accessible at any time. In a word, convenience.

The floppy drive is reliable, and hard disk should not be bought to combat reliability problems; if your floppy drive is unreliable, the chances are you are not treating it properly. Try buying a disk head-cleaning kit, and using it regularly. The user who lets floppy drives get in bad shape may well cause hard drives to go the same way; it's not a question of cleaning here, but of general handling. Don't bang the drive box about, and don't move it unless the drive heads have been 'parked' (this is done automatically with the Miracle drive, provided you don't switch off when the drive heads are still accessing data).

A secondary reason for purchase may appear to be the need to save space but, if you take the business of backing up files seriously, you will realise that space for many floppy disks, and their boxes, is still required, although half the number of floppies means roughly half the amount of space. Unfortunately, hard disk box is so large that it take up the space of the equivalent amount of floppies and their box.

You come back to speed of loading and saving, and the amount of instantly-available storage space, as the main reasons for buying hard disk. In rough terms, the speed is around three times that of 720 KB floppy drives. If you buy the Gold Card and 1.44 or 3.2 MB drives, the speed differential may become less of a major consideration. That's a lot, but 50 floppies don't cost too much (less than £20) and don't take up that much space either. It isn't hard to argue that improvements of this order are not worth the money.

In addition, there is the not-inconsiderable matter of having to locate a hard disk unit very close to the QL, because of the short connecting cable supplied. There are presumably good technical reasons for the short cable, and it doesn't seem a good idea to try and substitute a longer one. My own solution to the location problem was to turn the hard drive unit upside down and sit the QL on top of it; this enabled me to get away with no major revision to the system layout (which there's no space for), but it does preclude the use of a rom cartridge such as Ice, because the port is covered by the workbench.

Files 2 has proved an adequate replacement for Ice, so *that* drawback doesn't bother me unduly. The elevated position of the keyboard is rather hard to get on with, and – in the long term would be solved by mounting the hard disk under the workbench, with a cutout for the connecting ribbon cable. What is of much more concern is the noise, which is sufficient to take precedence over my noisy PC and laser printer, but current production units do not have a fan installed and the noise of the drive itself should not be too much of a distraction.

You can't generally buy software already installed on hard disk (the *Next* computer comes loaded with software, but it is an exception), and floppies are still the almost universal medium for distributing software. The very features which make the hard drive such a boon are also a considerable nuisance. Its sealed construction, and 'hidden' location within the bowels of a computer or add-on box, make it something that can be – to a fair extent – used and forgotten. These features also mean that you have to be able to trust it, for there is no simple changing of a cheap, push-in disk when error messages are seen on the screen. Whichever way you look at it, you are likely to be using floppies for the forseeable future.

Convenience

There needs to be more than the speed and extra storage to convince me of the viability of hard disk, and its *convenience* that provides it. A computer system really ought to help the user, and enable him/her to concentrate on the job in hand, rather than having to fiddle about with disks, making sure they are formatted and have sufficient space. In this respect, the hard disk shines. QL users are well-favoured compared to users of some other computers, as the 40 MB capacity provided by the Miracle unit is large for a 'small' computer. You can go on saving and adding files for a long time before a clear-out session becomes necessary, and regular saving of files is much less of a chore.

If asked to instance just one advantage my PC/AT has over the QL without hard disk, my answer would certainly be that Save times are very much shorter. Processor speed advantages may look great in adverts, but that's not what it's all about for most of us; we don't do much that demands fast processing, but we do make plenty of use of files on disk, and that's where time needs to be saved.

That might seem to be that, the overall convenience of hard disk outweighing all other considerations. For a similar outlay, can the Gold Card plus high-density drives offer as much, or more? Not yet having had the chance to see what the speed improvement is in normal use, I have to reserve judgement, to some extent, but I don't doubt that QL computing will be generally much more pleasant at the higher cpu speed. Even Quill should become acceptable nippy (but its bugs won't go away). The normal increase in speed is quoted as anything from x3 to x5. It takes a change of about 25% for an average person to be sure there is a difference, so the improvement with the Gold Card should be clearly evident.

Gold Card

Up to this point, no mention has been made of that other storage area, the ram, but the Gold Card brings with it about 2MB of that, to raise the practical maximum to more than twice that with a Trump Card. Now that is a feature well worth having, for those of us who want several programs loaded concurrently. At present, 896 KB is restrictive for me, both when reviewing programs and when doing other work. You can't have two major programs say, *text87* or *Perfection*, and *Archive* with a large database in use plus a utility such as *Files 2* and a pop-up like *FlashBack SE* loaded, and then load another large application.

Don't tell me nobody tries to do this, because I'm sure it's not only me that wants this many programs available, instantly oncall. To have 2 MB of ram, and have whatever is in it running at several times the speed it does with the Trump Card, is a very attractive prospect. What else? Well, there is a built in battery-backed clock. Not a major thing, but one does get used to having the time there automatically, without having to be set, and I'd much rather have this feature built-into the main expansion board than on a piggy-back board inside the QL. For most of my current purposes, the 1.44 MB drives would suffice; the potential to go to 3.2 *MB* gives some elbow-room beyond that.

With the proviso that we don't yet know how well the Gold Card and high-density drives will operate in long-term service, I think my choice would be to go for that combination, rather than for the hard disk. This conclusion does surprise me somewhat. With the PC/AT, my feeling has always been that hard disk first, and extra ram second, have been the priorities, with a faster processor scarcely being considered, but that was a faster machine to start with than the QL. The QL needs the extra speed, not really because 8-bit, 7.5 MHz wait-state operation is rather slow, but because many of our favourite programs were not written to give optimal speed. Compare *FlashBack* and *Perfection* with the Psion Quartet and you'll see what a difference smart programming can make.

SINCLAIR QL WORLD

THE NEW USER GUIDE

SECTION TWELVE

KEYWORD INDEX

This month in the Keyword Index, Mike Lloyd moves from CLOCK to DATA in SuperBasic, with an exposition on the subject of CURSOR.

CLOCK #chan, string

[SUPER TOOLKIT 2]

ON-SCREEN TIME AND DATE DISPLAY

chan	(optional) a valid screen channel
string	(optional) a format for the time display

CLOCK is a simple multi-tasking utility which prints the QL's internal clock time and date to the screen. If no channel is specified a Qdos channel is opened specifically for the clock: it is only suitable for high-resolution mode. When a SuperBasic channel is opened for the clock a string can be declared which determines the output format. The $ and % signs are used to prefix letters to denote the hour, minute, second, day, month and year. An example command and output are shown below:

```
100 OPEN #3, scr_
110 WINDOW #3, 300, 40, 0, 0
120 CLOCK #3, "Time: %h:%m:%s    Date: $d %d $m %y"

                    Time: 10:17:22   Date: Wed 03 Jan 86
```

CLOSE #chan

CHANNEL MANAGEMENT

chan	A number representing an open SuperBasic channel

When you have finished with a screen window, a channel to the printer or to a file, the CLOSE command will remove all links between the computer and the device. When windows are closed their contents remain on the screen but the screen area is inactive. If the command window (Channel #0) is closed it can only be reopened as a normal window.

CLS #chan, param

SCREEN COMMAND

#chan (optional)	A valid screen channel (default is 1)
param (optional)	0 Clear all the window (default)
	1 Clear above the cursor line
	2 Clear below the cursor line
	3 Clear whole of cursor line
	4 Clear cursor line to right of cursor

The CLS comand is one of the common factors among all Basic dialects; its influence even extends to MS-DOS. CLS normally has no parameters, but in SuperBasic the command is extended to declare firstly which screen window to clear and secondly what part of the window to clear. This greatly adds to its usefulness. For example, suppose you use the fifth line of a window for user input of indeterminate length. To keep the screen tidy, you could issue a CLS 3 or a CLS 4 command before and after each INPUT command. If you want to preserve the first four lines of a window but clear everything else, position the cursor with an AT 4,0 command and then issue a CLS 2.

CODE (string$)

TEXT HANDLING FUNCTION

string$	A valid string or string variable

CODE is the opposite of CHR$; it takes a string or string variable and returns the Ascii code value of the first character in the string. If the string is empty, CODE returns 0. The Ascii code value of non-printable characters (such as TAB, Newline, Bell, etc.) are also correctly returned.

COMMAND_LINE

TURBO TOOLKIT
CONTROL COMMAND

COMMAND_LINE is a command that takes no parameters. Its purpose is to grasp the attention of the command line interpreter – the part of the QL which responds to your instructions as they are typed in. Because the interpreter is always alert to instructions when SuperBasic is running, COMMAND_LINE is only of use in compiled programs.

COMPILED

TURBO TOOLKIT
COMPILER FUNCTION

Several *Turbo* features, such as IMPLICIT, are only available when the program containing them has been compiled. COMPILED is a function which returns 1 if the program currently running is compiled and 0 if the program is being interpreted. It is best used as follows:

```
200 IF COMPILED
210 REMark Turbo-specific commands follow...
220 .....
500 ENDIF
```

CONNECT #chan1 TO #chan2

TURBO TOOLKIT
CHANNEL COMMAND

#chan1	A channel already opened as a pipeline
#chan2	A previously unopened but valid channel number

The CONNECT command is an extremely neat way of pipelining data from one task to another. Anything output from the first channel can be read as input by the second channel command. Before using CONNECT, open a channel as a pipeline with a command such as OPEN #7, pipe_500. The integer following the underscores establishes the length of the pipeline in bytes. Data can now be sent to the pipe using PRINT#7... commands. To read from the pipeline it must be connected to another channel, eg CONNECT #7 TO #5. Now everything PRINTed to channel #7 can be read again as input from channel #5.

If the second channel has a number higher than the highest previously used channel a spurious SuperBasic error is generated. Note also that if the pipe is used to send information from one task to another the CHANNEL_ID function (see separate entry) must be used to identify the correct Qdos channel number.

CONTINUE

CONTROL COMMAND

CONTINUE is a command that has no parameters. Its purpose is to resume the execution of a SuperBasic program on the line following the one being interpreted when processing stopped. Processing can stop due to an error or due to the user pressing the CTRL-SPACE combination (BREAK). SuperBasic also includes a RETRY command which resumes execution by repeating the last command to be executed prior to the interruption.

COPY device1 TO device2	SUPERBASIC
COPY_N device 1 TO device2	SUPERBASIC
COPY_O device1 TO device2	SUPER TOOLKIT II
COPY_H device1 TO device 2	SUPER TOOLKIT II
	FILE/DATA COMMAND
device1	a filename or other data source
device2	a filename or other data destination (optional when used with Super TK2)

The most frequent use of the COPY command is to relocate or rename a file, eg COPY mdv1_filename TO flp 1_filename. However, Qdos treats all data-sending devices almost identically, so the source of the data could be a file, another QL on the network or another computer hooked up via the RS232 ports. Similarly, the destination can be any data-accepting device, such as the screen, a file or the printer. Files are different from other data entities in that they have a header which provides Qdos with information about their type, length, and so on.

If you are sending data to the screen the header information is automatically stripped off, but if you are copying a file to the printer it must be removed explicitly with the COPY_N version of the command.

Normal Qdos error trapping prevents you from using a destination filename which already exists, but the Super TK2 COPY_0 variant will overwrite an existing file of the same name without warning. In the right circumstances it can save a bit of processing, but if it is used thoughtlessly you might find that you have accidentally overwritten a valuable file.

COS (radians)

TRIGONOMETRY FUNCTION

radians	A variable or value representing an angle measured in radians.

Imagine a circle with a radius drawn to some point on its circumference. The input to the COS (cosine) function represents the distance between the top of the circle and the point where the radius touches the circumference. The output from the COS (cosine) function is the vertical distance between the end of the radius and the horizontal axis flowing through the centre of the circle. The direct distance between the end of the radius and the vertical axis is given by the SIN function, and so it is possible to draw a very accurate circle or ellipse using these two functions and the POINT command (which draws dots).

In the example below, the centre of the circle is located at 50,50 and the radius of the circle is 20. Try varying each of the absolute figures to see the effect they have. (Hint: changing only the SIN radius (eg from 20 to 40) will produce an ellipse.)

```
100 FOR dot = 0 TO 2*P1 STEP .02
110     POINT 50 + SIN(dot) * 20, 50 + COS(dot) * 20
120 ENDFOR dot
```

COT (radians)

TRIGONOMETRY FUNCTION

radians	A variable or value representing an angle measured in radians.

The cotangent function returns the ratio of sine to cosine for a given angle, thus COT(1) is a quick way of saying COS(1)/SIN(1).

CSIZE width, height

CHARACTER DISPLAY COMMAND

width	An integer value from 0 to 3
height	An integer value of 0 or 1

The QL was one of the first computers to offer the simultaneous display of different character sizes on-screen, although the options offered are quite limited. QL characters are based on grids 5 pixels wide and 9 pixels high. Allowing a one-pixel separation distance between each character the smallest permitted characters are thus 6 pixels by 10 pixels, achieved by typing CSIZE 0,0 when in the high-resolution monitor mode. Additional space between characters is added by specifying CSIZE 1,0. The characters can be doubled in width by specifying CSIZE 2,0 or CSIZE 3,0 and doubled in height by changing the second parameter to 1. Much more control over character size and font design is obtained by using *Super TK2* (or screen utilities like *Lightning*).

CURDIS #chan
CURSEN #chan
CURSOR_OFF#chan
CURSOR_ON #chan

SUPER TOOLKIT II
SUPER TOOLKIT II
TURBO TOOLKIT
TURBO TOOLKIT
SCREEN MANAGEMENT COMMAND

#chan	(optional) A valid screen channel – default #1

CURSEN, CURDIS CURSOR_ON and CURSOR_OFF take no parameters other than a valid screen channel; their role is to display and hide cursors. The QL's normal behaviour is to display only one cursor, usually a flashing character-sized block, no matter how many windows are open at the time. However, Qdos actually has a cursor for every open window, although they tend to be invisible when they are inactive. CURSEN and CURSOR_ON allow you to see the inactive cursors as solid, unblinking blocks. CURDIS and CURSOR_OFF hide them and make them inactive.

WARNING: It should be obvious that disabling the command window cursor (eg CURSOR_OFF #0) will disable SuperBasic itself because no further command input can be obtained. If you do this accidentally there is no cure except to reboot the computer, losing anything which has not previously been saved.

CURSOR #chan x1, y1, x2, y2

#chan	(optional) A valid screen channel – default #1
x1	(optional) Horizontal graphics co-ordinate
y1	(optional) Vertical graphics co-ordinate
x2	Horizontal pixel co-ordinate
y2	Vertical pixel co-ordinate

The CURSOR command's behaviour can seem so mysterious that it is often ignored by programmers, which is a shame because it is extemely useful. It allows you to place text with pixel-point accuracy anywhere within a window. The command's parameters refer to the location of the top left pixel of the next character to be printed. Moreover, by using both the graphics and the pixel co-ordinate systems the CURSOR command lets you mix text and graphics freely in the same window.

In the default window, with no channel parameter, the CURSOR command works correctly on most QLs. If it is followed by one pair of co-ordinates SuperBasic treats them as references to the pixel-based co-ordinate system. If a second pair of co-ordinates are included the first pair are assumed to relate to an absolute position on the graphics grid and the second pair become an offset from the first location using the pixel grid.

If this seems complicated, think of the reasons why SuperBasic author Jan Jones came up with the command. Graphics can be drawn to any scale with an origin anywhere in the window by using the SCALE command. It is often the case that a graphics image such as a pie chart or technical drawing needs to be annotated with text. SuperBasic needed a way of making it easy to add text to graphics easily and flexibly, particularly so that a SCALE command would not dislocate picture and words. CURSOR allows you to select the exact point on a drawing against which some text is to be placed and then allows you to specify, in pixels, the offset from that point where the text should begin.

CURSOR begins to act up when a window reference is made, because a bug in many versions of the QL's rom permits only two or four parameters to be passed to the command. CURSOR #2, 40, 32, 36, 10 should be valid, but it has five parameters and is rejected by the interpreter. The answer is to omit the last parameter – the pixel-based vertical offset – and all is well. Minerva owners can disregard this workaround because the Minerva rom correctly looks for an even number of parameters disregarding any channel number.

As a quick demonstration of CURSOR's power, here is a routine to draw a clock face. Refer to COS() for an explanation of how the co-ordinates were calculated. Note that each number is printed x/12ths of the way around the circumference of a circle which measures 2*pi graphics units.

```
100 CLS
110 FOR X = 1 TO 12
120     LOCN = X/12 * 2*P1
130     CURSOR 50 + 30 * SIN(LOCN), 50 + 40 * COS(LOCN), 0,0
140      PRINT X
150 ENDFOR X
```

DATA a, b, c, d, ...

DATA INDICATOR

a,b,c, etc — An arbitrary list of variables or values

The DATA command is used in conjunction with READ and RESTORE as a means of obtaining input from within a program itself. A full explanation of these commands is provided under the entry for READ. DATA simply flags to the interpreter that what follows is a set of data values rather than an executable statement. There are no restrictions on separating DATA and other statements with colons on a single line, but it is more usual to keep DATA apart from executable program statements.

DATAD$
DATA_USE directory

SUPER TOOLKIT II
SUPER TOOLKIT
DIRECTORY FUNCTION AND COMMAND

directory — A filename prefix ending in an underscore.

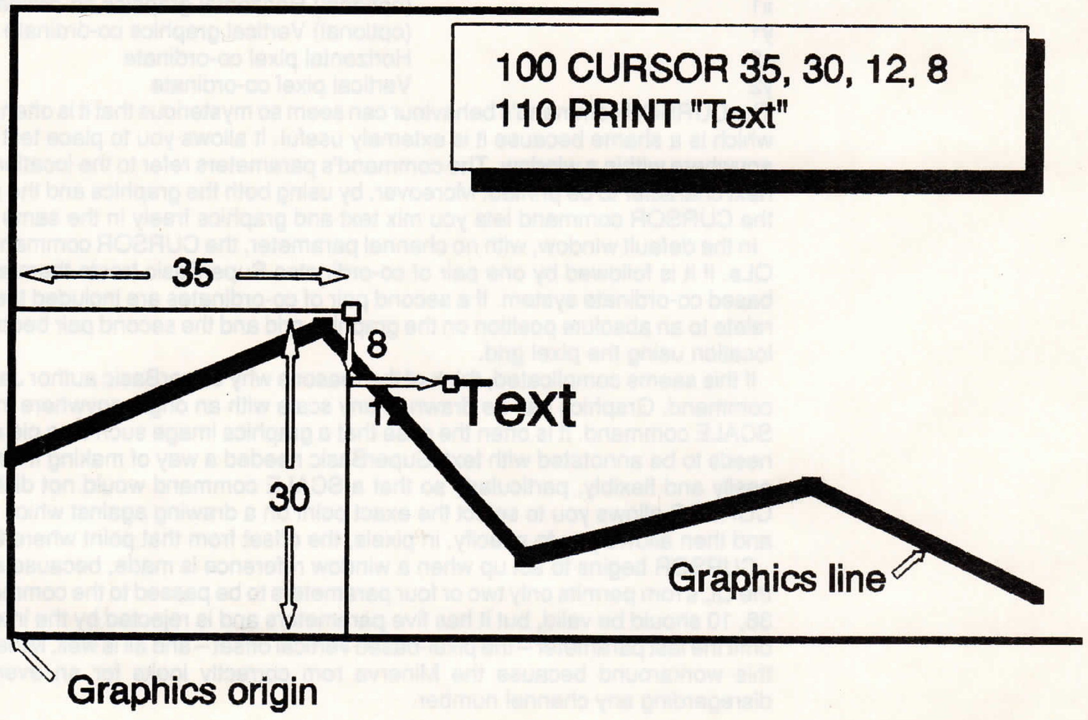
Super Toolkit II belatedly introduced the concept of directories, a means of grouping files together on a disk or microdrive. The idea is that a floppy disk typically contains scores of files which normally exist in one directory. Life would be easier if a bunch of related files could share the same prefix, but nobody likes the thought of having to type long filenames all the time.

Super Toolkit II solves this dilemma by automatically coping with filename prefixes, for instance by prefixing the names of all working files with "flp1_work_". Other data files could be prefixed with "flp2_temp_" or "mdv2_letters_" or whatever seems useful at the time. The prefix is set using the DATA_USE command, which is followed by a text string enclosed in quotes. The string should include the device, eg DATA_USE "flp 1_temp_". If the final underscore is omitted, Super Toolkit II adds it. The data prefix can temporarily be overridden simply by giving the full pathname, eg OPEN#5, "flp1_junk_data". Incidentally, the quotes around filenames are not obligatory but they are a good idea.

With DATA_USE in effect, a file created with the command OPEN_NEW#3, "filename" will actually be called "flp1_temp_filename", assuming that "flp1_temp_" is the current data prefix. The prefix currently in effect is revealed by the command PRINT DATAD$. Note that DATA_USE and DATAD$ are only used by data-related files: files containing SuperBasic and other executable programs have their own prefix.

Note that DATA_USE affects many file-related commands, such as DIR and DELETE.

Further explanations of directories and the Super Toolkit commands which implement them are provided under the individual commands, such as DDOWN, DUP, PROG_USE, and DEST_USE.



```
100 CURSOR 35, 30, 12, 8
110 PRINT "Text"
```

JUST FOR STARTERS

In Don smith presents a set of 'DIY instructions' for programming in SuperBasic. Using a database as an exmple, he builds up a set of routines.

There must be many new QL owners who would enjoy writing their own programs, tailored to their own specific needs, if they knew where to start. I must confess that I have never bought any software, haven't a clue about compilers, assemblers, etc. and don't use boot files or the Psion quartet. Despite this, the QL runs my business, prints invoices, order forms, stocktaking, VAT returns etc. and operates dozens of different databases for my hobbies.

This article serves the same purpose as do the instructions with a DIY car kit. When completed, the menu will enable a number of simple operations to be carried out. Before commencing a program, the programmer must be able to define the basic requirement and then enumerate the sections that will fulfil that requirement.

Database

I have chosen a database as a useful example simply because it contains procedures that can

```
100  open#1,scr_509x200a2x1: ink#1,7: paper#1,2: ink#0,7
110  open#2,scr_509x200a2x1: ink#2,2: paper#2,7:
     open#4,ser1: baud 9600
```

Listing one

```
120  menu
130  :
140  defproc menu
150  cls: print\\\to 36;'MENU'\\\to 10;'A'to 15;'Load data'
     to 50;'B'to 55;'Save data'
160  print\to 10;'C'to 15;'Enter data'to 50;'D'to 55;'Amend'
170  print\to 10;'E'to 15;'To screen'to 50;'F'to 55;'To printer'
180  print\to 10;'G'to 15;'Print program'to 50;'H'to 55;
     'Save program'
190  print\to 10;'I'to 15;'Stop'
200  i=code(inkey#(-1)): cls
210  sel on i
220   =97: lod: =98: sav
230   =99: records: =100: alter
240   =101: scren: =102: prin
250   =103: listprog: =104: newprog
260   =105: stop
270  end sel
280  enddef
```

Listing two

```
290  :
300  rem------LOAD DATA----------
310  defproc lod
320  cls#0: at 3,10: print'How many entries this sitting?':
     input#0,'Number'to 10;max
330  at 3,60: print max
340  at 5,10: print'Are there any previous entries? (1/0)':
     prev=inkey$(-1)
350  at 5,60: if prev: print'Add to file': else: print'New
     file'
360  at 7,10: print'Name of file': cls#0: input#0;'Name'to 10;
     file$
370  at 7,60: print file$
380  at 12,25: print'Is this correct? (1/0)': correct=inkey$(-1)
390  cls#0: if not correct: cls: lod
400  if prev then
410   open_in#3,"mdv1_"&file$
420   input#3,previous: max=max+previous
430   dim item$(max,30),num(max)
440   for x=1 to previous: input#3,item$(x),num(x)
450   close#3: num(0)=previous
460  else
470   dim item$(max,30),num(max)
480  endif
490  menu
500  enddef
```

Listing three

be used in other operations. The basic requirement of a database is to manipulate data, which might consist of books, addresses, workshop tools or music tapes. In order to keep this introductory article as simple as possible, a bare minimum of processes are considered: 1. allocate ram space (load/re-load), 2. enter data 3. store data 4. print to screen 5. save program. Sorting, abstracting and printing the program or data are dealt with later.

First, the screen format is adjusted for ease of use. The default setting when switched on splits the screen in two, with #(channel)2 on the left and #1 on the right. I use a full screen for both channels, never finding the need for special windows. So the program commences with the lines in **Listing one.**

Colours

With a green monochrome monitor, this makes program lines green on a black background and program output the reverse. The ink#0 in line 100 makes editing brighter. If your printer has a parallel interface, alter ser1 to par. If you have a colour monitor, choose your preferred colours. To enter these two lines proceed as follows:

Type auto and press ENTER. A 100 appears below, so don't type the line number, but type everything else exactly as shown – no capitals. Press ENTER and see what appears. Clever QL! From this point on, always press ENTER after an instruction to type something in inverted commas. 110 will now have appeared below, so type the second line and ENTER. Now follow these steps:

(1) Hold down CTRL and press the SPACE bar. You have now left auto. (2) Type 'run' (but not the inverted commas). (3) Type 'cls' (4) Type 'list'. Your two lines now reappear. Don't worry about part of the screen staying dark, it will soon clear. If it offends, type 'cls#2' then type 'list' again.

From now on, all subsections will be self contained PROCedures operated by a menu which is itself a PROCedure. This is listed below but is not operational until you have written the remain-

```
510  :
520  rem-----ENTER DATA----------
530  defproc records
540  if prev: x=num(0): seeline
550  for x=num(0)+1 to max
560   cls#0: print#0,'Name(just ENTER to exit)'\fill$('-',30):
      input#0,nam$
570   if nam$='': exit x
580   item$(x)=nam$
590   input#0,'Number'to 10;num(x): seeline
600   num(0)=x
610  endfor x
620  menu
630  enddef
640  :
650  rem-----SEE A LINE----------
660  defproc seeline
670  print x,num(x),item$(x)
680  enddef
```

Listing four

```
690  :
700  rem------SAVE DATA----------
710  defproc sav
720  delete "mdv1_"&file$
730  open_new#3,"mdv1_"&file$: print#3,num(0)
740  for x=1 to num(0): print#3,item$(x)\num(x)
750  close#3
760  delete "mdv2_"&file$
770  copy "mdv1_"&file$ to "mdv2_"&file$
780  menu
790  enddef
```

Listing five

```
800  :
810  rem------SCREEN SHOW----------
820  defproc scren
830  for x=1 to num(0)
840   seeline
850   if x/20=int(x/20) then
860    cls#0: print#0,'Press any key to continue or ESC - back
       to menu': c=code(inkey$(-1))
870    if c=27: menu: else: cls
880    endif
890  endfor x
900  cls#0: print#0,to 30;'Press any key for MENU': pause
910  menu
920  enddef
930  :
940  rem------SAVE PROGRAM-----------
950  defproc newprog
960  delete mdv1_database
970  save mdv1_database
980  delete mdv2_database
990  copy mdv1_database to mdv2_database
1000  menu
1010  enddef
```

Listing six

ing PROCedures. To continue writing the program, type 'auto 120' and copy **Listing two** in the same way.

Ascii

Each character on the keyboard has an Ascii code number (see the *User guide,* Concepts, p.59), a=97, b=98 etc. Though capitals are printed in the menu, this is just a guide to which key to press, which will always be lower case. In line 200, inkey$ waits for ever until a key press. Code reads the code number and transfers it to the variable i. This could just as easily have been q, z or bottle! SELect checks the value of i and chooses the correct PROCedure. From now on all PROCedures will be separated by a colon and headed with a REM statement.

The QL keeps a tally of all PROCedure names and accepts the name as a user command to execute that PROCedure. You could type it from the keyboard with the same effect. Line 260 however, incorporates an exclusive QL rom command which means 'stop everything and take a coffee break'. The backslash (\) forces a new line, so three of them force three blank lines. The next PROCedure combines both loading and ram space allocation and will inform the QL of your requirements – how many entries, whether commencing for the first time or additional ones and the file name. When starting, make up a name such as address, tool or card, etc. You can always check the filename if your list is mislaid by typing 'dir mdv1_'. Now type in **Listing three.**

Yes/no

When the program is finished, but not yet, wherever 1/0 appears, press either 1 for yes or 0 for no. In line 360 you type the name and press ENTER because it is an INPUT. Lines 340 and 380 require a single keypress only, because that is all that inkey$ is waiting for. Notice how, in line 380, by choosing a suitable variable 'correct' instead of 'x', the next line reads understandably; otherwise one would have had to type 'if not a' or 'if not x', which means little to a beginner.

The computer interprets the situation as follows: if a variable has a value of 0 (nought) it is considered FALSE while values greater than zero are considered TRUE. Line 390 is therefore really stating 'if not TRUE' (ie if not having a value greater than zero) then clear screen and start again. By pressing 1, the variable 'correct' has a TRUE value and so the final two commands at the end of the line are ignored and processing continues to line 400. Here again, is 'prev' true or false? If false, then line 460 onwards gives alternative commands. Dim stands for DIMension, the size of the data space required.

Now type in **Listing four.**

Previous

If there were some previous entries, 'prev' will be true (from line 340), so at 540 your last entry can be shown because num (0) held your total previous entries – line 600. By encapsulating line 670 in a separate PROCedure, it only needs typing once, even though it may be used a number of times. If this is a new entry, num(0) will equal zero automatically and since zero + 1 = 1, x will commence with a value of one. With 35 previous entries, x would commence at data number 36. At 560, fill$ prints out 30 dashes to make sure that your entry doesn't exceed the size of the array you have set up. Now type in **Listing five.**

When this program is completed, but not until then, you will always commence by loading (A in the menu), whether there is anything to load or not, because this PROC is also used to dimension the arrays for data storage. You will therefore have already given file$ a name and therefore lines 720 to 770 have all the information the QL needs. Lines 720 and 760 are there because the QL is unconcerned whether there is anything to delete or not, so just one PROCedure is required whether this is a new start or overwriting a previous block of data. It covers both options. The backslash in 740 is necessary to separate the two data items on the cartridge. The QL automaticaley adds a backslash of its own after the num(x) before the loop returns with the next values of x.

Errors?

If you are a keen student of the manual, you may think that there are some grammatical errors here and there. In line 740, the loop appears unfinished, with no 'end for x'. This does work, providing that nothing else is added to the line. Again, when reading the manual, it does appear that all commands and even mathematical signs are separated from each other by a space and, in most instances, this is essential. Since I will never proceed to disks, I tend to be mean over cartridge space. In line 590 for example, the command 'TO' follows immediately after the inverted commas. Evidently the QL does not consider the (') to be a specific character but merely an identifier of characters which need printing to the screen and so it also acts as a space. However, there must be a space after the 'TO'. Also, despite a tendency to contract program lines, I always leave a space after every colon which separates one command from another, because it makes the line more readable.

Two short PROCedures complete this initial article in **Listing six.** Printer printout is omitted at this stage because of the need for a number of other print PROCedures.

Scrolling

Note how, at 840, we have used PROC 'seeline for the third time. Lines 850 to 870 stops the screen scrolling after 20 items have been printed and waits for further instructions. Line 900 is essential, otherwise when the x loop was completed the program would zip straight back to the menu. Finally, at 940, you need to be able to save the program. You are now ready to try it out. This is a very basic program with no provision as yet for amending data, abstracting or sorting. If it doesn't work, or you get error statements on the screen, you have mistyped or omitted something – the QL is never at fault. If this happens, hold CTRL and press SPACE-bar. Type 'list' and ENTER then quickly hold CTRL and press F5. This stops the screen scrolling. Pressing F5 alone restarts it. You can now work through the listing while checking the lines for errors. An error statement at the bottom of the screen even tells you (usually) which line to check. There are some excellent and clear instructions in the QL User Guide for correcting program lines. Read the Introduction, p.5 (Using the QL) and Beginner's Guide, p.9 (Editing a Program). You can easily get back to the menu anytime by ENTERing the word 'menu'. The word 'database' in lines 960-990 should be altered to something of your choice.

Suggestion

Suggestions for a try-out are as follows:

1. Format two cartridges, each at least three times if new. (See the manual.)
2. Go to menu, press A and answer the questions, having decided on a name for your data.
3. Back at the menu, press C. Type in about two dozen names and numbers then exit as per screen instructions.
4. Having automatically returned to the menu, press B.
5. Press E to see how the data looks on the screen.
6. Press H to save the program.
7. Either switch off the computer *after* removing the cartridges temporarily, or press the reset button at the side.
8. Type '1run mdv1_your name' and ENTER. run both loads and starts running in one go.
9. Press A and you should be back in business once more, ready for further data. Press E to check that all previous data is back in. It is no good pressing D, F or G since these PROCedures are not yet written, but you could try, just to see what happens.

The next article increases the number of fields and adds additional PROCedures. It therefore also includes a number of short cuts for altering or enlarging a program and indeed, once hooked on programming you will be forever tinkering and smoothing out and streamlining your progs.

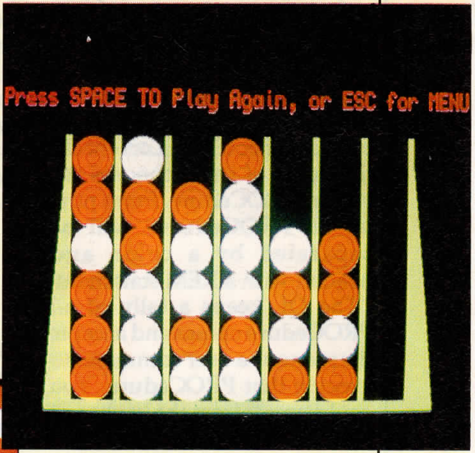
SOFTWARE FILE

FIVE GAME PACK

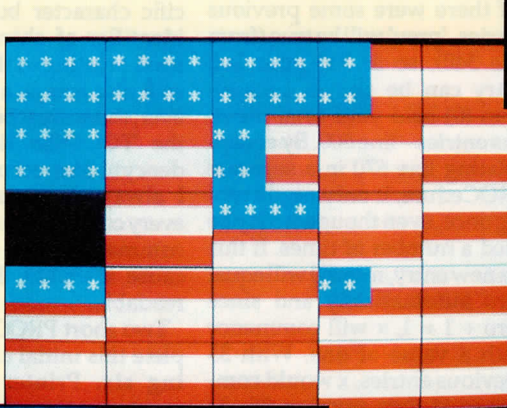
INFORMATION:
Program: *Five Game Pack.*
Supplier: C.G.H. Services
Cwm Gwen
Hall, Pencader
Dyfed
SA39 9HA.
Price: £12.50 disk only. (p&p inc.).
Expansion not required.
Compatible with Minerva and GoldCard.

John Shaw got his hands on a set of classic games, and we haven't been able to prise him away yet.

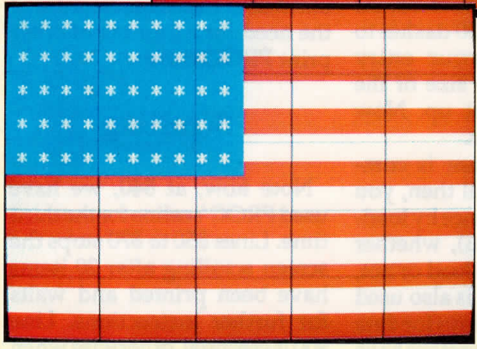
Wreford Davies is to be commended for programming this pack in such a professional way. The five games, all of which are well known, have been revitalised by first class graphics and speed of execution.



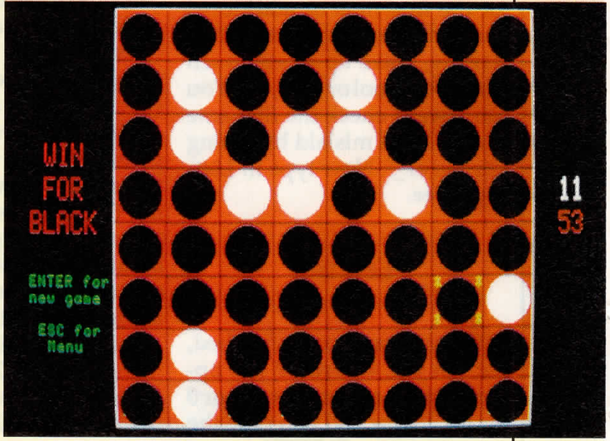
Quadliner, or four-in-a-row



A scrambled Old Glory. You also get the Union jack and the Rising Sun...



. . . Old Glory in all its glory



Othello, or Reversi – a game of cunning. . .

Roulette

In Roulette, the roulette wheel and betting table are set out on the screen, side by side and are visually excellent. For those of you not familiar with the intricacies of this famous casino game, Wreford has provided graphical instructions on the press of <F1>. It demonstrates the different ways in which bets may be placed and the meanings of such words as Manque, Passe, Pair and Impair. In addition the odds for each type of bet are shown.

To place a bet, your coin is placed in position by moving the cursor and then pressing any number from 1 to 0 (0 bets 10).

Your initial allowance is £100.00. Once satisfied with your system you press <F5>. This sets off a clever graphical display where the ball spins around the wheel in an ever decreasing spiral, generating a sound quite close to the real thing.

When the moment of truth occurs and the wheel stops spinning, the computer either credits or debits your account according to how you fared.

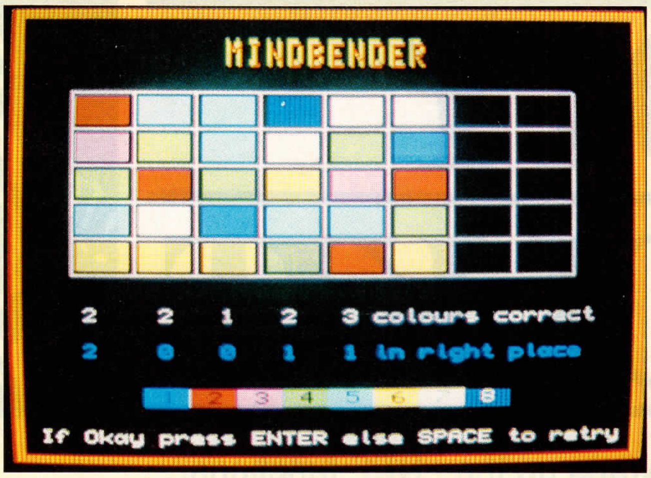
Results

A third screen may be called up which gives all the statistics of your endeavours. The sequence of results and the number of Spins, Highs, Lows, Evens, Odds, Blacks and Reds are all made available to you to increase your chances of making a fortune.

Try the 'Double if you lose – original stake if you win' system. See whether it is worth going to a real Casino!

Quadliner

This is an established party favourite, better known as 'Four in a Row'. In this smooth-run-



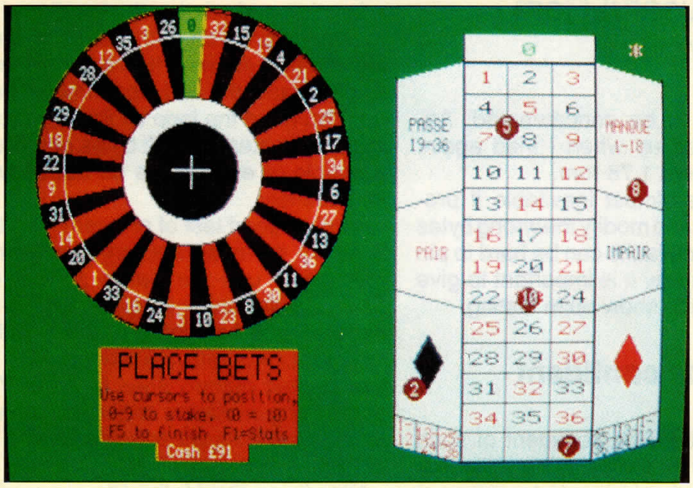
MINDBENDER

2 2 1 2 3 colours correct

2 0 0 1 1 in right place

If Okay press ENTER else SPACE to retry

Mindbender, a logical challenge of colours.

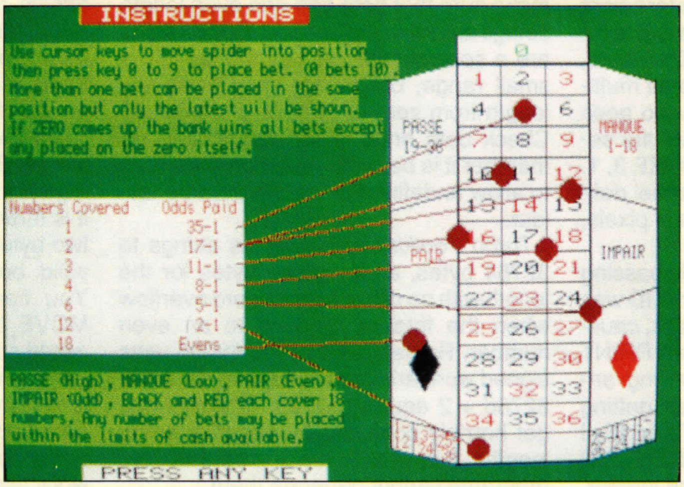


PLACE BETS

Use cursors to position, 0-9 to stake. (0 = 10) F5 to finish F1=Stats

Cash £91

In Roulette, you can pit your luck against the wheel. . .



INSTRUCTIONS

Use cursor keys to move spider into position then press key 0 to 9 to place bet. (0 bets 10). More than one bet can be placed in the same position but only the latest will be shown. If ZERO comes up the bank wins all bets except any placed on the zero itself.

Numbers Covered	Odds Paid
1	35-1
2	17-1
3	11-1
4	8-1
6	5-1
12	2-1
18	Evens

PASSE (High), MANQUE (Low), PAIR (Even), IMPAIR (Odd), BLACK and RED each cover 18 numbers. Any number of bets may be placed within the limits of cash available.

PRESS ANY KEY

. . . and see how you fare!

ning version, you drop discs into slots by moving the cursor keys and using the space bar to drop them. The object of the game is to get four discs in a row, either horizontally, vertically or diagonally, taking turns with the QL to drop them. A score is kept and displayed at the end of each game. This is a good game against a very challenging opponent.

Mindbender

This is the QL version of a puzzle which was produced under various names, the most notable being Mastermind. For those readers who have never played the game, the essence of the challenge is as follows:

The QL prompts you to choose a number of colours (between 3 and 8). The display then shows a grid, five slots deep and eight slots wide. Your task is to establish the undisclosed sequence of five colours chosen by the program and the sequence in which they are stored. You have eight tries and the slots fill up with your chosen colours. You are given hints as to your progress and to how well you are doing, e.g. 'four colours correct – 2 in the right place'.

This is an absorbing game for those who enjoy logical mind challenges.

Othello

Othello, or Reversi as it is sometimes known, is a board game simulation played against the QL.

The object is to place discs on the board in such a position as to trap your opponent's discs between them. This then automatically changes them to your colour. Your opponent is, of course, trying to do the same and as the game progresses the discs change back and forth reflecting the skills of each player. The waiting time is very low and the QL a very clever player. A full set of instructions is provided on-line.

A tip for those of you who are prepared to take on the QL: try and get your colour counters in the corners and on the sides; if you don't then the QL certainly will!

Flag Puzzle

Here is a challenging puzzle if there ever was one. The opening screen gives you a choice between three flags, the Union Jack, Old Glory (Stars and Stripes) and the Rising Sun (Japanese). You are then asked for a choice of difficulty (0 to 9).

The screen then clears and momentarily your chosen flag is shown complete. It is then broken up into squares which are then shuffled around according to the level of difficulty you chose.

Your job is to unscramble them using the cursor keys which move the squares around. The 0 level is comparatively simple, involving some twelve or so moves. Level 9 is mind boggling. Squares dash around the screen at a frantic rate.

Fortunately Wreford has provided a Help facility which, if taken up, points an arrow in the direction in which the square should go. If you get totally lost then there is an excellent command which will totally retrace all your steps and indeed all those needed to finish the game.

This Games Pack is very professionally produced and will easily stand comparison with any other commercially written program. I have thoroughly enjoyed playing each and every game and I feel most readers will as well.

DIY TOOLKIT

Simon Goodwin wraps up the _DEF functions, explores QL bugs, and uncovers useful Rom vectors never revealed by Sinclair.

Last month DIY Toolkit introduced three new SuperBasic functions, _DEF%, _DEF$ and _NAME$. This column presents the remainder of the assembly code listing, and discusses the developments added to the prototype sent in by reader Anders Hartzellius. I also reveal eleven previously-undocumented rom routines that are easily called from Basic or code.

Anders uses Minerva 1.79 which corrects many faults of Sinclair roms. I have shuffled the registers used to suit older roms which clobber registers D4 and D6 when fetching parameter values. Alas, this complicates the listing.

Anders' code assumed a character height of ten pixels when scrolling. This is fine for most QL setups but upsets NTSC tv users, who expect characters to be some multiple of eight pixels in height. It also goes wrong if double-height characters are selected, such as CSIZE 0, 1 or CSIZE 3, 1. The revised code reads the channel definition's YINC, the height of lines in pixels, and scrolls accordingly.

The prototype made two slips in passing and returning values. The channel ID was not removed from the Maths Stack, causing 'creep', as I discovered when PRINT 100 + _DEF%(#2) gave the wrong answer. The #2 was left on the stack, getting in the way of the first number. To fix this I added the instruction ADDQ.L #2,BV_RIP (A6) after the code that reads the integer.

The most subtle slip was in _DEF$, and _NAME$ too, at first. The code to move text from the Name List to the Maths Stack copied one byte too many. This is an easy mistake to make, and tricky to spot when you have to contend with DBRA, odd and even boundaries and byte or word length prefixes, which may or may not be included in the total.

My attention was grabbed when PRINT _DEF$ seemed to work fine, but PRINT "QL" & _DEF$ crashed the JM rom. The extra byte stacked by _DEF$ clobbered the length of the first string above it on the stack, waiting to be concatenated. Another QL locked up when I tried again, using the Minerva 1.76 rom.

Problems are almost inevitable if programs address and modify the wrong bytes of memory. Such faults can be hard to pin down, which is why it is important to give them a chance to show up in testing.

Code confident

One way to become confident of such code is to step through it using a monitor such as Qmon, checking register values at every stage of a real example. It is wise to make sure that both odd and even length strings are correctly handled. Basic names are a special case, as lengths fall into a small range, but strings may have any length from zero to 32K, or thereabouts. Check both ends of the range, and remember it is better to report an error than to allow undefined or inconsistent behaviour.

Turbo Toolkit explicitly limits strings to 32764 bytes, allowing two bytes for the length with no risk of signed word overflow when the total is rounded to an even number. Remember that the 68008 uses signed word arithmetic and considers that 32766 + 2 equals -32768. Sinclair roms get horribly confused by negative string lengths; it is wise to make sure that your programs cannot generate them.

I explain this not to embarrass Anders, whose suggestions were the best documented that I have received from readers in many months, but to show how the same bugs crop up time and again in SuperBasic extensions. I catalogued these in my coverage of 'deadly sins' for *QL World* in September and August 1988, but I keep finding and making and fixing familiar mistakes.

It is important to test functions by calling them from the middle of expressions, as well as printing the result directly. Some problems appear if you run a loop, calling a function and printing the result and free memory alongside. If the stack is 'creeping' the free memory figure shrinks every so often, as more and more rubbish builds up. You need the loop as Basic allocates memory in steps of 512 bytes or more, so a couple of extra bytes do not show up immediately.

Another good test of programming tools is to try them with a large program in memory. I use the Turbo parser, over 130K of Basic tokens with about 150 procedures and functions, and a trivial but enormous program made with a FOR loop that printed dummy lines to a file. Word addressing problems only show up when locations are displaced by more than 32K of ram.

After fixing the Maths Stack I modified _DEF% and _DEF$ to store Name Table offsets relative to the start of the table, rather than the start of Basic. The program file and other areas come before the Name Table, so simple word offsets are inadequate if a large program is loaded.

The revised code subtracts the value of BV_NTBAS from each offset before storing it in the table. As each word is retrieved it is moved to a data register with the top two bytes clear, so offsets up to 64K are valid, before BV_NTBAS is added back. You can see this in the **Listing,** at MOVE_LOOP. MOVEA.W 0 (A3,A6.L),A5 would not work for words over 32767.

It might be simpler to store and recall long words, but that would only accommodate half as many in a given amount of buffer space. The lower three bits of each word are redundant, as entries are eight bytes long, but it is not worth shifting them out and in to boost the capacity to 65535 names as the Name Table can never get that big. I could have chosen to store offsets in the Name List, rather than point at the Name Table entries. It makes little practical difference.

Anders stored the words by pushing them onto the A7 stack, with a limit of 51 entries to prevent stack overflow. The prototype restored the original value of A7 when finished. This worked fine, as long as SuperBasic did not move, but that was not a valid assumption.

```
* QL World DIY TOOLKIT _DEF%, _DEF$, _NAME$ functions, version 1.5
* (c) Anders Hartzelius 1991, DIY changes (C) 1991 Simon N Goodwin
* PART 2 - CONTINUED FROM LISTING 1 of QL World January 1992
*
io_fbyte        equ     $Ø1                     Trap key to fetch a byte
io_sstrg        equ     $Ø7                     Trap key to send a string
sd_pos          equ     $1Ø                     Set row & column position
sd_tab          equ     $11                     Tab cursor to a column
sd_prow         equ     $15                     Previous row
sd_nrow         equ     $16                     Next row
sd_scrol        equ     $18                     Scroll window
sd_setst        equ     $28                     Set STRIP colour
sd_setin        equ     $29                     Set INK colour
bv_ntbas        equ     $18                     Offset to Name Table
bv_rip          equ     $58                     Limit of RI Stack
bv_chrix        equ     $11a                    Vector to check RI space
err_nc          equ     -1                      NOT COMPLETE error code
err_or          equ     -4                      OUT OF RANGE
enter           equ     $Øa                     Code for ENTER key
space           equ     32                      Code for SPACE bar
esc             equ     $1b                     ESC
up              equ     $dØ                     Up arrow
down            equ     $d8                     Down arrow
*
* Highlight the first name and allow selection from the menu
*
choose          moveq   #Ø,d1                   Column
                moveq   #Ø,d2                   Row
                moveq   #sd_pos,dØ
                bsr     trap3                   AT #X,Ø,Ø
                swap    d4                      Retrieve char. height.W
move_loop       moveq   #Ø,d1                   Prepare for unsigned maths
                move.w  Ø(a3,a6.l),d1           D1:= index in name table
                add.l   bv_ntbas(a6),d1         Relate to start of Table
                movea.l d1,a5                   A5 is offset inside BASIC
                bsr     invert                  Inverted video
                bsr.s   print_name
                bsr     invert                  Revert
key_loop        moveq   #io_fbyte,dØ
                bsr     trap3                   Get keypress
                cmp.b   #esc,d1                 ESC ?
                bne.s   test_enter
                moveq   #err_nc,dØ              Not complete
                bra     return                  Return to basic
test_enter      cmp.b   #enter,d1               ENTER ?
                beq     done
                cmp.b   #space,d1               SPACE ?
                beq     done
                cmp.b   #up,d1                  UP ?
                bne.s   test_down
                cmpa.l  (a6),a3                 First entry ?
                beq.s   key_loop                Yes: ignore
                subq.l  #2,a3                   No: previous entry
                moveq   #sd_prow,d6             Cursor up
                bra.s   arrow                   Move cursor
test_down       cmp.b   #down,d1                DOWN ?
                bne.s   key_loop
                moveq   #sd_nrow,d6             Cursor down
                addq.l  #2,a3                   Next entry
                cmpa.l  a2,a3                   Past the last entry ?
                bne.s   arrow                   No: move cursor
                subq.l  #2,a3                   Yes: stay at the end
                bra.s   key_loop                Wait for keypress
*
arrow           moveq   #Ø,d1                   First pos on line
                moveq   #sd_tab,dØ              Move cursor
                bsr.s   trap3                   to start of line.
                bsr.s   print_name
                move.l  d6,dØ                   Move cursor up / down
                bsr.s   trap3
                beq.s   no_scroll               Out of range ?
                moveq   #sd_scrol,dØ            Yes: scroll window
                move.w  d4,d1                   Scroll one line
                cmp.w   #sd_prow,d6             Which direction ?
                beq.s   scroll                  Scroll down
                neg.w   d1                      Scroll up
scroll          bsr.s   trap3
no_scroll       moveq   #Ø,d1                   First pos on line
                moveq   #sd_tab,dØ              Move cursor
                bsr.s   trap3
                bra.s   move_loop               Wait for keypress
*
print_name      moveq   #Ø,dØ                   Prepare for unsigned sums
                move.w  2(a5,a6.l),dØ           DØ -> entry in name list
                add.l   a4,dØ                   Add NL base address
                movea.l dØ,a1
                moveq   #Ø,d2                   Clear high bytes of D2
                move.b  Ø(a1,a6.l),d2           D2:= name length
                cmp.w   d7,d2                   Name too long ?
                bmi.s   len_ok
                move.w  d7,d2                   Truncate name to fit window
len_ok          addq.l  #1,a1                   A1 -> name
                moveq   #io_sstrg,dØ            Print string
                trap    #4                      Signal A6 relative
*
trap3           trap    #3                      IO trap
                tst.l   dØ                      Error ?
                beq.s   return                  No: return to caller
```

The QL may re-allocate transient program memory at any time as it runs other tasks. QDOS moves all the SuperBasic memory contents and tweaks SuperBasic's A6 and A7 registers so they still point to the right part of memory. It is never safe to save and restore the value of USP A7, because the restored value will be wrong if SuperBasic moved after A7 was saved.

I decided to fix this and remove the limit on the number of names in the menu, storing the index in SuperBasic's Buffer area. This always contains room for at least 128 bytes, or 64 name indices, and grows if long program lines are loaded or entered. There is an easier way to expand the buffer, but it has never been documented – until now. . .

Code vectors

Machine code routines inside Qdos are accessible in two ways. Most can be invoked by the use of TRAP instructions, which I documented in July's QL World and DIY Toolkit Volume T. Many other routines are accessible to Basic CALL or machine code JSR instructions, via 'vector' addresses stored in a table early in every QL rom.

The table of vectors always starts at a fixed address, but the values therein vary between rom versions, as Sinclair and others have shuffled code around. As long as you read the vectors you can call any routine without worrying about its address in a particular rom.

Each entry is a word – the address of a rom routine. The actions performed, and the parameters needed, make it most useful to call vectors from machine code, but you can reach them from SuperBasic. You could reserve NUM bytes of space for the interpreter's maths stack with CALL PEEK_W(282),NUM. The vector address for BV.CHRIX is 282, and the parameter NUM is passed to the code in register D1.

Most of the vectored routines are in the first 32K of the QL rom, but you must add 16384 to the value from the table when calling microdrive routines, as they are located in the second (16K) QL rom chip. It cannot be addressed directly by a signed word, which can only reach address 32767.

By default the 68008 considers values over 32767 to refer to negative addresses, ranging from -32K to -1, but unsigned values from 0-65535 are more useful if you wish to refer to a table with a fixed start.

The code for these DIY extension functions is often complicated by the need to distinguish between signed and unsigned words. Many bugs in SuperBasic and the Psion packages have stemmed from the unplanned use of signed arithmetic.

A case in point is the SuperBasic Name Table, which contains word offsets that point at text in the Name List. The maximum capacity of the list is doubled, from 32K to 64K, if unsigned offsets are used, but extra code is needed to stop the QL treating the extra values as negative numbers, prob-

```
                cmp.w    #err_or,dØ               Out of range ?
                beq.s    return                   Yes: return to caller
quit_out        addq.l   #4,a7                    Return to earlier call
return          tst.l    dØ
                rts
*
invert          move.l   d5,d1                    D1:=previous ink / strip
                swap     d5                       Ink:=strip , strip:=ink
                moveq    #sd_setst,dØ
                bsr.s    trap3                    Set strip to ink
                swap     d1
                moveq    #sd_setin,dØ             Set ink to strip
                bra.s    trap3                    Call IO trap
*
done            clr.w    d7
                swap     d7                       Retrieve the % / $ flag
stacker         movea.l bv_rip(a6),a1             A1 -> the RI Maths Stack
                moveq    #2,d1                    At least 2 bytes needed
                cmp.b    #3,d7                    Return % or $  ?
                beq.s    alloc                    Jump if %
                moveq    #Ø,dØ                    Clear high bytes
                move.w   2(a5,a6.l),dØ            Pick up offset, Ø..65535
                adda.l   dØ,a4                    A4 -> name
                add.b    Ø(a4,a6.l),d1            D1:= 2 + name length
                addq.w   #1,d1                    Make length
                bclr     #Ø,d1                    even.

alloc           move.l   d1,d6                    Save number of bytes
                movea.w bv_chrix\w,a2             Request space on
                jsr      (a2)                     Maths Stack.
                movea.l bv_rip(a6),a1             Fetch new ri-stack
                suba.l   d6,a1                    Adjust pointer
                move.l   a1,bv_rip(a6)
                cmp.b    #3,d7                        Return % or $ ?
                bne.s    string                       Jump if $
                move.w   4(a5,a6.l),Ø(a1,a6.l)        Stack line No.
goodbye         move.l   d7,d4                        Result type
no_problem      moveq    #Ø,dØ                        No error
                rts
*
string          subq.w   #2,d6                        Adjust count
                clr.b    Ø(a1,a6.l)                   Clear length.H
copy            move.b   Ø(a4,a6.l),1(a1,a6.l)        Copy bytes
                addq.l   #1,a4
                addq.l   #1,a1                    Adjust pointers
                dbra     d6,copy                  Copy the whole name
                bra.s    goodbye                  Return to BASIC
*
get_colours     moveq    #Ø,d1                    Clear odd bytes of D1
                move.b   69(aØ),d1                Strip
                swap     d1                       Make room for ink
                move.b   7Ø(aØ),d1                Ink
                bra.s    no_problem
*
get_csizes      moveq    #Ø,d1                    Clear high word of D1
                move.w   28(aØ),d1                XSIZE
                move.w   38(aØ),d2                XINC
                divu     d2,d1                    D1.W := number of chars
                subq.w   #1,d1                    Don't use last column
                swap     d1
                move.w   4Ø(aØ),d1                Put YINC in high word
                swap     d1
                bra.s    no_problem
*
define          dc.w     Ø,Ø,3                    No procedures 3 functions
                dc.w     name-*                   Offset to code
                dc.b     6,"_NAME$"               Name length and text
                ds.w     Ø                        Align to a word boundary
                dc.w     defs-*                   Offset to code
                dc.b     5,"_DEF$"                Name length and text
                ds.w     Ø                        Align to a word boundary
                dc.w     defn-*                   Offset to code
                dc.b     5,"_DEF%"                Name length and text
                ds.w     Ø                        Align to a word boundary
                end
```

ably missing the list altogether!

The standard rom vector table includes 50-odd entries, from MM.ALCHP at address 192, which allocates Common Heap entries, to MD.SECTR at 298, which reads a microdrive sector header. These are well-known and often used, listed in books such as Dickens' *QL Advanced User Guide* and Pennell's *Qdos Companion*.

A further eight vectors were added in JS and MG roms, including Thors and Minervae; they point to tables and code used by the SuperBasic interpreter. Other vectors near the end of MG roms indicate language-specific options.

The example call to BV_CHRIX is not very useful, as reliable QL functions reserve Maths Stack space as they go along – but it would often be useful to reserve space in other parts of the SuperBasic memory area.

For instance, you might want to expand the Program File in a search-and-replace routine, or extend the Channel Table in a variant of OPEN. Code that generates values for SuperBasic may wish to make room in the Variable Values area.

INPUT allows longer lines and commands like COPY and MORE go faster if you increase the size of SuperBasic's Buffer area. This is most important on AH and JM roms, which otherwise limit INPUT to 128 characters.

Corruption of programs that use more than nine parameters or local variables in one DEF can be avoided by making room at the end of the Name Table. Allow eight bytes for every name, and then a bit more to be on the safe side, to prevent program corruption when using AH, JM and JS roms.

It is hard work to extend Basic areas using the routines documented by Sinclair. BV_CHRIX will grab space for the Maths Stack, but the only other system routine to expand Basic is MT_ALBAS, MTRAP 24. This reserves space in the middle of the interpreter's memory area, above all the tables and before the stacks.

If you need room somewhere else it appears necessary to move all the intervening stuff, adjusting various Basic memory pointers as you go along. This is a slow and fiddly process. Sinclair must have routines to do this sort of thing already buried in the rom – but how can we get hold of them?

Some of Anders' questions prompted me to investigate SuperBasic storage allocation. In that process I uncovered a further eleven vectors that are reliable in all QL versions, from AH to Minerva; they are called in popular programs, yet they have never been documented.

If you have explored the QL rom with a disassembler you may have noticed that the BV_CHRIX vector points to one entry in a table of routines that check for space in Basic tables and stacks. You can access other entries by adding standard offsets to the BV_CHRIX vector address.

The table sets out offsets to check for D1 bytes in each area. Thus, to reserve N free bytes at the end of the Program File:

```
CALL PEEK_W(282) + 52, N
```

I checked these on Sinclair's JM rom, Minerva 1.64, and on my Thor XVI version 6.41; they should be the same on your rom, and every other. The order and offsets of these entry points is fixed, even in Minervae. They are reliable as they are used by Toolkits and disk systems to expand the Channel Table.

The _DEF functions use the SuperBasic Buffer area to hold the index to procedure and function names. This always has at least 128 bytes, allowing 64 names, but is usually larger, and can be expanded by SPACE bytes with CALL PEEK_W(282) + 28, SPACE. 1000 bytes is usually plenty for MORE and the _DEF functions.

Code commentary

This listing continues from listing one in January's DIY Toolkit. Label CHOOSE is reached when the window has been filled with names. The program moves the print position to the top left corner of the screen. The INVERT subroutine swaps foreground and background colours so that the name appears highlit when re-displayed by PRINT_NAME.

At KEY_LOOP the program reads a key

with a call to ID_FBYTE. A 'bad parameter' error is detected here if you have chosen a SCR channel, which does not allow keyboard input.

Anders' comprehensive error-traps come in handy again in the ARROW routine, which moves to a new line and scrolls the screen if necessary. The keyboard handler loads the required TRAP key into D6: SD_PROW to move up or SD_NROW to go down. The screen scrolls if the row TRAP indicates an 'out of range' error. The value of D6 determines the direction scrolled; the number of pixels comes from D4, set by last month's EXTOP call. The previous name is reprinted in default colours before the move takes place, so the highlight appears to move between names.

Next comes a clutch of subroutines. PRINT_NAME locates text in the Name List, and trims it if necessary to fit the window. The address in A1 is relative to A6, as it is inside SuperBasic, so TRAP #4 is used before the name is printed with ID_SSTRG. The TRAP3 routine is used throughout the listing. It returns only ERR_OK (no problem) or ERR_OR (out of range), propagating other errors to the previous caller, to be reported to the user.

The menu functions end by jumping to DONE, which checks that there is room for the result on the Maths Stack and copies it into place. _NAME$ joins in at STACKER.

INVERT exchanges INK and STRIP colours, swapping the values in D5 at each call. GET_COLOURS and GET_CSIZES are the two EXTOP routines called to read channel definition details. Each returns a pair of values in the high and low words of D1. They could be combined by packing values together or using A1 as well as D1 to return all four results.

At the end of the listing, DS.W 0 is used to ensure that table values are at even addresses. This is not strictly necessary, as good assemblers do not try to put word values at odd addresses, where the 68008 cannot reach them. The Computer One assembler uses ALIGN and other assemblers use CNOP 0,2 for the same effect.

Code extensions

The existing code could be usefully extended in several ways. When editing large programs it would be convenient to be able to page through the menu, or skip directly to the start or end, as most recently added names will be at the end. Qptr users might add pointer control, so that names could be selected with a mouse or track-ball.

Complete source, binary code and documentation for these functions is available on disk or cartridge from **DIY Toolkit, Cwm Gwen Hall, Pencader, Dyfed, Cymru SA39 9HA, tel. (0559) 384574.** These routines are part of Volume A, along with CODEVEC, ALIAS and INVERSE from November's QL World, and variants.

Material from past DIY Toolkit columns, and much besides, is available in 18 volumes priced at £3 each, plus £4 per order to cover disks, processing and postage. Bargain Bundles of six volumes cost £20 each. Please enclose one formatted cartridge for each volume required on microdrive.

Offset	Area Name
+0	RI Stack
+6	Backtrack Stack
+12	Temporary Graph Stack
+18	Name Table
+24	Return Table
+28	Buffer
+32	Token List
+36	Name List
+40	Variable Values
+44	Channel Table
+48	LIST Line Table
+52	Program File

Table one: BV_CH__X Vector offsets.

SOFTWARE FILE

PROSPERO PASCAL

With programmers who are interested in Pascal in mind, A F Wilson tries out Prospero Software ProPascal.

For QL programmers constricted by the slowness and non-multitasking capability of SuperBasic, or for the serious programmers thinking of writing code for a number of machines or thinking to the future when they have to upgrade their trusty QLs, moving over to Pascal, in particular to *Prospero Pascal* should seriously be considered.

Pascal is a truly portable language and much easier to learn and program in than the 'C' language. SuperBasic in many ways is very similar to Pascal and this should make the transition from SuperBasic to Pascal much easier.

Prospero software versions of Pascal not only conform to the ISO 7185 level 0 Pascal standard, but are also available for a wide range of different computers – CPM, MS-DOS, OS/2 and the 68 K series: QL and Atari ST. Prospero are one of the main language houses offering a wider audience to C, Pascal and Fortran than even Microsoft and Borland who concentrate mainly in the MS-DOS and OS/2 arena.

Prospero's Pascal range not only offers one of the most powerful of the Pascal implementations but provides a natural upgrade route. For me, moving from ProPascal (CPM) on my Memotech FDX system to the QL was painless. Approximately 80 per cent of the code was as CPM. The remaining 20 per cent was the machine dependent code – sound, graphics and file handling quirks brought about by the different Operating systems, like filename convention (ie size and structure).

Effortless

Converting SuperBasic programs was just as effortless, even more so as the Graphic command extensions provided by Prospero were almost identical to SuperBasic. Pascal though, is a bit stricter than SuperBasic with respect to type assignments and unlike SuperBasic certainly does not allow any form of coercion (*see QL User Guide, Concepts page 46*).

However, this does force the programmer to plan the design of the program more thoroughly and ultimately to the generation of better code. The obvious benefits of Pascal over SuperBasic were the ability to multi-task, speed of execution, compact code and compatibility when upgrading to a different computer.

In addition to the standard Pascal language, Prospero implementations provide a number of powerful extensions, many of which are universally accepted, even Borland's Turbo Pascal compilers adhere to them. In particular, floating-point arithmetic and dynamic string handling.

The ProPascal range adheres to the same extensions throughout the different implementations whether on the QL or OS/2. However, one or two specialist commands are still needed to access the different operating systems, ie QTRAP on the Sinclair QL or SYS on the Atari ST.

ProPascal costs £101.50 and is available from Prospero Software or from TK Computerware. ProPascal consists of 3 microdrive cartridges, PRL rom (version 1.51) and an A5 hardback ring-binder manual. **Figure one** contains a list of what is found on the three microdrives and a brief description of their function. The obvious omission was the lack of Ascii or text Editor. However, this was no big disappointment for the QL programmer who is blessed with a number of excellent Ascii editors.

An A5 hardbacked ring-binder manual contains all the formal information on the Pascal language and the ProPascal extensions with one or two examples. A number of demonstration listings are provided on disk, these illustrate key programming techniques in Pascal. The manual also contains information on interfacing Fortran and 68 K code with Pascal.

The manual lacks a decent tutorial. However, as Pascal is one of the main teaching languages there are a number of excellent books around catering for the novice to the advanced user:

Mastering Pascal Programming by E Huggens, published by MacMillan (for the novice user).

Pascal, an Introduction to Methodical Programming by W Findlay and D A Watt, published by Pitman (for the advanced user). First of all disconnect the QL power supply and plug in the development Prospero Resident Library Rom (v1.51) into the cartridge port at the back. Reconnect the QL power supply and switch on. The opening page after switch on will indicate that PRL has been installed correctly.

The next step is to make working copies of the program to either disk or microdrive depending on your system. I have a JS rom QL with extended ram and a disk interface. Therefore I copied the contents of the three microdrives to disk 1 – marked DISK version. This was repeated to disk 2 – marked RAM version.

Verifying

Running the utility pcheck (exec flpl_pcheck) on disk 1 and subsequently on disk 2 to verify that the copying was okay (no problems with the WCOPY command on TK2). The next step was to update some of the programs to recognise that they are to be run from disk or ram. Utility setdev will do this, (see manual part III).

Microdrive users will need to split up the compiler onto three microdrives due to capacity restrictions. Prospero recommend that files PAS, PROPAS1, PROPAS_ERR

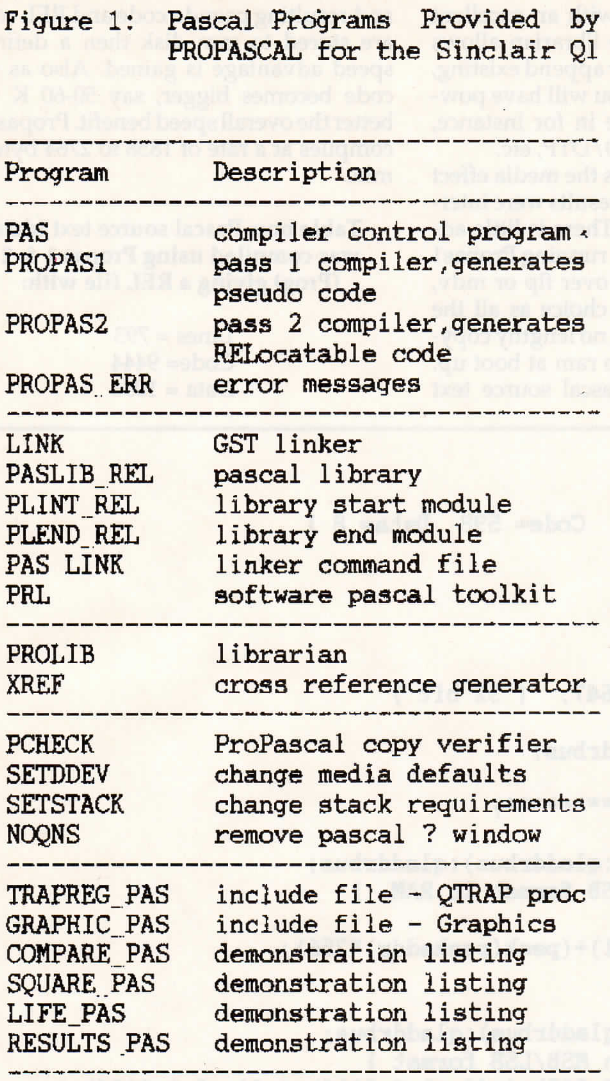
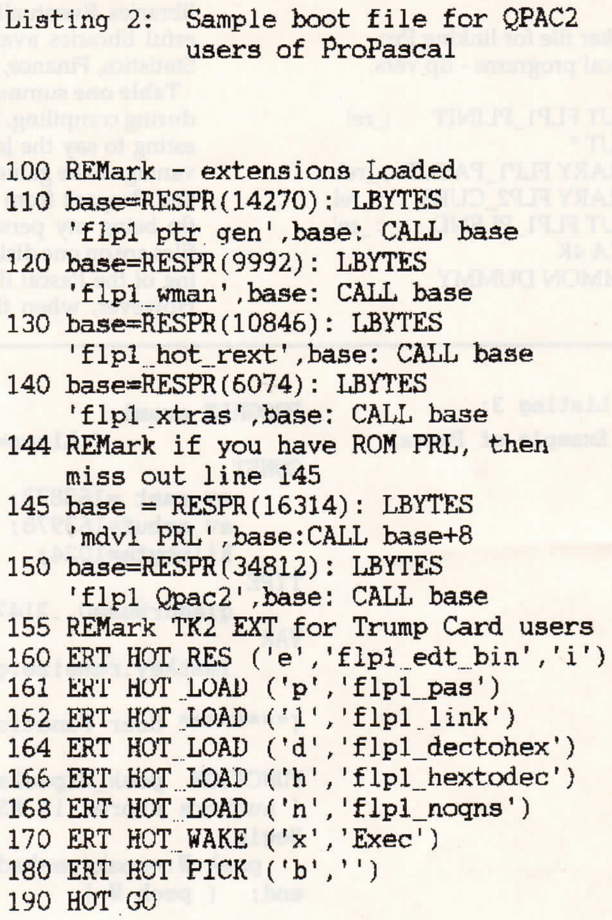


Figure 1: Pascal Programs Provided by PROPASCAL for the Sinclair Ql

Program	Description
PAS	compiler control program
PROPAS1	pass 1 compiler,generates pseudo code
PROPAS2	pass 2 compiler,generates RELocatable code
PROPAS_ERR	error messages
LINK	GST linker
PASLIB_REL	pascal library
PLINT_REL	library start module
PLEND_REL	library end module
PAS_LINK	linker command file
PRL	software pascal toolkit
PROLIB	librarian
XREF	cross reference generator
PCHECK	ProPascal copy verifier
SETDDEV	change media defaults
SETSTACK	change stack requirements
NOQNS	remove pascal ? window
TRAPREG_PAS	include file - QTRAP proc
GRAPHIC_PAS	include file - Graphics
COMPARE_PAS	demonstration listing
SQUARE_PAS	demonstration listing
LIFE_PAS	demonstration listing
RESULTS_PAS	demonstration listing



Listing 2: Sample boot file for QPAC2 users of ProPascal

```
100 REMark - extensions Loaded
110 base=RESPR(14270): LBYTES
    'flp1_ptr_gen',base: CALL base
120 base=RESPR(9992): LBYTES
    'flp1_wman',base: CALL base
130 base=RESPR(10846): LBYTES
    'flp1_hot_rext',base: CALL base
140 base=RESPR(6074): LBYTES
    'flp1_xtras',base: CALL base
144 REMark if you have ROM PRL, then
    miss out line 145
145 base = RESPR(16314): LBYTES
    'mdv1_PRL',base:CALL base+8
150 base=RESPR(34812): LBYTES
    'flp1_Qpac2',base: CALL base
155 REMark TK2 EXT for Trump Card users
160 ERT HOT_RES ('e','flp1_edt_bin','i')
161 ERT HOT_LOAD ('p','flp1_pas')
162 ERT HOT_LOAD ('l','flp1_link')
164 ERT HOT_LOAD ('d','flp1_dectohex')
166 ERT HOT_LOAD ('h','flp1_hextodec')
168 ERT HOT_LOAD ('n','flp1_noqns')
170 ERT HOT_WAKE ('x','Exec')
180 ERT HOT_PICK ('b','')
190 HOT_GO
```

be stored on one microdrive and PROPAS and PROPAS2 and PROPAS_ERR on the second and the remaining files on the third microdrive. I would recommend users upgrade the ram capacity of their QLs to take advantage of the speedier ramdisk.

Don't forget to update the boot file. **Listing two,** described fully under the Multi-tasking heading shows a typical bootup file for ProPascal running on *Qpac2*. Lastly, the Pascal source text generated with the Editor should be kept separate from the programs and all files must end with the '_PAS' extension.

The two-pass compiler converts the Pascal program (source text) into a relocatable binary machine code file (object code). Typing 'exec flp1_pas', invokes the Pascal controller program (PAS). PAS requires the filename (without extension) of the Pascal source text to be entered, ie flp2_square and the compile time options:

- Compact code or Faster speed
- ISO only Pascal
- Pointer/Indexing/Assignment checks
- Log/Prn files
- Track source names and Line numbers at runtime.

PAS hands over this information to PROPAS1 which checks the source text syntax and if legal coverts it to pseudo or intermediate code. Any errors are extracted from PROPAS_ERR and echoed to the console, and echoed to a LOG file on disk. There are 405 different errors that can be highlighted compared with the feeble 22 available in SuperBasic.

PAS only passes valid pseudo code to PROPAS2 for the final part of the compilation. The pseudo code is converted into a relocatable binary file conforming to the Sinclair Relocatable format. The file generated has _REL as the extension.

Libraries

The next step is to link the REL code with the Pascal libraries. The linker used is the standard GST linker as this conforms to the Sinclair REL format. The result is a binary file, with the _BIN extension. The BIN file can be EXECuted using exec flp1_fname_bin. The BIN file can also be multitasked with other executable files.

Note that either the rom PRL or the software PRL must be initialised before the file will execute properly. PRL is a Runtime Pascal toolkit containing the most used Pascal commands. Runtime toolkits minimise ram constraints and enable a number of Pascal programs to be run concurrently. This was particuarly important before cheaper ram expansion. The TurboBasic SuperBasic compiler also requires a similar Runtime toolkit – called XTRAS.

The bulk of the Pascal program will use standard or extended Pascal Functions/Procedures provided. During the linking stage, the linker will extract the REL code from the Pascal library and link the REL program file to produce the BIN file.

Any part of the program which draws commands from say 68 K or from Fortran REL libraries should be included in the linking stage. Note that only REL libraries that conform to the Sinclair REL format are suitably ProFortran and the GST macro assembler both do this.

If you do use a number of different libraries then I suggest you write a Linker command file like the one in **Listing one.** When you come to link your REL program with the libraries all you have to enter is flp_2 program flp1_pas. "pas" in this case is that contained in listing one. The two main libraries here are PASLIB and CURS. CURS is a 68 K rel file that allows the cursor to be switched on and off in different windows, similar to the toolkit 2 commands CURSEN and CURDIS.

Listing one: Linker Command File: Link_pas

```
; Linker file for linking Pro
; Pascal programs - flp vers.
;
INPUT FLP1_PLINIT      ;_rel
INPUT *
LIBRARY FLP1_PASLIB ;_rel
LIBRARY FLP2_CURS   ;_rel
INPUT FLP1_PLEND    ;_rel
DATA 4K
COMMON DUMMY
```

This package comes with an excellent librarian program. The librarian allows the user to build new, or append existing, libraries. Eventually, you will have powerful libraries available in for instance, Statistics, Finance, CAD/DTP, etc.

Table one summarises the media effect during compiling. The results were interesting to say the least. There is little advantage to be gained by running Propas1 and Propas2 from ram over flp or mdv, flp being my personal choice as all the files are on one disk and no lengthy copying of the Pascal files to ram at boot up. However, when the Pascal source text and resulting pseudo code and REL code are stored to ram disk then a definite speed advantage is gained. Also as the code becomes bigger, say 50-60 K the better the overall speed benefit. Propascal compiles at a rate of 1858 to 2764 bytes/min.

Table one: Pascal source text (data) was compiled using Propas 1 & 2 (Prog) giving a REL file with:

Lines = 793
Code= 9444
Data = 1036

Listing 3:
Example of Pascal

```
PROGRAM exam1;
  { Lines= 60  Code= 598  Data= 8 }
CONST
  sv_ramt =163872;
  sv_arbuf=163978;
  kilobyte=1024;
TYPE
  qladdrbus=0..2147483647;  { 32 bit }
VAR
  lastkey,ramsize:qladdrbus;

{******** User Functions ********}

FUNCTION peek_W(peekaddr:qladdrbus):qladdrbus;
{ numbers stored in MSB/LSB format in RAM. }
Begin
   peek_W:=peek(peekaddr+1)+(peek(peekaddr)*256);
end;  { peek_W }

FUNCTION peek_L(peekaddr:qladdrbus):qladdrbus;
{ numbers stored in RAM in MSB/LSB format }
{ (addr+3 * 2^0)+(addr+2 * 2^8)+(addr+1 * 2^16)+(addr+0 * 2^24) }
begin
  peek_L:=peek(peekaddr+3)+(peek(peekaddr+2)*256)+
          (peek(peekaddr+1)*65536)+(peek(peekaddr)*16777216);
end; { peek_L }

{* Graphics extensions to PASCAL - machine dependent ( Ql - QDOS ) *}

PROCEDURE mode(highres: boolean);  EXTERNAL;

PROCEDURE window(VAR w: text;
         width,height,Xorigin,Yorigin: integer);  EXTERNAL;

PROCEDURE cls(VAR w: text; part: integer);  EXTERNAL;

BEGIN { main program }

mode(true);                    { Monitor mode }
window(output,420,160,20,20);  { redefine/reposition default window }
cls(output,0);                 { cls whole window }
ramsize:=peek_L(sv_ramt);
lastkey:=peek_W(sv_arbuf);
writeln(output,' Ql Ram Size is ',trunc(ramsize/kilobyte):4,' K ');
writeln(output,' Last Key Pressed ASCII number was ',lastkey );
writeln(output);
writeln(output,' Press ENTER to exit ');
readln(input);
END.
```

Media		Propas1	Propas2	Total	Speed
Prog	Data	<——	seconds	——>	Factor
MDV	MDV	140	165	305	1.0
FLP	FLP	140	165	305	1.0
RAM	FLP	125	150	275	1.1
FLP	RAM	95	120	215	1.4
RAM	RAM	85	120	205	1.5

I have used Propascal as a stand-alone package, attempted to use with *Taskmaster* before opting for Qpac2. Propascal is fully compatible with both Taskmaster and Qpac2, however, it is totally unworkable under Taskmaster.

The difficulty with Pascal and Taskmaster is a simple one. All ProPascal programs on completion remove themselves from ram, similar to NEWing in SuperBasic whether compiling or linking. This means that they are also removed from Taskmaster, as Taskmaster requires memory resident programs which only remove themselves from ram when the user selects, for instance – QUIT as used for *Quill* or *Editor* or *Abacus* etc. As a result using Pascal within this environment is useless.

On the other hand, Qpac2 is ideal for the job, see listing two for a typical Qpac2/Hotkey boot file. Qpac2 can accommodate both memory resident applications (using HOT_RES) and non-memory resident applications (using HOT_LOAD). Pressing ALT P is all that it takes to run PAS everytime with ease. Running from RAMdisk is extremely fast and benefits from Qpac2/Hotkey environment.

Only compaints

I have used ProPascal on my QL for a couple of years now and my only complaints about the implementation have been:

- Lack of debugger/monitor to help find bugs in Runtime code or to aid in trapping errors
- Lack of DIR command within the Pascal extensions
- Ability to compile fully without the need of PRL. No PRL to install. Although there is no royalty problem with the software PRL being included in any commercial software you write
- Executing a Pascal produced program results in a default window opening – this is needed to display possible runtime errors. However, it would have been nice to have a utility that would let you reposition/reduce this window without having to load the BIN file into your editor and changing from there. This includes converting the width to its hexadecimal lsb/msb components, for instance 480 becomes EO (capslock) and 01 (CTRL A).

Although a bit pricey considering there is no editor or symbolic debugger, ProPascal is an excellent implementation and very powerful to boot. Ideal working environment must include Qpac2/Hotkey with ram expansion. As far as I am concerned it is the only way to progress from SuperBasic because of the similarities. The prospect of re-using the programs I write, if I ever have to upgrade to say another computer, ie OS/2 or Unix machine, is an appealing thought.

Prospero Software provide excellent technical backup of their products and produce free *Pronews* magazines every four months to those that register with them. [Aside: Propascal (CPM) runs fine under *Success* the CPM emulator from DP.] Lastly, **Listing three** provides an example of how easy it is to produce two new commands – PEEK_W and PEEK_L like that provided by SuperBasic.

INFORMATION
Program: *ProPascal* (mmq 1.17 and PRL rom V1.51)
Price: £101.50
Supplier: Prospero Software, 190 Castelnau, London, SW13 9DH
Supplier: TK Computerware, Stone Street, North Stanford, Ashford, Kent, TN25 6DF
Tel: (0303)-81-2801

access CLUB

FRANCE

QL Contact France. Contact: Jean-Louis Dianoux, 38-40 Rue Stephenson, 75018 Paris, France.

BELGIUM

Club Sinclair BruQsL (Belgium) Contact: Jaques Tasset, Aarlenstraat 104, 1040 Brussels, Belgium

SWEDEN

International QL Conference bulletin board system (Swedish and English). Contact: Michael Cronsten, System Operator, Jamten-TCL, S Soere 1073, 83030 Lit, Sweden.

USA

New England Sinclair QL User Group (USA) Membership Secretary: Sherm Waterman, PO Box 8763, Boston, MA 02114 8763, USA. Magazine: *NESQLUG News.* Editor: Peter Hale, 195 Central Ave., Chelsea, MA 02150, USA.

NORWAY

Norwegian All Sinclair Association (NASA) Contact: P Monstad, NASA, N-5580 Oelen, Norway. Magazine: *Sinclair Magazine.*

ITALY

Qitaly Club Chairman: Roberto Orlandi, Via Brescia 26, 25039 Travagliato (BS), Italy. Tel. (local) +39 30 6863311. Magazine: *Qitaly Magazine.* Editor: Dr Eros Forenzi, Via Valeriana 44, 23010 Berbenno (SO), Italy. Tel. (local) +39 342 492323.

TURKEY

QL Qlub (Turkey). Contact: Bulent Artuz, Prof. Sitesi B/1 D/5, Etiler 80600, Istanbul, Turkey.

AUSTRIA

DER ComputerClub. Contact: Peter Postl, Stiegerg 5, 1150 Wien, Autstria.

ENGLAND

Quanta (UK) Membership Secretary, Bill Newell, 213 Manor Road, Benfleet, Essex SS7 4JD. Magazine: *Quanta.* Editor: Bill Fuggle, 20 Widnes Avenue, Selly Oak, Birmingham B29 6QE.

Bristol sub-group: Roy Brereton, 94 Teignmouth Rd, Clevedon, Avon BS21 6DR.

Essex sub-group: Dave Walker, 22 Kempton's Mead, Potters Bar, Herts EM6 3HZ.

London sub-group: Jeremy Davis, 6 Elmcroft Crescent, Harrow, Middlesex HA2 6HN.

Northern Ireland: Billy Turkington, Fairyhill, Rostrevor, Newry, Co., Down BT34 3BB.

Qubbesoft PD library. Contact: Ron Dunnett, 38 Brunwin Rd, Rayne, Braintree, Essex CM7 5BU.

SCOTLAND

Scottish QL Users Group Contact: Alan Pemberton, 65 Lingerwood Rd., Newtongrange, Midlothian EH22 4QQ. Newsletter.

HOLLAND

Sin_QL_Air (Netherlands) Membership Secretary: Bob Visser, Snelrewaard 6, 2904 SN Capelle, a/d Ijssel, Netherlands. Magazine: *Quasar.* Editor: C H M Biemans, Elzenstraat 5, 5461 CL Veghel, Netherlands.

GERMANY

Sinclair QL User Club eV (Germany) Foreign Contact: Franz Herrmann, Talstrasse 21, d-W5460 Ochenfels, West Germany. Magazine: *Quasar.*

SPAIN

Qliper Editor: Marcos Cruz, Acacias 44, 28023 Madrid, Spain. Magazine: *Qliper.*

Systemati
Machine Code
Programmi

In the fourth part of this series on developing machine code skills, Alan Bridewell takes his routines and merges them together to make an example program.

In this series, we are building up a library of small chunks of assembler language code. Each of these will do a recognisable job within a program, and be fully annotated, so that it will be clear exactly how to join them to other chunks of code to make a program and, in particular, how to make any necessary changes to make the chunks fit together.

In part 3, we saw how to handle screen output control (at least all that we could without using floating point numbers).

Now to try some of this out: clearly a simple demonstration of visual effects is not going to result in a very useful program. But then I am not so much trying to give you programs, as the means to produce your own! What I shall do is demonstrate some interesting effects, rather than try to put together a useful program.

Not a flowchart

Once again, rather than use the flow chart of the professional, I will put together a series of statements about what the program should do, together with the name of the chunk of code to do it, and any important points to be made.

1. We start with the usual "JOBSTART', with an appropriate job name.

2. Next, using 'CONSOLE', we open a console channel to cover most of the screen (except #0), put a border and clear the window.

3. Having cleared the screen of any remaining garbage, we redefine a small window, with 'WDEFINE', position the cursor with 'POSITIONCURSOR' and print a message using "MESSAGE'.

Fixing the border

4. We want this message in a protected border, so we use "BORDER' to overwrite the message with a transparent border.

5. Using "COLOURS' three times, we redefine Paper, Strip and Ink, then CLS to clear the whole window inside the trans-

Listing 1

```
; *****************************************************************
;                         'JOBSTART'
;
                   BRA.S      START        ; BRANCH TO START OF CODE
                   DC.L       0            ; (THIS IS STANDARD FORMAT FOR
                   DC.W       $4AFB        ; START OF A JOB)
; ---- ALTER CHARACTER COUNT AND JOB NAME ----
                   DC.W       7            ; CHARACTER COUNT OF JOB NAME
                   DC.B       'DISPLAY'     ; NAME OF JOB
;
; *****************************************************************
;                         'CONSOLE'
;
; OPEN THE CONSOLE CHANNEL
;
; ---- ALTER LABEL TO .START ----
.START             LEA.L      PBLOCK,A1                ; PBLOCK ADDRESS IN A1
                   MOVE.W     $C6,A2                   ; UT_CON VECTOR IN A2
                   JSR        (A2)                     ;
;
; SAVE THE CHANNEL ID WHICH UT_CON ROUTINE LEAVES IN AO.
;
                   MOVE.L     AO,-(A7)              ; SAVE CONSOLE ID ON STACK
                                                    ; * NOTE * THIS CHANGES A7
                                                    ; BY -4, READY FOR NEXT
                                                    ; STACK ENTRY.
;
; CLEAR THE WINDOW
;
                   MOVE.W     #$FFFF,D3                ; INFINITE TIMEOUT
                   MOVEQ      #$20,DO                  ; #SD_CLEAR IN DO
                   TRAP       #3                       ;
; ---- ALTER BRANCH ADDRESS TO WDEF1 ----
                   BRA.S      WDEF1                    ; SKIP BLOCK
;
; ---- ALTER BLOCK TO REQUIRED WINDOW ----
.PBLOCK            DC.B       2                        ; RED BORDER
                   DC.B       2                        ; 2 PIXELS WIDE
                   DC.B       0                        ; BLACK PAPER/STRIP
                   DC.B       7                        ; WHITE INK
                   DC.W       512                      ; WIDTH
                   DC.W       200                      ; HEIGHT
                   DC.W       0                        ; X POSITION
                   DC.W       0                        ; Y POSITION
;
; *****************************************************************
;                         'WDEFINE'
;
; ---- DELETE LINES TO ENTER ID IN AO AND TIMEOUT IN D3 ----
; ---- ALTER LABEL TO .WDEF1 ----
; ---- ALTER LOAD ADDRESS TO WDEFINE1 ----
.WDEF1             LEA        WDEFINE1,A1  ; WDEFINE BASE IN A1
                   MOVEQ      #$D,DO      ; #SD_WDEFINE IN DO
; ---- ALTER NEXT TWO LINES FOR REQUIRED BORDER ----
                   MOVEQ      #$2,D1      ; RED BORDER
                   MOVE.W     #$2,D2      ; BORDER 2 PIXELS WIDE
                   TRAP       #3          ;
; ---- ALTER BRANCH ADDRESS TO CURS ----
                   BRA.S      CURS        ; SKIP BLOCK
;
; ---- ALTER LABEL TO WDEFINE1 ----
; ---- ALTER BLOCK TO REQUIRED WINDOW ----
.WDEFINE1          DC.W       256         ; 256 PIXELS WIDE
                   DC.W       40          ; 40 PIXELS HIGH
                   DC.W       10          ; X ORIGIN
                   DC.W       5           ; Y ORIGIN
;
; *****************************************************************
```

c
ing

parent border, so leaving the message in place.

6. A second message in two lines is put into the window, using 'MESSAGE', 'MOVECURSOR' (for a NL, CR), and 'MESSAGE' again.

7. To show the channel can be used again to write a message elsewhere, we use 'WDEFINE' to give a new window, which is cleared with CLS. 'CSIZE' produces larger characters, before 'MESSAGE' writes a message in the new window.

WDEFINE eight times

8. 'WDEFINE' is now used eight more times to leave a series of borders, each one mostly overwriting the previous one, for a 3D effect. The final window uses CLS to clear, 'CSIZE' to produce a larger character, and 'COLOURS' to change the Ink, before using 'MESSAGE' to print in the window.

9. 'COLOURS' again changes the Ink and 'CSIZE' is used again to make smaller characters, before 'MOVECURSOR' is used three times to prepare for the last message with 'MESSAGE'.

10. The final embellishments are made to the display by using 'WRITEMODE' to put overwriting in XOR mode, before using 'BLOCKFILL' six times to change the colours of the words on the final message.

11. We end in the normal way by using 'CLOSE' to close the channel, and 'ENDJOB' to kill the job.

Putting all that together, we get:

JOBSTART, CONSOLE, WDEFINE, POSITIONCURSOR, MESSAGE, BORDER, COLOURS, COLOURS, COLOURS, CLS, MESSAGE, MOVECURSOR, MESSAGE, WDEFINE, CLS, CSIZE, MESSAGE, WDEFINE, WDEFINE, WDEFINE, WDEFINE, WDEFINE, WDEFINE, WDEFINE, WDEFINE, CLS, CSIZE, COLOURS, MESSAGE, COLOURS, CSIZE, MOVECURSOR, MOVECURSOR, MOVECURSOR, MESSAGE, WRITEMODE, BLOCKFILL, BLOCKFILL, BLOCKFILL, BLOCKFILL, BLOCKFILL, BLOCKFILL, CLOSE, ENDJOB

```
;                               'POSITIONCURSOR'
;
; ---- DELETE LINES TO ENTER ID IN A0 AND TIMEOUT IN D3 ----
.CURS           MOVEQ     #$10,D0    ; #SD_POS IN D0 (CHARS)
; ---- ALTER NEXT TWO LINES FOR REQUIRED ROW AND COLUMN ---
                MOVE.W    #$6,D1     ; COLUMN 6
                MOVE.W    #$0,D2     ; ROW 0
                TRAP      #3         ;
;
; ******************************************************************
;                                'MESSAGE'
;
; ---- ALTER LOAD ADDRESS TO MESSAGE1 ----
; ---- DELETE LINE TO ENTER ID IN A0 ----
                LEA.L     MESSAGE1,A1 ; BASE ADDRESS IN A1
                MOVE.W    $D0,A2      ; UT_MTEXT VECTOR IN A2
                JSR       (A2)        ;
; ---- ALTER BRANCH ADDRESS TO BORD ----
                BRA.S     BORD        ; SKIP MESSAGE
;
; ---- ALTER LABEL TO MESSAGE1 ----
; ---- ALTER TO REQUIRED CHARACTER COUNT AND MESSAGE ----
;
.MESSAGE1       DC.W      29          ; LENGTH OF MESSAGE
                DC.B      'THIS IS A MESSAGE IN A BORDER' ;
;
; ******************************************************************
;                                'BORDER'
;
; ---- DELETE LINES TO ENTER ID IN A0 AND TIMEOUT IN D3 ----
.BORD           MOVEQ     #$C,D0     ; #SD_BORDER IN D0
; ---- ALTER NEXT TWO LINES TO REQUIRED BORDER COLOUR AND WIDTH ----
                MOVEQ     #$80,D1    ; BORDER TRANSPARENT
                MOVE.W    #$A,D2     ; BORDER 10 PIXELS WIDE
                TRAP      #3         ;
;
; ******************************************************************
;                                 'COLOURS'
;
; ---- DELETE LINES TO ENTER ID IN A0 AND TIMEOUT IN D3 ----
                MOVEQ     #$27,D0    ; #SD_SETPA IN D0 (PAPER)
                MOVE.B    #$7,D1     ; WHITE PAPER
                TRAP      #3         ;
;
; ******************************************************************
;                                 'COLOURS'
;
; ---- DELETE LINES TO ENTER ID IN A0 AND TIMEOUT IN D3 ----
; ---- ALTER MOVEQ COMMAND TO #$28,D0 FOR STRIP ----
                 MOVEQ     #$28,D0   ; #SD_SETST IN D0 (STRIP)
                 MOVE.B    #$7,D1    ; WHITE STRIP
                 TRAP      #3        ;
;
; ******************************************************************
;                                 'COLOURS'
;
; ---- DELETE LINES TO ENTER ID IN A0 AND TIMEOUT IN D3 ----
; ---- ALTER TO MOVEQ #$29,D0 FOR INK ----
                MOVEQ     #$29,D0    ; #SD_SETIN IN D0 (INK)
; ---- ALTER TO MOVE.B #$2,D1 FOR RED ----
                MOVE.B    #$2,D1     ; RED INK
                TRAP      #3         ;
;
; ******************************************************************
;                                 'CLS'
;
; ---- DELETE LINES TO ENTER ID IN A0 AND TIMEOUT IN D3 ----
                MOVEQ     #$20,D0    ; #SD_CLEAR IN D0
                TRAP      #3         ;
;
; ******************************************************************
;                                'MESSAGE'
;
; ---- DELETE LINE TO ENTER ID IN A0 ----
; ---- ALTER LOAD ADDRESS TO MESSAGE2 ----
                LEA.L     MESSAGE2,A1 ; BASE ADDRESS IN A1
                MOVE.W    $D0,A2      ; UT_MTEXT VECTOR IN A2
                JSR       (A2)        ;
; ---- ALTER BRANCH ADDRESS TO MCURS ----
                BRA.S     MCURS       ; SKIP MESSAGE
;
; ---- ALTER LABEL TO MESSAGE2 ----
; ---- ALTER TO REQUIRED CHARACTER COUNT AND MESSAGE ----
;
.MESSAGE2       DC.W      36          ; LENGTH OF MESSAGE
                DC.B        BUT THIS IS IN THE REMAINING PART   ' ;
;
; ******************************************************************
;                               'MOVECURSOR'
;
; ---- DELETE LINES TO ENTER ID IN A0 AND TIMEOUT IN D3 ---
.MCURS          MOVEQ     #$12,D0    ; #SD_NL IN D0 (CR,LF)
                TRAP      #3         ;
;
; ******************************************************************
;                                'MESSAGE'
;
; ---- DELETE LINE TO ENTER ID IN A0 ----
```

Now comes the job of making all the small alterations needed to make the chunks fit together as a program. They are basically as follows:

Unique labels

1. Make sure all the definition blocks in 'CONSOLE', 'WDEFINE' and 'BLOCKFILL' have unique labels, and they are altered in the LEA command to match. Similarly, 'MESSAGE' must have a unique label for each message.

2. Each time there is a definition block or a message, it must be skipped over with the BRA.S command. So the following chunk must be given a unique label matching the BRA.S command.

3. All the definition blocks and messages must have their data adjusted as appropriate.

4. All unwanted comments should be removed.

5. As we have a succession of trap #3 calls using the same channel, the channel ID in A1 and the infinite timeout in D3 remain unaltered, so they do not need to be entered repeatedly. They can be deleted every time they occur except the first time.

If you have a printer, you could have a printout of the merged chunks of code, and then make alterations to this by hand, before altering the actual program. Alternatively, you can alter each chunk of code as it is merged into your listing file. **Listing one** shows the results of this work, but with two differences. In order to show how the final listing relates to the merged chunks of code, it contains the following rows of asterisks left between each chunk and additional comments, added to show where the code needs to be changed. These comments start and end with "——".

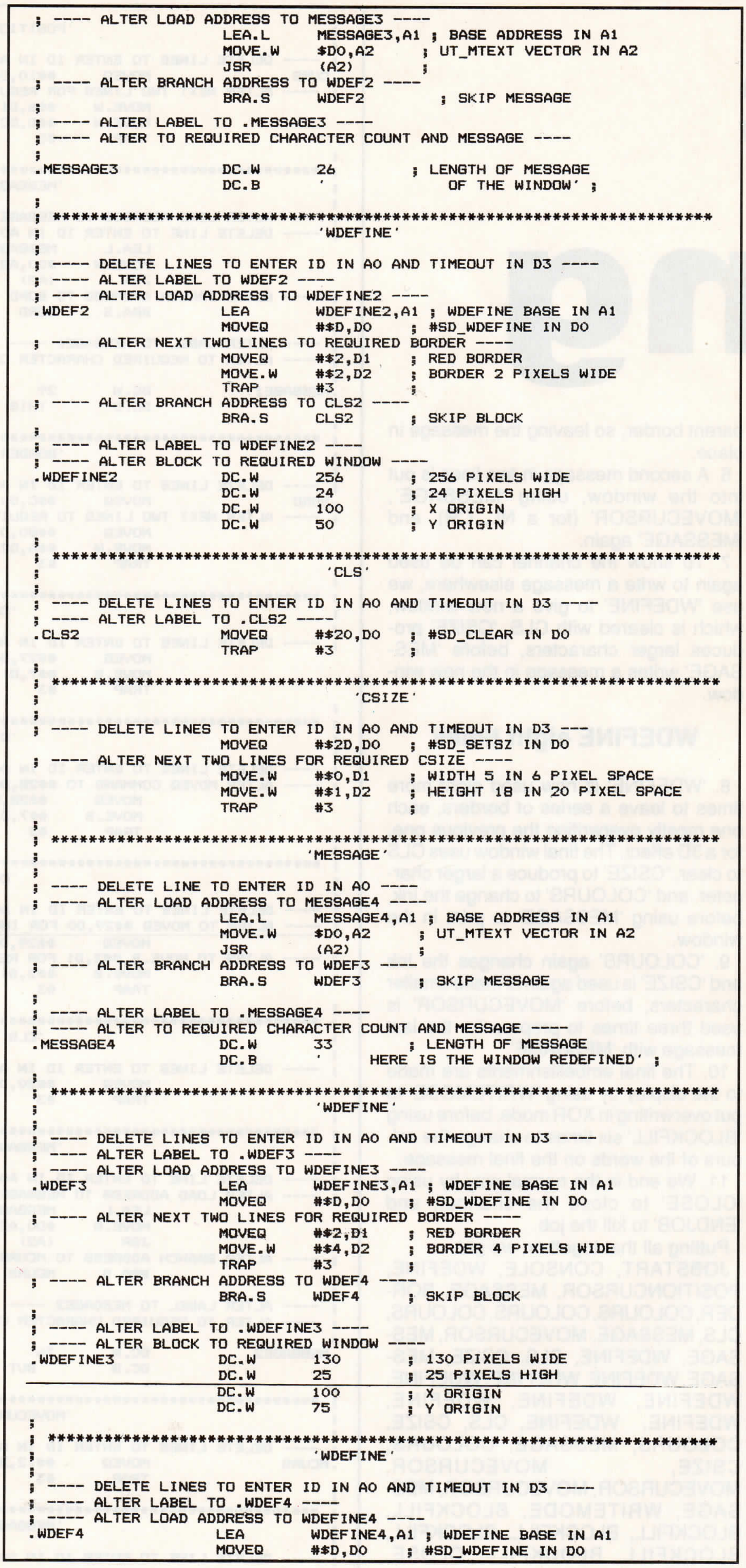
Through the steps

We will now go through the steps needed to be carried out if we were going to merge the chunks and alter them as we merge them. For the sake of brevity, I shall leave out any instructions for tidying up the listing by removing unwanted remarks, because they don't actually affect the code.

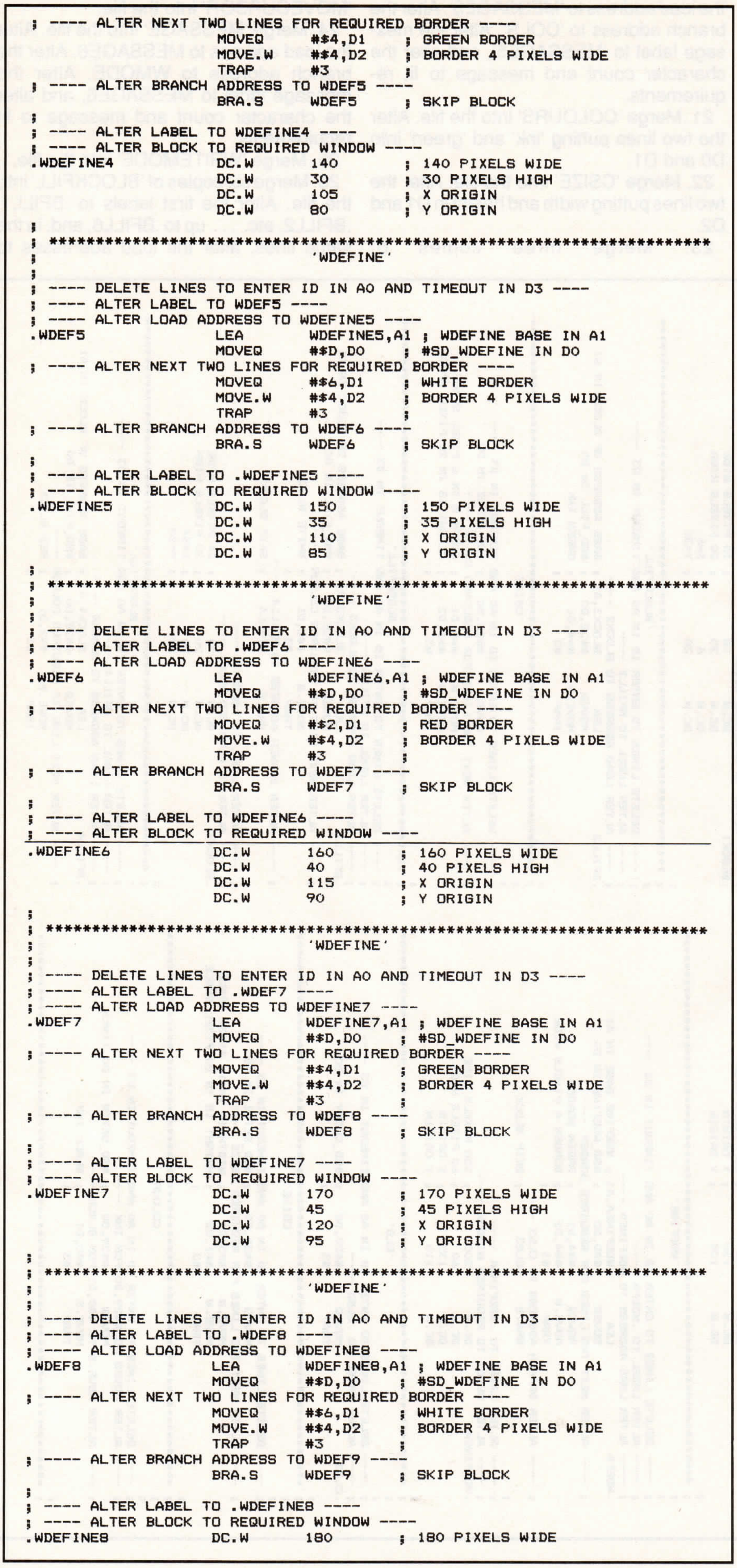
1. Merge 'JOBSTART' into the file, and alter the character count and job name.

2. Merge 'CONSOLE' into the file. Alter the first label to START and the branch address to WDEF1. Alter the window block to the required parameters.

In all the following steps that involve merging TRAP #3 calls into the file (that is, all the merged chunks except the 6 times 'MESSAGE' occurs, and the 'CLOSED' and 'ENDJOB' at the end), you can delete the two lines to put the channel ID in AO and the timeout in D3. This is not strictly necessary, but if you don't do it, the listing, and the code, will be considerably longer, and possibly run noticably slower.



```
; ---- ALTER LOAD ADDRESS TO MESSAGE3 ----
                    LEA.L     MESSAGE3,A1 ; BASE ADDRESS IN A1
                    MOVE.W    $D0,A2      ; UT_MTEXT VECTOR IN A2
                    JSR       (A2)        ;
; ---- ALTER BRANCH ADDRESS TO WDEF2 ----
                    BRA.S     WDEF2        ; SKIP MESSAGE
;
;---- ALTER LABEL TO .MESSAGE3 ----
;---- ALTER TO REQUIRED CHARACTER COUNT AND MESSAGE ----
;
.MESSAGE3           DC.W      26          ; LENGTH OF MESSAGE
                    DC.B      '              OF THE WINDOW' ;
;
;*************************************************************************
;                                'WDEFINE'
;
;---- DELETE LINES TO ENTER ID IN AO AND TIMEOUT IN D3 ----
;---- ALTER LABEL TO WDEF2 ----
;---- ALTER LOAD ADDRESS TO WDEFINE2 ----
.WDEF2              LEA       WDEFINE2,A1 ; WDEFINE BASE IN A1
                    MOVEQ     #$D,DO      ; #SD_WDEFINE IN DO
; ---- ALTER NEXT TWO LINES TO REQUIRED BORDER ----
                    MOVEQ     #$2,D1      ; RED BORDER
                    MOVE.W    #$2,D2      ; BORDER 2 PIXELS WIDE
                    TRAP      #3          ;
; ---- ALTER BRANCH ADDRESS TO CLS2 ----
                    BRA.S     CLS2        ; SKIP BLOCK
;
; ---- ALTER LABEL TO WDEFINE2 ----
; ---- ALTER BLOCK TO REQUIRED WINDOW ----
.WDEFINE2           DC.W      256         ; 256 PIXELS WIDE
                    DC.W      24          ; 24 PIXELS HIGH
                    DC.W      100         ; X ORIGIN
                    DC.W      50          ; Y ORIGIN
;
; ************************************************************************
;                                  'CLS'
;
; ---- DELETE LINES TO ENTER ID IN AO AND TIMEOUT IN D3 ----
; ---- ALTER LABEL TO .CLS2 ----
.CLS2               MOVEQ     #$20,DO     ; #SD_CLEAR IN DO
                    TRAP      #3          ;
;
; ************************************************************************
;                                  'CSIZE'
;
; ---- DELETE LINES TO ENTER ID IN AO AND TIMEOUT IN D3 ---
                    MOVEQ     #$2D,DO     ; #SD_SETSZ IN DO
; ---- ALTER NEXT TWO LINES FOR REQUIRED CSIZE ----
                    MOVE.W    #$0,D1      ; WIDTH 5 IN 6 PIXEL SPACE
                    MOVE.W    #$1,D2      ; HEIGHT 18 IN 20 PIXEL SPACE
                    TRAP      #3          ;
;
; ************************************************************************
;                                'MESSAGE'
;
; ---- DELETE LINE TO ENTER ID IN AO ----
; ---- ALTER LOAD ADDRESS TO MESSAGE4 ----
                    LEA.L     MESSAGE4,A1 ; BASE ADDRESS IN A1
                    MOVE.W    $D0,A2      ; UT_MTEXT VECTOR IN A2
                    JSR       (A2)        ;
; ---- ALTER BRANCH ADDRESS TO WDEF3 ----
                    BRA.S     WDEF3       ; SKIP MESSAGE
;
; ---- ALTER LABEL TO .MESSAGE4 ----
; ---- ALTER TO REQUIRED CHARACTER COUNT AND MESSAGE ----
.MESSAGE4           DC.W      33          ; LENGTH OF MESSAGE
                    DC.B      '    HERE IS THE WINDOW REDEFINED' ;
;
; ************************************************************************
;                                'WDEFINE'
;
; ---- DELETE LINES TO ENTER ID IN AO AND TIMEOUT IN D3 ----
; ---- ALTER LABEL TO .WDEF3 ----
; ---- ALTER LOAD ADDRESS TO WDEFINE3 ----
.WDEF3              LEA       WDEFINE3,A1 ; WDEFINE BASE IN A1
                    MOVEQ     #$D,DO      ; #SD_WDEFINE IN DO
; ---- ALTER NEXT TWO LINES FOR REQUIRED BORDER ----
                    MOVEQ     #$2,D1      ; RED BORDER
                    MOVE.W    #$4,D2      ; BORDER 4 PIXELS WIDE
                    TRAP      #3          ;
; ---- ALTER BRANCH ADDRESS TO WDEF4 ----
                    BRA.S     WDEF4       ; SKIP BLOCK
;
; ---- ALTER LABEL TO .WDEFINE3 ----
; ---- ALTER BLOCK TO REQUIRED WINDOW ----
.WDEFINE3           DC.W      130         ; 130 PIXELS WIDE
                    DC.W      25          ; 25 PIXELS HIGH
                    DC.W      100         ; X ORIGIN
                    DC.W      75          ; Y ORIGIN
;
; ************************************************************************
;                                'WDEFINE'
;
; ---- DELETE LINES TO ENTER ID IN AO AND TIMEOUT IN D3 ----
; ---- ALTER LABEL TO .WDEF4 ----
; ---- ALTER LOAD ADDRESS TO WDEFINE4 ----
.WDEF4              LEA       WDEFINE4,A1 ; WDEFINE BASE IN A1
                    MOVEQ     #$D,DO      ; #SD_WDEFINE IN DO
```



```
; ---- ALTER NEXT TWO LINES FOR REQUIRED BORDER ----
                    MOVEQ       #$4,D1      ; GREEN BORDER
                    MOVE.W      #$4,D2      ; BORDER 4 PIXELS WIDE
                    TRAP        #3          ;
; ---- ALTER BRANCH ADDRESS TO WDEF5 ----
                    BRA.S       WDEF5       ; SKIP BLOCK
;
; ---- ALTER LABEL TO WDEFINE4 ----
; ---- ALTER BLOCK TO REQUIRED WINDOW ----
.WDEFINE4           DC.W        140         ; 140 PIXELS WIDE
                    DC.W        30          ; 30 PIXELS HIGH
                    DC.W        105         ; X ORIGIN
                    DC.W        80          ; Y ORIGIN
;
; *******************************************************************
;                                'WDEFINE'
;
; ---- DELETE LINES TO ENTER ID IN A0 AND TIMEOUT IN D3 ----
; ---- ALTER LABEL TO WDEF5 ----
; ---- ALTER LOAD ADDRESS TO WDEFINE5 ----
.WDEF5              LEA         WDEFINE5,A1 ; WDEFINE BASE IN A1
                    MOVEQ       #$D,D0      ; #SD_WDEFINE IN D0
; ---- ALTER NEXT TWO LINES FOR REQUIRED BORDER ----
                    MOVEQ       #$6,D1      ; WHITE BORDER
                    MOVE.W      #$4,D2      ; BORDER 4 PIXELS WIDE
                    TRAP        #3          ;
; ---- ALTER BRANCH ADDRESS TO WDEF6 ----
                    BRA.S       WDEF6       ; SKIP BLOCK
;
; ---- ALTER LABEL TO .WDEFINE5 ----
; ---- ALTER BLOCK TO REQUIRED WINDOW ----
.WDEFINE5           DC.W        150         ; 150 PIXELS WIDE
                    DC.W        35          ; 35 PIXELS HIGH
                    DC.W        110         ; X ORIGIN
                    DC.W        85          ; Y ORIGIN
;
; *******************************************************************
;                                'WDEFINE'
;
; ---- DELETE LINES TO ENTER ID IN A0 AND TIMEOUT IN D3 ----
; ---- ALTER LABEL TO .WDEF6 ----
; ---- ALTER LOAD ADDRESS TO WDEFINE6 ----
.WDEF6              LEA         WDEFINE6,A1 ; WDEFINE BASE IN A1
                    MOVEQ       #$D,D0      ; #SD_WDEFINE IN D0
; ---- ALTER NEXT TWO LINES FOR REQUIRED BORDER ----
                    MOVEQ       #$2,D1      ; RED BORDER
                    MOVE.W      #$4,D2      ; BORDER 4 PIXELS WIDE
                    TRAP        #3          ;
; ---- ALTER BRANCH ADDRESS TO WDEF7 ----
                    BRA.S       WDEF7       ; SKIP BLOCK
;
; ---- ALTER LABEL TO WDEFINE6 ----
; ---- ALTER BLOCK TO REQUIRED WINDOW ----
.WDEFINE6           DC.W        160         ; 160 PIXELS WIDE
                    DC.W        40          ; 40 PIXELS HIGH
                    DC.W        115         ; X ORIGIN
                    DC.W        90          ; Y ORIGIN
;
; *******************************************************************
;                                'WDEFINE'
;
; ---- DELETE LINES TO ENTER ID IN A0 AND TIMEOUT IN D3 ----
; ---- ALTER LABEL TO .WDEF7 ----
; ---- ALTER LOAD ADDRESS TO WDEFINE7 ----
.WDEF7              LEA         WDEFINE7,A1 ; WDEFINE BASE IN A1
                    MOVEQ       #$D,D0      ; #SD_WDEFINE IN D0
; ---- ALTER NEXT TWO LINES FOR REQUIRED BORDER ----
                    MOVEQ       #$4,D1      ; GREEN BORDER
                    MOVE.W      #$4,D2      ; BORDER 4 PIXELS WIDE
                    TRAP        #3          ;
; ---- ALTER BRANCH ADDRESS TO WDEF8 ----
                    BRA.S       WDEF8       ; SKIP BLOCK
;
; ---- ALTER LABEL TO WDEFINE7 ----
; ---- ALTER BLOCK TO REQUIRED WINDOW ----
.WDEFINE7           DC.W        170         ; 170 PIXELS WIDE
                    DC.W        45          ; 45 PIXELS HIGH
                    DC.W        120         ; X ORIGIN
                    DC.W        95          ; Y ORIGIN
;
; *******************************************************************
;                                'WDEFINE'
;
; ---- DELETE LINES TO ENTER ID IN A0 AND TIMEOUT IN D3 ----
; ---- ALTER LABEL TO .WDEF8 ----
; ---- ALTER LOAD ADDRESS TO WDEFINE8 ----
.WDEF8              LEA         WDEFINE8,A1 ; WDEFINE BASE IN A1
                    MOVEQ       #$D,D0      ; #SD_WDEFINE IN D0
; ---- ALTER NEXT TWO LINES FOR REQUIRED BORDER ----
                    MOVEQ       #$6,D1      ; WHITE BORDER
                    MOVE.W      #$4,D2      ; BORDER 4 PIXELS WIDE
                    TRAP        #3          ;
; ---- ALTER BRANCH ADDRESS TO WDEF9 ----
                    BRA.S       WDEF9       ; SKIP BLOCK
;
; ---- ALTER LABEL TO .WDEFINE8 ----
; ---- ALTER BLOCK TO REQUIRED WINDOW ----
.WDEFINE8           DC.W        180         ; 180 PIXELS WIDE
```

3. Merge 'WDEFINE' into the file. Alter the first label to .WDEF1, and, in the same line, alter the load address to WDEFINE1. Alter the branch address to CURS. Alter the block label to .WDEFINE, and alter the block to the required window parameters.

4. Merge 'POSITIONCURSOR' into the file. Alter the two lines putting the column and row numbers in D1 and D2.

5. Merge 'MESSAGE' into the file. Alter the load address to 'MESSAGE1'. Alter the branch address to BORD. Alter the message label to MESSAGE1, and alter the character count and message to fit requirements.

6. Merge BORDER into the file. Alter the two lines putting the border colour and width in D1 and D2.

Three times COLOURS

7. Merge three copies of 'COLOURS' into the file. The first can stay as it is to set white paper. Alter the second to put $28 in DO for white strip. Alter the third to put $29 in DO and 2 in D1 for red ink.

8. Merge 'C1s' into the file.

9. Merge 'MESSAGE' into the file. Alter the load address to MESSAGE2. Alter the branch address to MCURS. Alter the message label to MESSAGE2, and alter the character count and message to fit requirements.

10. Merge 'MOVECURSOR' into the file.

11. Merge 'MESSAGE' into the file. Alter the load address to MESSAGE3. Alter the branch address to WDEF2. Alter the message label to MESSAGE3, and alter the character count and message to fit requirements.

12. Merge 'WDEFINE' into the file. Alter the first label to .WDEF2, and, in the same line, alter the load address to WDEFINE2. Alter the branch address to CLS2. Alter the block label to .WDEFINE2, and alter the block to the required window parameters.

13. Merge 'Cls' into the file.

14. Merge 'CSIZE' into the file. Alter the two lines putting width and height in D1 and D2.

15. Merge "MESSAGE' into the file. Alter the load address to MESSAGE4. Alter the branch address to WDEF3. Alter the message label to MESSAGE4, and alter the character count and message to fit requirements.

WDEFINE again

16. Merge eight copies of 'WDEFINE' into the file. Alter the first labels to .WDEF3, WDEF4, etc. . . . up to WDEFA, and, in the same lines, alter the load addresses to WDEFINE3, WDEFINE4, etc. . . . up to WDEFINEA. Alter the branch addresses to branch to the next chunk, so they become WDEF4, WDEF5, etc, except the last branch address, which becomes CLS3. The block labels follow the same pattern as

the load addresses, becoming .WDEFINE3, etc. . . . up to WDEFINEA. And, of course, alter all the blocks to the required window parameters.

17. Merge 'CLS' into the file.

18. Merge 'CSIZE' into the file. Alter the two lines putting width and height in D1 and D2.

19. Merge 'COLOURS' into the file. Alter the two lines putting 'ink' and 'black' into D0 and D1.

20. Merge 'MESSAGE' into the file. Alter the load address to 'MESSAGE5'. Alter the branch address to 'COLS'. Alter the message label to 'MESSAGE5', and alter the character count and message to fit requirements.

21. Merge 'COLOURS' into the file. Alter the two lines putting 'ink' and 'green' into D0 and D1.

22. Merge 'CSIZE' into the file. Alter the two lines putting width and height in D1 and D2.

23. Merge three copies of 'MOVECURSOR' into the file.

24. Merge 'MESSAGE' into the file. Alter the load address to MESSAGE6. Alter the branch address to WMODE. Alter the message label to MESSAGE6, and alter the character count and message to fit requirements.

25. Merge 'WRITEMODE' into the file.

26. Merge six copies of 'BLOCKFILL' into the file. Alter the first labels to .BFILL1, .BFILL2, etc. . . . up to .BFILL6, and, in the same lines, alter the load addresses to

```
                DC.W      50         ; 50 PIXELS HIGH
                DC.W      125        ; X ORIGIN
                DC.W      100        ; Y ORIGIN
;
; *******************************************************************
;                            'WDEFINE'
;
; ---- DELETE LINES TO ENTER ID IN A0 AND TIMEOUT IN D3 ----
; ---- ALTER LABEL TO .WDEF9 ----
; ---- ALTER LOAD ADDRESS TO WDEFINE9 ----
.WDEF9          LEA       WDEFINE9,A1 ; WDEFINE BASE IN A1
                MOVEQ     #$D,D0     ; #SD_WDEFINE IN D0
; ---- ALTER NEXT TWO LINES FOR REQUIRED BORDER ----
                MOVEQ     #$2,D1     ; RED BORDER
                MOVE.W    #$4,D2     ; BORDER 4 PIXELS WIDE
                TRAP      #3         ;
; ---- ALTER BRANCH ADDRESS TO WDEFA ----
                BRA.S     WDEFA      ; SKIP BLOCK
;
; ---- ALTER LABEL TO .WDEFINE9 ----
; ---- ALTER BLOCK TO REQUIRED WINDOW ----
.WDEFINE9       DC.W      190        ; 190 PIXELS WIDE
                DC.W      55         ; 55 PIXELS HIGH
                DC.W      130        ; X ORIGIN
                DC.W      105        ; Y ORIGIN
;
; *******************************************************************
;                            'WDEFINE'
;
; ---- DELETE LINES TO ENTER ID IN A0 AND TIMEOUT IN D3 ----
; ---- ALTER LABEL TO .WDEFA ----
; ---- ALTER LOAD ADDRESS TO WDEFINEA ----
.WDEFA          LEA       WDEFINEA,A1 ; WDEFINE BASE IN A1
                MOVEQ     #$D,D0     ; #SD_WDEFINE IN D0
; ---- ALTER NEXT TWO LINES FOR REQUIRED BORDER ----
                MOVEQ     #$4,D1     ; GREEN BORDER
                MOVE.W    #$4,D2     ; BORDER 4 PIXELS WIDE
                TRAP      #3         ;
; ---- ALTER BRANCH ADDRESS TO CLS3 ----
                BRA.S     CLS3       ; SKIP BLOCK
;
; ---- ALTER LABEL TO .WDEFINEA ----
; ---- ALTER BLOCK TO REQUIRED WINDOW ----
.WDEFINEA       DC.W      200        ; 200 PIXELS WIDE
                DC.W      60         ; 60 PIXELS HIGH
                DC.W      135        ; X ORIGIN
                DC.W      110        ; Y ORIGIN
;
; *******************************************************************
;                            'CLS'
;
; ---- DELETE LINES TO ENTER ID IN A0 AND TIMEOUT IN D3 ----
; ---- ALTER LABEL TO .CLS3 ----
.CLS3           MOVEQ     #$20,D0    ; #SD_CLEAR IN D0
                TRAP      #3         ;
;
; *******************************************************************
;                            'CSIZE'
;
; ---- DELETE LINES TO ENTER ID IN A0 AND TIMEOUT IN D3 ----
                MOVEQ     #$2D,D0    ; #SD_SETSZ IN D0
; ---- ALTER NEXT TWO LINES FOR REQUIRED CSIZE ----
                MOVE.W    #$3,D1     ; WIDTH 10 IN 16 PIXEL SPACE
                MOVE.W    #$1,D2     ; HEIGHT 18 IN 20 PIXEL SPACE
                TRAP      #3         ;
;
; *******************************************************************
;                            'COLOURS'
;
; ---- DELETE LINES TO ENTER ID IN A0 AND TIMEOUT IN D3 ----
; ---- ALTER MOVEQ TO #$29,D0 FOR INK ----
                MOVEQ     #$29,D0    ; #SD_SETIN IN D0 (INK)
; ---- ALTER MOVE.B TO #$0,D1 FOR BLACK ----
                MOVE.B    #$0,D1     ; BLACK INK
                TRAP      #3         ;
;
; *******************************************************************
```

```
; *******************************************************************
;                            'WRITEMODE'
;
; ---- DELETE LINES TO ENTER ID IN A0 AND TIMEOUT IN D3 ----
.WMODE          MOVEQ     #$2C,D0    ; #SETMD IN D0
                MOVE.W    #$FFFF,D1  ; XOR INK INTO BACKGROUND
                TRAP      #3         ;
;
; *******************************************************************
;                            'BLOCKFILL'
; ---- DELETE LINES TO ENTER ID IN A0 AND TIMEOUT IN D3 ----
; ---- ALTER LABEL TO BFILL1 ----
; ---- ALTER LOAD ADDRESS TO BLOCK1 ----
.BFILL1         LEA       BLOCK1,A1  ; BASE ADDRESS OF BLOCK IN A1
                MOVEQ     #$2E,D0    ; #SD_FILL IN D0
; ---- ALTER NEXT LINE TO REQUIRED COLOUR ----
                MOVE.B    #$2,D1     ; RED BLOCK
                TRAP      #3         ;
; ---- ALTER BRANCH ADDRESS TO BFILL2 ----
                BRA.S     BFILL2     ; SKIP BLOCK
;
; ---- ALTER LABEL TO .BLOCK1 ----
; ---- ALTER TO REQUIRED BLOCK ----
.BLOCK1         DC.W      18         ; 18 PIXELS WIDE
                DC.W      20         ; 20 PIXELS HIGH
                DC.W      6          ; X=6
                DC.W      20         ; Y=20
;
; *******************************************************************
;                            'BLOCKFILL'
; ---- DELETE LINES TO ENTER ID IN A0 AND TIMEOUT IN D3 ----
; ---- ALTER LABEL TO BFILL2 ----
; ---- ALTER LOAD ADDRESS TO BLOCK2 ----
.BFILL2         LEA       BLOCK2,A1  ; BASE ADDRESS OF BLOCK IN A1
                MOVEQ     #$2E,D0    ; #SD_FILL IN D0
                MOVE.B    #$4,D1     ; GREEN INK
                TRAP      #3         ;
;
; *******************************************************************
;                            'CSIZE'
;
; ---- DELETE LINES TO ENTER ID IN A0 AND TIMEOUT IN D3 ----
                MOVEQ     #$2D,D0    ; #SD_SETSZ IN D0
; ---- ALTER NEXT TWO LINES FOR REQUIRED CSIZE ----
                MOVE.W    #$0,D1     ; WIDTH 5 IN 6 PIXEL SPACE
                MOVE.W    #$1,D2     ; HEIGHT 18 IN 20 PIXEL SPACE
                TRAP      #3         ;
;
; *******************************************************************
;                            'BLOCKFILL'
; ---- DELETE LINES TO ENTER ID IN A0 AND TIMEOUT IN D3 ----
; ---- ALTER LABEL TO BFILL3 ----
; ---- ALTER LOAD ADDRESS TO BLOCK3 ----
.BFILL3         LEA       BLOCK3,A1  ; BASE ADDRESS OF BLOCK IN A1
                MOVEQ     #$2E,D0    ; #SD_FILL IN D0
; ---- ALTER NEXT LINE TO REQUIRED COLOUR ----
                MOVE.B    #$6,D1     ; WHITE BLOCK
                TRAP      #3         ;
; ---- ALTER BRANCH ADDRESS TO BFILL4 ----
                BRA.S     BFILL4     ; SKIP BLOCK
;
; ---- ALTER LABEL TO .BLOCK3 ----
; ---- ALTER TO REQUIRED BLOCK ----
.BLOCK3         DC.W      24         ; 24 PIXELS WIDE
                DC.W      20         ; 20 PIXELS HIGH
                DC.W      54         ; X=54
                DC.W      20         ; Y=20
;
; *******************************************************************
;                            'BLOCKFILL'
;
; ---- DELETE LINES TO ENTER ID IN A0 AND TIMEOUT IN D3 ----
; ---- ALTER LABEL TO BFILL4 ----
; ---- ALTER LOAD ADDRESS TO BLOCK4 ----
.BFILL4         LEA       BLOCK4,A1  ; BASE ADDRESS OF BLOCK IN A1
                MOVEQ     #$2E,D0    ; #SD_FILL IN D0
; ---- ALTER NEXT LINE TO REQUIRED COLOUR ----
                MOVE.B    #$2,D1     ; RED BLOCK
                TRAP      #3         ;
```

BLOCK1, BLOCK2, etc. . . . up to BLOCK6. Alter the branch addresses to branch to the next chunk, so they become BFILL2, BFILL3, etc. . . . except the last branch address, which becomes CLOSE. The block labels follow the same pattern as the load addresses, becoming .BLOCK1, .BLOCK2, etc. . . . up to .BLOCK6. And, of course, alter the blocks to the required parameters.

27. Merge 'CLOSE' into the file.
28. Merge 'ENDJOB' into the file.

You could, of course, simply copy listing one. But that would rather defeat the point of the exercise. It is much better to merge the required chunks of listing into one file, and make the appropriate modifications. You would then come to see just how easy it is to put together. To show yourself that you have really understood the system, you could make your own changes to the program – I'm sure it can be greatly improved! Perhaps you could try out the few commands that do not come into this listing, for example, put in some underlining, or (in 8 colour mode) make some words flash, or recolour a whole window at some stage. You could use some of this to embellish the program put together in part 2 of this series.

Remember, though, that this is not the complete armoury of screen control commands – just the easiest ones. Next time, we shall look at those other, not so easy, screen commands, involving floating point numbers. Happy coding!

```
;                                  'MESSAGE'
;
; ---- DELETE LINE TO ENTER ID IN A0 ----
; ---- ALTER LOAD ADDRESS TO MESSAGE5 ----
                     LEA.L      MESSAGE5,A1 ; BASE ADDRESS IN A1
                     MOVE.W     $D0,A2      ; UT_MTEXT VECTOR IN A2
                     JSR        (A2)        ;
; ---- ALTER BRANCH ADDRESS TO COLS
                     BRA.S      COLS        ; SKIP MESSAGE
;
; ---- ALTER LABEL TO .MESSAGE5 ----
; ---- ALTER TO REQUIRED CHARACTER COUNT AND MESSAGE ----
.MESSAGE5            DC.W       8           ; LENGTH OF MESSAGE
                     DC.B       '    OOPS!' ;
;
; ***********************************************************************
;                                  'COLOURS'
;
; ---- DELETE LINES TO ENTER ID IN A0 AND TIMEOUT IN D3 ----
; ---- ALTER MOVEQ TO #$29,D0 FOR INK ----
.COLS                MOVEQ      #$29,D0     ; #SD_SETIN IN D0 (INK)
; ---- ALTER MOVE.B TO #$4,D1 FOR GREEN ----
; ---- ALTER NEXT LINE TO REQUIRED COLOUR ----
                     MOVE.B     #$4,D1      ; GREEN BLOCK
                     TRAP       #3          ;
; ---- ALTER BRANCH ADDRESS TO BFILL3 ----
                     BRA.S      BFILL3      ; SKIP BLOCK
;
; ---- ALTER LABEL TO .BLOCK2 ----
; ---- ALTER TO REQUIRED BLOCK ----
.BLOCK2              DC.W       18          ; 18 PIXELS WIDE
                     DC.W       20          ; 20 PIXELS HIGH
                     DC.W       30          ; X=30
                     DC.W       20          ; Y=20
;
; ***********************************************************************
;                                  'MOVECURSOR'
;
; ---- DELETE LINES TO ENTER ID IN A0 AND TIMEOUT IN D3 ----
                     MOVEQ      #$12,D0     ; #SD_NL IN D0 (CR,LF)
                     TRAP       #3          ;
;
; ***********************************************************************
;                                  'MOVECURSOR'
;
; ---- DELETE LINES TO ENTER ID IN A0 AND TIMEOUT IN D3 ----
                     MOVEQ      #$12,D0     ; #SD_NL IN D0 (CR,LF)
                     TRAP       #3          ;
;
; ***********************************************************************
;                                  'MOVECURSOR'
;
; ---- DELETE LINES TO ENTER ID IN A0 AND TIMEOUT IN D3 ----
                     MOVEQ      #$12,D0     ; #SD_NL IN D0 (CR,LF)
                     TRAP       #3          ;
;
; ***********************************************************************
;                                  'MESSAGE'
;
; ---- DELETE LINE TO ENTER ID IN A0 ----
; ---- ALTER LOAD ADDRESS TO MESSAGE6 ----
                     LEA.L      MESSAGE6,A1 ; BASE ADDRESS IN A1
                     MOVE.W     $D0,A2      ; UT_MTEXT VECTOR IN A2
                     JSR        (A2)        ;
; ---- ALTER BRANCH ADDRESS TO WMODE ----
                     BRA.S      WMODE       ; SKIP MESSAGE
;
; ---- ALTER LABEL TO .MESSAGE6 ----
; ---- ALTER TO REQUIRED CHARACTER COUNT AND MESSAGE ----
.MESSAGE6            DC.W       30          ; LENGTH OF MESSAGE
                     DC.B       ' AND ALL WITH ONLY ONE WINDOW!' ;
;
; ---- ALTER BRANCH ADDRESS TO BFILL5 ----
                     BRA.S      BFILL5      ; SKIP BLOCK
;
; ---- ALTER LABEL TO .BLOCK4 ----
; ---- ALTER TO REQUIRED BLOCK ----
.BLOCK4              DC.W       24          ; 24 PIXELS WIDE
                     DC.W       20          ; 20 PIXELS HIGH
                     DC.W       84          ; X=84
                     DC.W       20          ; Y=20
;
; ***********************************************************************
;                                  'BLOCKFILL'
; ---- DELETE LINES TO ENTER ID IN A0 AND TIMEOUT IN D3 ----
; ---- ALTER LABEL TO BFILL5 ----
; ---- ALTER LOAD ADDRESS TO BLOCK5 ----
.BFILL5              LEA        BLOCK5,A1   ; BASE ADDRESS OF BLOCK IN A1
                     MOVEQ      #$2E,D0     ; #SD_FILL IN D0
; ---- ALTER NEXT LINE TO REQUIRED COLOUR ----
                     MOVE.B     #$4,D1      ; GREEN BLOCK
                     TRAP       #3          ;
; ---- ALTER BRANCH ADDRESS TO BFILL6 ----
                     BRA.S      BFILL6      ; SKIP BLOCK
;
; ---- ALTER LABEL TO .BLOCK5 ----
; ---- ALTER TO REQUIRED BLOCK ----
.BLOCK5              DC.W       18          ; 18 PIXELS WIDE
                     DC.W       20          ; 20 PIXELS HIGH
                     DC.W       114         ; X=114
                     DC.W       20          ; Y=20
;
; ***********************************************************************
;                                  'BLOCKFILL'
; ---- DELETE LINES TO ENTER ID IN A0 AND TIMEOUT IN D3 ----
; ---- ALTER LABEL TO BFILL6 ----
; ---- ALTER LOAD ADDRESS TO BLOCK6 ----
.BFILL6              LEA        BLOCK6,A1   ; BASE ADDRESS OF BLOCK IN A1
                     MOVEQ      #$2E,D0     ; #SD_FILL IN D0
; ---- ALTER NEXT LINE TO REQUIRED COLOUR ----
                     MOVE.B     #$2,D1      ; RED BLOCK
                     TRAP       #3          ;
; ---- ALTER BRANCH ADDRESS TO CLOSE ----
                     BRA.S      CLOSE       ; SKIP BLOCK
;
; ---- ALTER LABEL TO .BLOCK6 ----
.BLOCK6              DC.W       42          ; 42 PIXELS WIDE
                     DC.W       20          ; 20 PIXELS HIGH
                     DC.W       138         ; X=138
                     DC.W       20          ; Y=20
;
; ***********************************************************************
;                                  'CLOSE'
;
; ---- DELETE LINE TO ENTER ID IN A0 ----
.CLOSE               MOVEQ      #2,D0       ; #IO_CLOSE IN D0
                     TRAP       #2          ;
                     BRA.S      END_JOB     ;
;
; ***********************************************************************
;                                  'ENDJOB'
;
.JOB_END             MOVE.W     $CA,A2 ; UT_ERRO in A2
                     JSR        (A2)   ;
;
.END_JOB             MOVEQ      #5,D0       ; #MT_FRJOB IN D0
                     MOVEQ      #-1,D1      ; ID OF THIS JOB IN D1
                     TRAP       #1          ;
;
; ***********************************************************************
```

QL HARDWARE

Ingenieurbüro
Wilfried Krummrey
Postfach 47 04 41
D-1000 Berlin 47

QL SOFTWARE

QDesign

The first graphic programme, which runs completely under the Wman by QJump, it is configurable with the QJump standard Config. It is shipped with the **Pointer Environment** by QJump and the **Menu Extensions** by J.Merz. If QPac 2 exists, it uses the Button Frame. ***QDesign*** can work with **Vectorfonts. Extensive block operations** like reducing, enlarging, mirroring, rotating put it into the ideal tool to treat every kind of graphics. Pattern exist in the standard QJump format – therefore every sprite can be used as pattern. ***QDesign*** includes an **universal printer driver** for 9–, 18–, and 24 pin printer. Professional Publisher pages and Atari screens can be loaded, treated and saved directly. The maximal resolution is 3200*4800 pixels. The Extended Mode of the "Atari–QL" is supported, the full screen is used. Now there is a **Laserprinter version available** (HP LaserJet DeskJet, InkJet and compatible). Obviously this version supports the pin printer.

QDesign *english, disk 3.5", **(order no. 250014) DM 111,– / £37***
QDesign Laser *english, disk 3.5", **(order no. 250016) DM 129,– / £43***

VecEdit

the program to edit BGI vector fonts, as they are used in the program ***QDesign***. There **are various edit modes** available (insert, move and delete ending points of a character) and an undo–function. Single characters can be copied, respectively whole character areas can be inserted from other fonts, **complete fonts created** or transformed. The Pointer Environment by QJump is shipped with it. A mouse or the keyboard can be used for control.

VecEdit *english, disk 3.5", **(order no. 250021) DM 54,– / £18***

SYSTEM

The system extension for your expanded QL or Atari–QL. Now there are **system attributes for the floppy– and winchester drives**, i.e. single files can be made **write protected** or **hidden** or **useable for specified user.** A write protection of a disk can be detected. ***SYSTEM*** works with all floppy and winchester controller, especially in your NET. ***SYSTEM* is the ideal completion of the TK II** from Tony Tebby. ADIR, ASTAT (alphabetical sorted directories), a lot of useful commands, i.e. **fully syntax compatible with TK II.** As every of our software it comes with a printed guide.

SYSTEM *english, disk 3.5" **(order no. 25331) DM 94,– / £31.30***

FOR MORE INFORMATION: ORDER OUR FREE OF CHARGE PRICE–LIST!
Export prices excl. VAT. E&OE. Please add DM 10,– for postage.
Payment: Cheques, Eurocheques or postal orders (DM). Mail order only.

☆ sinclair QL ☆

E.E.C. PROVIDE

A COMPLETE COMPUTER SYSTEM FOR £125!

The expandable system for businesses, beginners and experts.
COMPLETE QL's - JS ROM - TESTED PRIOR TO DELIVERY - 6 MONTHS GUARANTEE

Package includes PSU, TV Lead, QL Software 2.35: Quill - Wordprocessor, Abacus - Spreadsheet Archive - Database & Easel - Business Graphics. Comprehensive instruction manual. One years free membership to QUANTA - the independant user group who provide help, newsletters & a library of 400+ programs - mostly free.

Backup units are available which comprise just the QL computer and power supply:
Backup JM QL - £85 -also, enquire about our unique exchange scheme. Computers accepted include: QL, Spectrum - 16k, 48k, 128k, +2, +3, Commodore, BBC etc

UNIVERSAL DISK DRIVES (all our drives are brand new product)

Universal 3.5" 1Mb (unformatted) single - £70, double - £130
Universal 3.5" 2Mb (unformatted) single - £90, double - £170
THE UNIVERSAL DRIVE MAY BE FITTED TO MANY TYPES OF COMPUTER BY ATTACHING A CABLE AND SELECTING THE CORRECT DIP SWITCH SETTINGS. I.E. QL, SPECTRUM, AMIGA, PC)
Uncased 3.5" 1Mb (unformatted) - £39 each
Uncased 3.5" 2Mb (unformatted) - £59 each

QL to PC KEYBOARD ENHANCEMENTS

QL/PC 102 keyboards - £25 UK version, XT/AT switchable - £35
QL/PC keyboard interface - £75 External fitting box & cable - £17

PRINTERS

Seikosha SP1900+ 9 pin dot matrix, up to 192 cps, FX850 emulation - £120
Canon BJ10ex bubblejet, very high quality (360dpi), A4 size, - £259
We can supply almost any make/model printer - please ask for details.

MONITORS - colour & mono - please call for details of our very latest offer
QL / Spectrum spares & repairs - please call for details

EEC Ltd
18-21 MISBOURNE HOUSE
CHILTERN HILL
CHALFONT ST.PETER
SL9 9UE
TEL: 0753 888866

CWO, Access, VISA et cetera
Cheques - allow 10 days
Courier - £9 Postage - £3
Product subject to availability, E&OE.

FAX: 0753-887149

CARE ELECTRONICS

QPAC II by Tony Tebby

MAKES YOUR QL TRULY MULTI - TASKING

QPAC II Main menu windows adjust automatically to size. Files, directory, view, print, delete, backup, jobs, pick, Rjob, sort, channels, things exec, wake, buttons, Hotkey, Hotjobs. Fully multitasking, allows many programs to run at once. Requires min of 256k expanded memory .... **SPECIAL OFFER ONLY £39.48b**

Upgrade QRAM to QPAC II return master . ... @29.61b

PLEASE SEND SAE FOR A COPY OF THE QPAC II REVIEW

QPOWER REGULATOR

Switch mode power supply, QL runs cool, stops lock ups (due to voltage drop). Helps filter noise out. Simple to fit (no soldering) £25.85c

QL SUPERTOOL KIT II by Tony Tebby

Over 118 Commands: Full Screen Editor, Key Define Print Using, Last Line Recall, Altkey, Job Control, File Handling, Default Directories, Extended Network.
16k Eprom Cartridge Version .... @£23.50d
Configurable Version of Microdrive .... @£23.50d

ZITASOFT SOFTWARE by Steve Jones

LOCKSMITHE copies M/DRIVE - M/DRIVE .... @£14.10c
4MATTER + LOCKSMITHE copies M/DRIVE - DISC .... @£23.50c
SIDEWINDER - High resolution printer driver prints full screens or parts of screens from postage stamp size to large banners. Prints sideways, invert, scale, mirror, text insertion .... @£20.21c
SIDEWINDER PLUS - (for expanded QL's) includes all the features of above, PLUS multiple label printing, desktop publishing files and printer drivers for The LC10 JX80 24 pin and colour printers (Please state 3.5" disc or M/Drive) .... @£23.97c
Upgrade to Sidewinder Plus .... @£8.46c

SOFTWARE 87 (State MDV or Disc)

TEXT87 Plus 4 .... @ £79.00b
FOUNTED 89 .... @ £15.00
FOUNTED 88 .... @ £25.00c
248B PRINTER DRIVER .... @ £15.00c
Upgrade to Text 87 Plus 4 from V3.00. Return master £39.00b
Upgrade to Text 87 Plus 4 from all other versions. Return Master £49.00b
Send SAE for further info on above.

MIRACLE SYSTEMS PRODUCTS

0K Disc Interface (inc Tool Kit II) .... @ £99.64b
QL Centronics Printer interface .... @ £29.61d

QL HARDWARE

QLKeyboard Membrane .... @ £11.75d
Care Eprom Cartridges each .... @ £6.11c
ULA CHip ZX8301 .... @ £15.98c

READY MADE LEADS

RGB QL TO PHONO £6.11c
RGB 8-6 PIN £7.52c
RGB 8-7 PIN HITACHI £7.52c
RGB 8-7 PIN FERGUSON £7.52c
RGB 8 PIN TO SCART £11.75c
RGB 8PIN TO CM8833 MK II £11.75c
6WAY PCC TO 25WAY D £9.87c

TONY TEBBY SOFTWARE (QJUMP)

QTYP Type/Spell Checker .... @ £30.55d
QPAC 1-Desk top accessories, calendar, alarm, calculator, typewriter, digital clock sysmon .... @ £22.56d

CARE ELECTRONICS
DEPT QL, 15 HOLLAND GDNS
GARSTON, WATFORD,
HERTS, WD2 6JN.
Tel: 0923-894064
Fax: 0923-672102

Please add carriage
a=£6.11 b=£3.76 c=no charge d=£2.35

HOW TO ORDER

By Post. Enclose your Cheque/PO made payable to CARE Electronics or use ACCESS/VISA. Allow 7 days for delivery

QUANTA

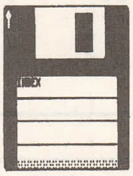
Independent QL/Thor Users Group

Worldwide Membership is by subscription only, and offers the following benefits:
Monthly Newsletter – up to 40 pages
Massive Software Library – mostly FREE!
Free Helpline and Workshops
Regional Sub-Groups. One near you?
Advice on Software & Hardware problems
Discounts from most major suppliers
Subscription just £14 for UK members
Overseas subscription £17

Barclaycard: Visa: Access: Mastercard

* Now in our EIGHTH successful year *

Further details from the Membership Secretary



Bill Newell
213 Manor Road
Benfleet
Essex. SS7 4JD
Tel (0268) 754 407

We are at present having difficulties in exporting copies of Sinclair QL World. So that you do not miss any copies, we invite you to take advantage of the following offer:

Why not buy yourself Knock down subscription to your favourite magazine!

Subscribe for one year and you will receive 15 issues for the price of 12.

That's 3 issues Free!!

Subscribe for two years and and get 3 FREE issues AND a discount of £4.00!!

Fill in the coupon below and send it with your remittance to: MSM Subscriptions Department, Lazahold Ltd., P.O. Box 10, Roper Street, Pallion Industrial Estate, Sunderland SR4 6SN. (the first issue of a new subscription to be delivered will be one or two issues after the one you placed your order in.)

Please send me my 12/24 month subscription to Sinclair QL World. at the special discount rate of £________ I will receive 3 free issues before my subscription becomes due for renewal.

I enclose my cheque/money order for £ ______ made payable to MCPC Ltd or debit my Access/ Visa card. No.______________________ Expiry date ______________________

Name ______________________

Address ______________________

____

______________________(Please enter postcode to ensure prompt delivery)

Signed. ______________________ Date. ______________________ QL 0292

EUROPE	£32.90 12 mths	REST OF	£40.90 12 mths
	£61.80 24 mths	WORLD	£77.80 24 mths

Overseas rates include Airmail service

INSTANT

INSTANT HARDWARE

Care Electronics
0923 672102
Peripherals and accessories.

Thor International
010 4533 930305
010 4533 937544

Digital Precision
081-527 5493
Discs, interfaces, drives, keyboards, RAM expansions.

EEC Ltd
0753 888866
QLs, peripherals, accessories

Miracle Systems
0904 423986
Trump Card interface, modems, hard disc drive.

Strong Computer Systems
0267 231246

Sector Software
0772 454328
Peripherals.

S.P.E.M
Via Aosta 16, 10154 Torino, Italy.
+39 11 857924 Fax +39 11280009
QL disk interfaces, ram, eprom holders.

TF Services
071 724 9053
Repairs, spares, peripherals.

TK Computerware
0303 812801
Peripherals.

INSTANT SERVICES

Adman Services
0952 255895

Joe Atkinson
36 Ranelagh Road, London W5 5RJ
Roms, mdvs and all spares.

Suredata
081-902 5218
Repairs.

QL World
(reader services)
Subscriptions
091 510 8787 (Lazerhold)
Microdrive Exchange
0772 454328 (Sector)
Recent back issues : write to editor

Quanta
User Group, support.
Hon. secretary, Phil Borman.
0933 460690

Xchange User Group
Freepost
Beckenham
Kent BR3 2BR

INSTANT SOFTWARE

Athene Consultants
0705 511439

ARK Distribution
0983 79496
Archivist, Master Spy and other business and utility software.

Call Us What You Like
55 Greenan, Shaws Road, Belfast BT11 8LX. No telephone. Superspy.

Care Electronics
0923 672102
Tony Tebby software. Miracle hardware, peripherals, utilities.

CGH Services
0559384 574
Adventures. QLAF, DIY Toolkit.

Compware
0270 582301
Task Swopper, Mega Toolbox, Expert, System Shell, Speedscreen and others.

COWO Electronic
010 41 45 211478 (Switzerland)
QTop, Atari QL emulator, Thor support

Digital Precision
081-527 5493
Turbo Basic compiler, Desktop Publisher, Editor, Eye-Q Graphics. Sprite generator, Super Forth, Better Basic, Professional and Super Astrologer, Media Manager, Supercharge, Blocklands, Droidzone, Aracadia, Reversi, Backgammon, Master Bridge Tutor, C-M Emulator, C compiler, Lightning speed enhancer.

DJW Software
0256 881701
Homebanker

Dilwyn Jones Computing
0248 354023
Home Budget, Vision Mixer, QL Genealogist, Textidy, Spellbound and others.

Eidersoft
089283 2552
Games utilities, Ice. Pyramide.

ImaQLate Software
42 Albion Street, Broadstairs, Kent CT10 1NE.
Biograph.

Jochen Merz Software
010 49 203501274
QL/Atari Emulators, utility and games software.

Lear Data Systems
6 Southview Green, Bentley
Ipswich, Suffolk IP9 2DR
Qtris

Liberation Software
081 546 7795
Basic compiler (QLib) and utilities.

QFILE
Apartado 2110 p-1103 Lisboa Codex, MS-Link and Discopy. Portugal.

QL Supersoft
0256 475992
Spell, Numbers, Superdraw.

Qubbesoft PD
38 Brunwin Road, Rayne, Braintree, Essex CM7 5BU
Public domain software.

Sector Software
0772 454328
Flashback and other utilities.

SD Microsystems
0462 422897
Business software including General Ledger. Small Traders' Pack/Invoicer and Stock Accounting.

Software 87
33 Savernake Road, London NW3 2JU
Text 87 and updates.

Talent Software
0303 813883
Utilities, games.

TK Computerware
0303 812801
Add-ons, spares, software.

WD Software
0534 81392

DJC

Dilwyn Jones Computing

41 Bro Emrys, Tal-y-Bont,
Bangor, Gwynedd LL57 3YT
Tel: Bangor (0248) 354023

PAGE DESIGNER 2 PLUS £40.00

WRITTEN BY DILWYN JONES

QL desktop publishing made simple with this long awaited revision of the Page Designer 2 program, previously sold by Sector Software.

Mix standard text, high resolution text, and graphics on documents up to 9 pages long (depending on available memory size). Use it to make posters, leaflets, booklets, newsletters, and so on. Pages can be saved, loaded, printed, copied, etc. and can be in up to 4 colours, if you have a colour printer (driver for Star Colour LC10 included).

* Very easy to use via simple on screen menus
* Import wordprocessor text into columns across pages.
* Text direct from either Quill "_doc" or plain text files
* Cut and Paste for moving things around and making clipart
* Clip Art pictures supplied on separate disks
* Large range of fonts supplied
* Improved high resolution text - larger fonts now possible
* Now Gold Card compatible
* New, improved manual
* Many new and improved features
* Retains popular features of previous versions
* Can still use files from older versions
* Conversion routines for fonts & pages from other programs
* Flexible, reconfigurable printer driver
* Configurable startup - page size, fonts, etc
* Graphics section - lines, patterns, circles, etc
* Load screens from art programs like The Painter

PAGE DESIGNER 2 PLUS requires a minimum of 512k RAM to operate. It is available on floppy disk (3.5" or 5.25") only. PAGE DESIGNER 2 PLUS is much enhanced compared with the original, but still costs only £40.00!

UPGRADES - £20.00. If you have an older version of Page Designer 2 and would like to upgrade it to the "2 PLUS" version, please send us the original master disk or proof of purchase from Sector Software (it will be returned to you) with an upgrade fee of £20.00. We will send you the new version on new disks with a new manual.

PAGE DESIGNER 2 PLUS IS SCHEDULED FOR RELEASE IN MID-JANUARY, BUT YOU MAY PREFER TO CONTACT US BEFORE ORDERING IN CASE OF DEVELOPMENT DELAYS. WE WILL TAKE YOUR DETAILS AND CONTACT YOU WHEN IT IS READY IN THIS CASE. PLEASE CONTACT US IF YOU WOULD LIKE AN INFORMATION SHEET CONTAINING FURTHER INFORMATION AND A SAMPLE PRINTOUT FROM THIS SUPER PROGRAM.

BIBLE TEXT DISKS £20.00

King James Bible text available on a set of disks in either Quill "_doc" file format, or in plain text (editor) format. *Please specify which format you would prefer when ordering.*

LIBERATION SOFTWARE PRODUCTS

QLIBERATOR V3.35 £50.00

This great QL SuperBASIC compiler is now available from DJC. QLiberator is compatible with the full range of SuperBASIC commands and functions, and Minerva compatible (even with 2 screen mode). Produce faster, multitasking code from your BASIC programs. Version 3.35 can produce ROMable, re-entrant code, libraries of compiled procedures and functions, and can even produce toolkit routine to expand the BASIC interpreter! Support for error trapping in your programs and even the WHEN VARiable construct where implemented. Requires minimum of 256k memory.

BUDGET QLIBERATOR £25.00

Budget price, not a budget specification! This is a full function, easy to use SuperBASIC compiler that converts slow QL BASIC programs into fast multitasking jobs without fuss. COMPACT DESIGN MAKES IT SUITABLE FOR USE ON AN UNEXPANDED QL. Full language compatibility and useable with commercial toolkits, support for pipes, error trapping, etc.

QLOAD 7 QREF £15.00

QLOAD is a utility for speeding up the loading and saving of SuperBASIC programs on the QL. Even long BASIC programs load in seconds rather than the minutes it could normally take! Even allows the use of machine code extensions in a program. QREF is a BASIC cross-reference utility, a boon for BASIC programmers. Prints sorted lists of variables, procedures, functions etc in a program. Wild card selection of names, and a FIND function for the BASIC editor (or Toolkit 2's ED). We use QLOAD and QREF and would not be without them when programming in QL SuperBASIC.

IMAGE PROCESSOR V2 £15.00

WRITTEN BY DILWYN JONES

Screen picture image processing program. Features computerised Image Enhancement, Edge Detection, Screen Mode Conversion, Recolouring, Decolouring, Pixel Editing, Brush for hand alterations or simple drawing, very useful for tidying up digitised pictures or other applications where you need to alter and tidy up screen pictures. All facilities in Version 2 now work in both mode 4 and mode 8.

UPGRADE FROM OLDER VERSIONS £10.00 - send us your master disk or proof of purchase (it will be returned to you) with upgrade fee of £10.00 and we will supply you with a new disk and manual.

SEE OUR MAIN ADVERT ELSEWHERE IN THIS ISSUE FOR OUR RANGE OF QL PRODUCTS AND POSTAGE RATES OVERSEAS (DJC SOFTWARE IS SENT POST FREE TO UK ADDRESSES) AND PAYMENT METHODS, INCLUDING CREDIT CARDS ACCEPTED.

THE FIRST PC TO BREAK THE SOUND BARRIER.

We gave the Amstrad PC5286™ all the best features you need for serious business use and great games entertainment.

Then we gave it features which were unheard of.

As you'd expect, it has state-of-the-art graphics with a 14″ high quality VGA monitor which will display up to 256 colours from a palette of over a quarter of a million.

Terrific for business applications but all that colour is wasted on games if your sound system comes from the dark ages.

So we gave it a sound system that makes it unique.

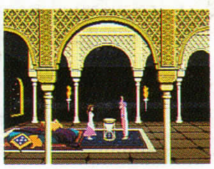
Its fully 'Ad-Lib' compatible soundcard with its twin external speakers, brings you closer to the action bringing a whole new dimension to the PC.

BUILT FOR BUSINESS

We gave the PC5286™ a massive 1 Mb RAM, 40 Mb hard drive and 1.44 Mb 3.5″ floppy disc drive.

LINKS

F-15 STRIKE EAGLE II


PRINCE OF PERSIA

And when it comes to using all that storage capacity it has a fast 80286 chip running at 16 Mhz so you'll be able to get through the business of the day with enough time left to play around.

THE COMPLETE SYSTEM

If golf isn't your game the Amstrad PC5286™ games pack comes with two other great games. Together with fully featured keyboard, mouse, DOS 3.3 with user-friendly graphical interface and a joystick.

So at £899 + Vat (£1056.33 Inc Vat)*, it has to be a sound business proposition.

AMSTRAD®

FOR FURTHER DETAILS RING **0277 262326** OR FILL IN THE COUPON BELOW

I like the sound of the PC5286™ Please tell me more. SQLW2

Name ____________________

Address ____________________

____

Postcode ____________________

Coupon information may be kept on a database. This may be passed to your Amstrad dealer, if you object please tick box □

Send to: Amstrad plc, PO Box 462, Brentwood, Essex CM14 4EF. Tel: (0277) 262326.

R.R.P's correct at 1/10/91, may change without notice. © Amstrad plc October 1991. All rights reserved.

Available from selected branches of Allders, Dixons, John Lewis, Rumbelows, Rymans, Wildings and over 3,000 registered Amstrad computer dealers.