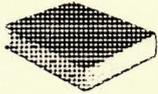


QReview

£2.00

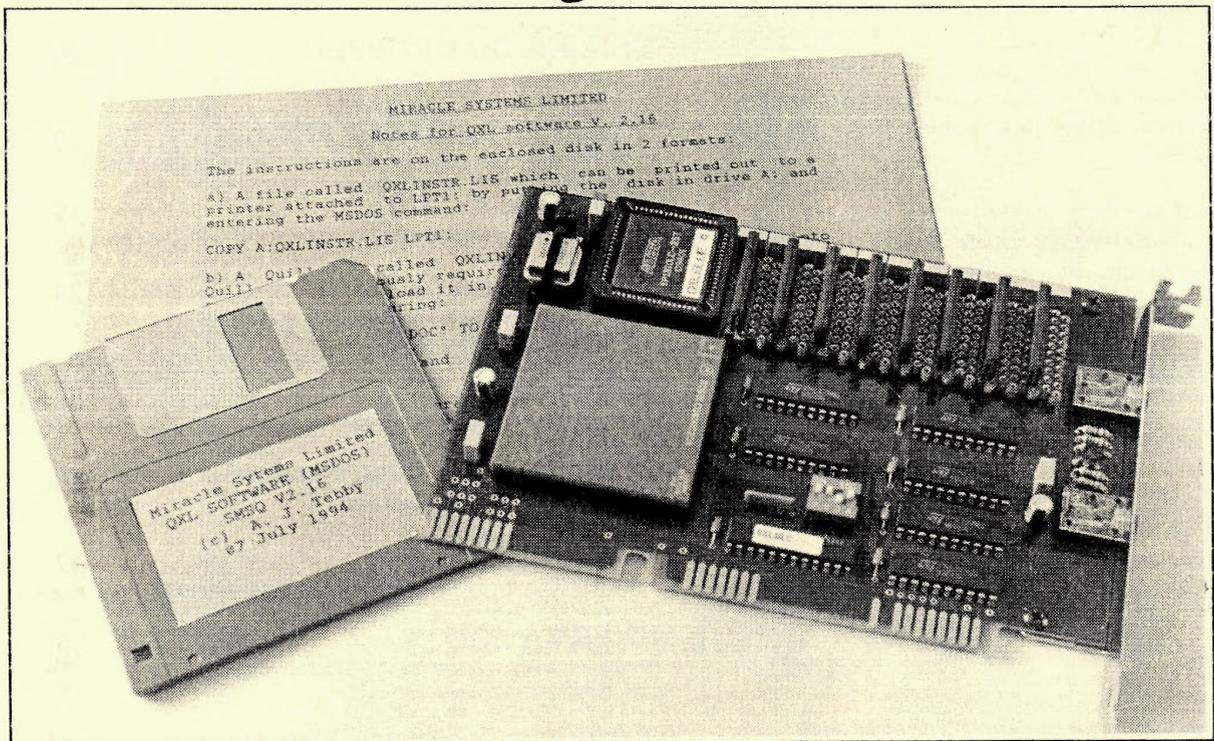


The reference guide to QDOS and its derivatives

Volume 2 Issue 1

September 1994

Miracle Systems QXL



COMPETITION
WIN UP TO 7 DJC
PROGRAMS WORTH £80

HELPDESK
BEFORE YOU

F U N C T I O N A L I T Y

DEA

SVENSKA QL-GRUPPEN

PLUS MUCH MORE

QReview

(ISSN 1351-1343)

is published by

© Quo Vadis Design

57 Shaftesbury Road

Romford Essex

RM1 2QJ

Tel: 0708-755 759

Editor: Bruce Nicholls

QReview is published, roughly, quarterly and is available by subscription for four issues, at a cost of £8 in the UK, £10 in Europe and £15 any other country. Back issues are available from the publisher. Please make payments payable to 'Quo Vadis Design'. Payment in UK pounds Sterling currency only, cheques (drawn on a UK branch of a Bank or Building Society), Postal Orders, International Postal Orders and Eurocheques are all accepted. Prices quoted above are valid until 31/12/94.

Editorial enquiries should be sent to the address given above. We welcome letters, comments, suggestions and articles from our readers. The submission deadline for the next issue is the 10th of December 1994.

Articles submitted for publication should be on a 3.5 DD or HD disk, or microdrive, in either Quill, Perfection, Text87 or Editor format and be accompanied by a printed copy of the article. Saved Screen dumps produced with the Sbytes command are also accepted together with screens saved by most QL drawing packages (Please indicate which package you used to produce the graphics).

Advertising rates are £25 for a full A4 page, £15 for half an A4 page and £10 for a quarter A4 page. Discounts are given for multiple page adverts.

QReview is independent and not affiliated to any vendor of hardware, software or services.

While every care has been taken with the compilation of this publication, we cannot accept responsibility for any errors or omissions. If any error or omission is found, we will endeavour to publish clarification in the next available issue.

All rights reserved. No part of this publication may be reproduced in any form or by any means without the prior permission in writing of the publishers. All trademarks are acknowledged as the property of their respective owners.

The opinions expressed are those of the authors and are not necessarily those of the publishers.

QReview is produced on a QXL using a HP Deskjet 500 printer. The software used to produce the magazine is Text87Plus4 Publisher (Software87), Linedesign (PROGS), Image Processor (DJC) and Screen Snatcher (DJC).

CONTENTS

NEWS	3
The latest news from around the world.	
COMPETITION	5
Your chance to win up to 7 DJC programs in this free to enter competition.	
BEYOND QUILL	6
Is it worthwhile to move on from Quill to another wordprocessor? Geof Wicks finds out in this series.	
REVIEW - FDU	9
Need to recover a file from a corrupt disk? Norman Dunbar explains all with this utility from Ergon Development.	
SVENSKA QL-GRUPPEN	14
Sone Thorén talks about the Swedish QL Group and the QL in Sweden.	
HARDWARE REVIEW - QXL	16
Bruce Nicholls talks about Miracle Systems card to transform a PC into a QL compatible.	
HELPDESK	19
Hardware and software queries/problems.	
REVIEW - DEA	23
An Intelligent disassembler? Norman Dunbar assembles the answer with this product from Ergon Development.	
INTERRUPT	27
Norman Dunbar upsets the purists with the help of BP_LET.	
PICTURES AT A WORKSHOP	30
QL personalities at a Workshop. To be continued	
FUNCTIONality	32
Dilwyn Jones continues this regular section on SuperBASIC Functions and Procedures.	
MINERVA BOOT	35
In this two part series Andrew Pratt utilises MINERVA's ROM enhancements to create a powerful BOOT program.	
REVIEW - SCREEN COMPRESSION	39
John Southern checks out this compression utility from DJC.	
ADVERTS	
QUANTA	8
Dilwyn Jones Computing	12-13
QUBBESoft P/D	18
Miracle Systems	20-22
TF Services	26
S.J.P.D.	31
IQLR	34
Jochen Merz	40



NEWS



MIRACLE SYSTEMS ON THE MOVE

Miracle Systems have now moved from their York address to Bristol. Their new address and telephone number is :-

20 Mow Barton
Yate
Bristol
BS17 5NF
UK

(0454) 883602

DJC - RELEASE P.A.Y.E. MASTER

DJC has released an archive based program for the management of employees P.A.Y.E. records.

It is specifically designed for the UK P.A.Y.E. rules and is aimed at employers who need to manage all the records associated with P.A.Y.E.

The program is available on disk or microdrive and expanded memory is needed to run the program.

P.A.Y.E. MASTER costs £30 (UK) + £1 for overseas postage and is available from Dilwyn Jones Computing (0248) 354023, 41 Bro Emrys, Tal-y-Bont, Bangor, Gwynedd, LL57 3YT.

PROGS - REDUCE PRICE OF PFLIST & PFDATA

PROGS have reduced the price of both PFList and PFData, to make them comparable with their other products, to BEF 1000.

If you have already bought the programs at the full cost you are entitled to a free update if you send PROGS your master disk. This includes the new ,faster, PROforma which includes an Epson ESC/P2 printer driver. You are also entitled to a price reduction of BEF 750 from the PROforma fontpack, this price reduction is only valid until the end of October.

For more information contact PROGS (016) 488952, Haachtstraat 92, 3020 Veltem, Belgium.

SMSQ - QDOS COMPATIBLE

SMS2 has been available for some time now but has never been advertised to the QL user for various reasons. Now the original plans have changed and it has been decided to concentrate on one operating system for all systems called SMSQ (SMS - QL compatible). The operating system will run on all systems with a faster processor and a minimum amount of 1MB of RAM, this includes Atari ST/STE/TT, QXL, GoldCard and SuperGoldCard.

SMSQ includes a new QDOS compatible operating system which provides a much better performance. Included in SMSQ is SBASIC, a new compatible SuperBasic language which is able to run as many interpreted SBASICs as you want. SBASIC can be started with EX and runs at about the same speed as compiled SuperBasic (QLiberated). You can also position SBASIC anywhere on screen even on larger than normal QL screen sizes. Another facility included is the ability to access a DOS or TOS disk exactly like a QDOS disk, the system detects the format of the disk automatically.

The above system is available now but it is planned to extend the system to include a new Screen Driver to allow the updating of overlapping windows (at the moment the job is suspended), to allow "background" floppy/harddisk operation (no waiting for the format to finish etc.) and a pointer-integrated SBASIC environment with Single-step/trace facilities and more.

For more information contact Jochen Merz Software (Germany 0203 501274), Im stillen Winkel 12, 47169 Duisburg, Germany.

	Feature	ATARI ST(E)/TT	(Super)GoldCard	QXL
This exists NOW! Shipping from mid/end of July.	New Operating System	NEW DM 199,-	NEW	already ex.
	Multiple, fast BASICs	NEW (if you own the QJUMP-drivers)	NEW	already ex.
	Flexible Level 3 Drivers	NEW Level C, D or E	NEW DM 199,-	already ex.
	HD Disk-drive support (STE/TT)	NEW already else	already ex.	already ex.
	TT Fast RAM support	NEW DM 249,-	impossible	impossible
And this will come later..	Monochrome Screen-driver	NEW + DM 50,-	impossible	impossible
	New Screen-driver	NEW + DM 50,-	NEW + DM 50,-	NEW DM 249,-
	"background" Disk/Harddisk	NEW + DM 50,-	impossible	NEW + DM 50,-
	BASIC-Development-Environment	NEW + DM 50,-	NEW + DM 50,-	NEW + DM 50,-
	Total price (when all is available)	with rebats, without monochrome DM 349,-	DM 299,-	DM 349,-

QUBBESoft P/D - MINI TOWER SYSTEMS

QUBBESoft P/D is offering to build custom cases for those people who would like their QL to be in an all in one case similar to the PC. Thus avoiding the need for multiple power sockets and avoiding the inevitable forgotten monitor lead syndrome. The cost of this service is entirely dependent on what the user would like and whether the component parts have to be bought or the user already has the parts. For an estimate of the cost QUBBESoft P/D has produced an enquiry form detailing the users requirements.

If you would like an enquiry form contact QUBBESoft P/D (0376) 347852, 38 Brunwin Road, Rayne, Braintree, Essex, CM7 5BU.



NEWS



PROGS - RELEASE FONTPACK

PROGS have released a product which includes one hundred high quality fonts for use with their graphic engine PROforma (Suitable for Linedesign, PFdata etc.).

The fonts have been licensed from Bitstream^R and include professional quality fonts with high quality outlines, a complete set of characters, full hinting (they are still readable at small sizes) and include a complete kerning table.

The fontpack includes a lot of body fonts in several styles (roman, bold, italic, bold italic), display fonts and some dingbat fonts including maths.

The fontpack includes a manual with a printed copy of all the characters. The fontpack costs BEF 4000 and is available from PROGS (016) 488952, Haachtstraat 92, 3020 Veltem, Belgium.

Amelia	Iowan	Old Style	BREMEN BOLD	Kuenstler 480	∞ΔΓαβ	Goudy	Handtooled
Carlton	Open					ian	Bodoni
Revival	565						
Shelley	Colley						
Venetian	30						
Calligraphi							
Carmina	bold	italic	Oblique	ITAL	ITALIC	LYUITH	ITALICA
Old	Dreadfal	no.7	Goudy	Old Style	Freeform	721	Dom Casual
							Amerigo
							Carmina Italic

PROforma fontpack **SHOTGUN**
Staccato SSS agull Bold iel Gothic

All fonts have high quality outlines, a full charset, hinting and kerning!

JOCHEN MERZ SOFTWARE - RELEASE SERIAL/PARALLEL CONVERTER

Jochen Merz Software has released a device that converts standard 25pin serial output to Centronics parallel output. The device is configurable from 300 to 57600 Baud including 9600 and 19200 Baud. The device also includes a 4K buffer and power-supply connector.

The device costs DM 159,-(P&P DM 14,- for Europe). For more information contact Jochen Merz Software (Germany 0203 501274), Im stillen Winkel 12, 47169 Duisburg, Germany.

PROGS - UPDATE LINEDESIGN & PROFORMA

PROGS have produced new versions of Linedesign (V2.06) and Proforma (V1.04).

The new versions include the reintroduction of the unit, bug in convert to cures is now fixed, fit to screen is improved, Adobe Illustrator files are now loaded with the proper colours, and there is improved support for the PROforma fontpack. The new version of Proforma is also faster and includes a new Epson ESC/P2 printer driver.

An update costs BEF 500 and can be obtained by sending the program disk only to PROGS (016) 488952, Haachtstraat 92, 3020 Veltem, Belgium.

DJC - RELEASE KOINEE GREEK NEW TESTAMENT BIBLE FILES

The Koinee Greek New Testament Bible Files consists of a copy of Quill patched to allow the greek characters to be used and the twenty seven New Testament Nestle Greek Scripture books in Quill format.

The Koinee Greek version of the Bible is considered as the definitive version, which has not suffered in the same way from translation (or slight mis-translation perhaps) to English, etc.

The Quill version comes on 4 disks and costs £10 (UK). A text87 or plain text version is also available on 2 disks costing £5 each (UK). Overseas postage is an extra £1 per program. For more information contact Dilwyn Jones Computing (0248) 354023, 41 Bro Emrys, Tal-y-Bont, Bangor, Gwynedd, LL57 3YT.

HELP	CURSOR	TEXT Insert:	Type at #	TYPEFACE	COMMANDS
press F1	move +	New para:	Press ENTER	press F4	press F7
PROPTS	with	Delete:	CTRL & F14		ESCAPE
press F2	keys +	Change mode:	SHIFT & F4		press ESC

The Letter of Jude.

Κουδω 1 'Ιουδωσ 'Ιησου Χριστου δοκωσ, αδελφωσ δε 'Ιακωβου,
Κουδωσ 1 τοισ εν δευ πατρις συστρεφωμενοι και 'Ιησου Αριστυ
Κουδωσ 1 τεταραμενοισ ελθουσι·

Κουδωσ 2 ελθωσ ων και εισηγη και άνομη πληθυνθεισ.

Κουδωσ 3 'Αγαπητοι, πασαν σπουδην ποιουμενοσ θραδεις ων περι
Κουδωσ 3 τωσ καιρωσ 'ημωσ σωτηριασ διαφυγωσ εστων υμωσ λογη
Κουδωσ 3 παρακαλωσ εμπερικραθεισ τη σωτη 'ακαθαρτων τοισ
Κουδωσ 2 θις τοισ υμωσ·

Κουδωσ 4 παρετεθειμεν ωσ ειπεσ ανθρωποι, οι οφθαλ προεστραμμενοι
Κουδωσ 4 εις τον ουρανον, οφθαλσ, την τον θεου οραν
Κουδωσ 4 αμωσ τε μετρησθησεν εις φθωσιν και παρ υμωσ
Κουδωσ 4 δεσποτων και κυριωσ 'ημωσ 'Ιησου Χριστου αποκρυφωσ.

MODE: INSERT WORDS: 617 PAGE: 1
TYPEFACE: Normal DOCUMENT: "Jude"

DJC - IMAGE PROCESSOR UPDATED

Image Processor, the graphics imaging tool from DJC, has been upgrade to allow the saving of screen images in the Pointer Environment Bitmap format. The area of the screen to be saved can be limited to a small area of the screen up to a maximum area of 512*256 (A normal QL screen size).

An upgrade costs £3 (UK) or £4 (Overseas) and can be obtained by sending the master disk back to Dilwyn Jones Computing (0248) 354023, 41 Bro Emrys, Tal-y-Bont, Bangor, Gwynedd, LL57 3YT.



NEWS



QUBBESoft P/D - RELEASE QUBIDE

QUBIDE is an IDE hard disk interface for the QL. It allows you to connect 16 bit IDE hard drives (common on the PC) to the QL.

The interface plugs into the ROM slot of the QL and includes the software. The hard disk drive can be of any size but only 128mb QL drives can be created. This is not a problem for drives over 128mb as you can partition the drive to behave as though it was two QL Drives of 128mb each ie. WIN1_ and WIN2_.

The cost of the Interface is £65 + P&P and can be obtained from QUBBESoft P/D (0376) 347852, 38 Brunwin Road, Rayne, Braintree, Essex, CM7 5BU.

THE QREVIEW SHOW

The QReview show will be a combination of Workshop and Traders fair to be held on Saturday January 28th 1995 at Rush Green Junior School, Romford. The show will be from 10am to 5pm.

Romford lies to the east of London and can easily be reached via the M25, A13 and A12. Romford can also be reached using public transport, it has a British Rail main line station and nearby London Underground Stations are only a short bus journey from the venue. More details will appear in the next issue.

The Show will have most UK QL Traders in attendance including Miracle Systems, Dilwyn Jones Computing, TF Services, QUBBESoft P/D and of course Quo Vadis Design. There may even be an appearance by several Traders from the continent. Talks and demonstrations are also being planned.

The finalised details of the show will appear in the next issue of QReview due out at the end of the year. If you would like further details on the show you can contact Bruce Nicholls at the Quo Vadis Design address/telephone number given at the front of the magazine.

COMPETITION

Win up to 7 Dilwyn Jones Computing programs in this free to enter competition

Dilwyn Jones Computing is offering 13 programs, in their ever expanding range of software, as prizes for this easy to enter FREE competition (no purchase necessary). The winner of the competition will receive 7 programs of their choice from the programs listed below, which could be worth £80. The second place winner will receive 4 programs of their choice and the third place winner will receive the remaining 2 programs.

To enter the competition all you have to do is submit a brief written outline of a new QL program you would like to see become available. The submission must be between 100 and 200 words long.

Please send your submissions to QReview (DJC Competition), Quo Vadis Design, 57 Shaftesbury Road, Romford, Essex, RM1 2QJ. The closing date for submissions is December 10th 1994, with the winners being announced in QReview Volume 2 Issue 2.

Basic Reporter	£10	Slowgold	£5	Convert-PCX	£10
Rob Roy Pack	£10	Image Processor	£15	Vision Mixer	£10
Quick Posters	£10	DJToolkit	£10	Sidewriter	£15
The Cat	£5	PD2 Clipart	£10	Screen Snatcher	£10
Screen Compression	£10				

B E Y O N D Q U I L L

In a recent magazine survey 90% of readers expressed an interest in word processing and 50% described it as being their favourite use of the QL. About half used Quill as their main or only wordprocessor and half another wordprocessor, which was usually either Perfection or Text87. Put in another way, although practically every QL user has obtained a good user friendly wordprocessor free of charge with his computer, about a third are prepared to spend more than £80 to replace this wordprocessor with another one.

Is it worthwhile to do this? I expect that most of us are using our wordprocessor for such things as letters and short reports, and are not writing an 80,000 word novel. Quill is reputed to be slow and difficult to use when a document exceeds about 3,000 words, but how many of us regularly write a document of this length? If we buy a new wordprocessor are we wasting our money on extra facilities that we shall never or rarely use? Is it, to paraphrase one writer, the equivalent of buying a furniture van to bring your shopping home from the supermarket?

This series of short articles aims to examine this question, to help you decide whether or not it is worthwhile to move on from Quill to another wordprocessor. If you have already moved on I hope that you will find some tips that will help you to get the best from your investment. I shall be making the comparison of Quill with Perfection (Special Edition) for the simple reason that Perfection is the wordprocessor which I use. I have used it from version 2 to the present version 5. I have no personal experience of Text87, but from the comments I have read I have no doubt that it is a very good program. Many of the arguments for a move from Quill to Perfection would also

apply to a move from Quill to Text87.

One thing that I would like to make clear from the start is that I am not one of the "speed-freaks". I know that Perfection works much more quickly and efficiently than Quill, but frankly Quill works faster than I can type so that it is only when I carry out searching and replacing operations that the speed is important. There are many other and better reasons than speed for moving on from Quill.

One of Quill's weaknesses is cursor navigation. It is possible to move the cursor by one character, one line, one word, or one paragraph per key press. There are also commands for going to the top or bottom of the document or for going to any page. Scrolling tends to be slow and, although I have never been troubled by it, some users report frequent crashes when scrolling backwards. Another wordprocessor will almost certainly offer more cursor movements. In addition to those available in Quill, Perfection allows you to move the cursor to the beginning or end of a line, by page or by screen depth or width. You can also go to the top or bottom of the document, to any page or to any marked place in the text. These extra commands allow you to travel through the document more efficiently and quickly and reduce the need for continual scrolling. If you do use scrolling in Perfection it works faster and with less problems than Quill.

There is, however, a disadvantage to these extra possibilities and that is that there are more commands to learn. For me this is no problem because I use an IBM-style keyboard and have the extra keys "Home", "End", "PgUp" and "PgDn". However, if you still use the original QL keyboard you have to remember a lot of combinations of

the Alt, Ctr, Shift and the cursor keys.

I think this is what puts many new users of Perfection off. You seem to be inundated with new commands, whereas Quill was so simple. There are a number of things that you can do to help in this situation. The simplest is not to use the new commands. I am not being facetious in suggesting this. Perfection works without the new commands and it is not compulsory to use them. There is a lot to learn in a new wordprocessor, and if you try to learn everything at once, you just become confused. In the beginning just use the cursor movements that you use in Quill. When you have got used to using Perfection, set aside an hour, load a long document (such as "help_data" on the Perfection disc), make a note of all possible cursor movements and practice using them.

Keep a copy of the list you have made of the cursor navigation commands and pin it up by your work area. Alternatively print out the help_data file and mark important points on it with a felt tip pen. Finally make use of the help that is available in Perfection by pressing F1. Help in Perfection is different from help in Quill. In Quill the help screens are a sort of mini-tutorial that you use when learning the program and then do not use again. The Perfection help screens are curter and crammed full of information. They are a reference guide to which you may constantly have to refer. You can use all the search commands that are available in Perfection when searching for something in the help file.

The Perfection help file is a normal Perfection document, which, if you wish, you can modify to suit your own needs. If you want to be clever you can even write your own help file. A Perfection user whose native language is not English, could write

BEYOND QUILL

a help file in his own language. If you do write your own help file, save it by using the export command (F3 I), and don't forget, maybe other QL users can use it too.

If you buy a new wordprocessor for your QL, there is a reasonable chance that, initially at least, you are going to be disappointed. This is not a criticism of the two alternative wordprocessors for the QL, Perfection and Text87. Both of these programs have many happy and satisfied users. It is rather a comment on the reaction of the average computer user when confronted with a different wordprocessor from the one he usually uses. A wordprocessor is not a toy or a computer game, but a serious tool that we use in our business and private lives, and ideally, just like a new car, we want to make serious use of it as soon as we have bought it. When you buy a new car you want to drive it straight away, and not have to stop in the middle of the road to consult the manual to see what you have to do next. However with your new wordprocessor you will often have to do the equivalent of that.

Let me give an example of what I mean. Footers are very easy to design in Quill. You press F3 followed by F, and at the bottom of the screen appear a series of instructions. You first have to choose where you want the footer on the page - None, Right, Left or Centre; you then are invited to enter the footer text and codes are given for the various numbering options - Arabic, Roman and Alphabetic; you then enter the number of blank lines to be placed between the bottom of the text and the footer; and finally you are given a choice between bold or normal text for the footer. It's as simple as that.

A footer in Perfection starts in the same sort of way, although you have

to press F3 three times followed by F (or Alt/Ctr/F) to access the command. Do this, however, and all you get is a blank screen. Suddenly you have a sinking feeling and wonder where to begin.

At first sight it appears that you have taken a step backwards in moving from Quill to Perfection, but it is Perfection that has the better footers. The reason that footers in Quill are so simple is that they are very limited in their possibilities. Some users want sophisticated footers. A writer of magazine articles, for example, may like to have a footer containing the page number followed by a blank line followed by his name and address in a different typeface from the main text. You cannot do this in Quill. You can do it in Perfection. In fact you can do practically anything in a Perfection footer. Look at how Digital Precision show off in the footers in the Perfection manual!

There is a price to pay for these extra possibilities and that is that Perfection makes more demands on the creativity of the user. Hence the blank screen when you edit a Perfection footer. You have to make your own choices of typeface, content, style and place of the numbering etc, and then look up and enter the appropriate control codes. It means making extensive use of the manual and the help screens. For example, you want to know the code for arabic numerals? Enter the help screen by pressing F1. Use the search facilities (Ctr/F) and look for "Arabic". Not found? Try "number".

If you move on from Quill to another wordprocessor you will be constantly confronted with extra decision making and extra demands on your creativity. How do you avoid the wordprocessing equivalent of sitting in a new car, desperately looking something up in the manual, with a whole line of motorists

behind you impatiently sounding their horns? In your first few sessions with the new wordprocessor forget all the wonderful things it can do; forget all the hype in the advertisements; and forget all the sales talk. Treat it as though it was still Quill and do only simple things.

In Perfection three commands, Load, Save and Print, are invoked in the same way as Quill by pressing F3 followed by the initial letter of the command. In the beginning just use these three commands. Type text without worrying about such things as margins, justification or underlining. Save it. Reload it. Try printing it out. There is a good chance that this will work without modifying the printer driver.

Next there are three commands that are entered in the same way as Quill, except that you have to press F3 two times. These are Justify, Margins and Tabs. With these commands there are some differences with Quill. Look these up in the manual or the help screen. Above all be prepared to experiment! Now try entering bold or underlined text. This is done differently from Quill. Flip through the commands at the top of the Perfection screen by pressing F3 continually to discover how to do it. Master these commands and you are already using the wordprocessor.

Flip through the commands again and try those that you have not yet tried. You want to learn how to write a footer? Enter the footer command and then try entering a simple footer similar to those in Quill. If you are successful, then be a little more ambitious and try something you cannot do in Quill. An hour set aside for experimentation saves a lot of time and frustration later on.

Geoff Wicks

QUANTA



Independent QL Users Group

Worldwide Membership is by subscription only, and offers the following benefits:

Monthly Newsletter - up to 40 pages

Massive Software Library - All Free !

Free Helpline and Workshops

Regional Sub-Groups. One near you?

Advice on Software and Hardware problems

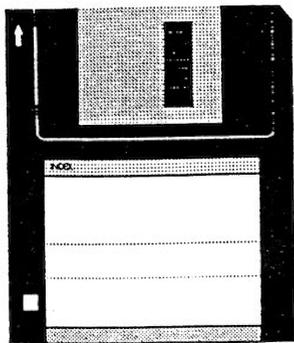
Subscription just £14 for UK members

Overseas subscription £17

Barclaycard: Visa: Access: Mastercard

* Now in our ELEVENTH successful year *

Further details from the Membership Secretary

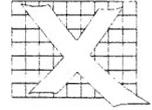


Bill Newell
213 Manor Road
Benfleet
Essex
SS7 4JD
Tel. (0268) 754407





FDU



FDU is a Qliberator compiled SuperBasic program that contains the QLIB runtime routines, DJToolkit 1.13 and some extensions with names identical to some of the Turbo Toolkit. The program was written by Davide Santachiara and is up to his usual standards. I was unable to make the program 'fall over' while testing it on a normal QL fitted with a Gold Card but it will not work for long on my QXL.

On the QXL, FDU comes up with its initial screen than a Qliberator error message informs the world that 'at line 30136 - STRINGF invalid parameter' has taken place. The options to continue, abort or retry can be used. Abort does just that, retry just causes the same error and continue tries to go on but crashes the QXL back to 'Messy DOS' with the error 'the QXL at IO address 2B0h is not responding'. This is the only program I have used that causes this, but I am sure that Ergon will have it fixed before long.

Just in case there was a clash between the Turbo Toolkit and the extensions used in the program which have the same names, and probably functions as well, I tried again without Turbo Toolkit loaded but the same thing happened.

There is only one version of the program on the supplied disc, and it contains the Qlib runtime library. Anyone who normally loads the runtimes in order to save on memory when running a number of Qliberated programs will have to waste around 10 k of memory when running FDU. Perhaps future releases will have a version without the runtimes built in as well.

FDU supplies a number of floppy disc utilities, hence the name, in a single program. These utilities allow floppy discs to be edited, copied, formatted and checked.

On execution, the program signs on with the common Ergon main screen and, at the press of any key barring ESC, the main menu is displayed. As with other Ergon programs, FDU uses the 'Basic Development System' of menus. This gives a familiar feel to the program if you have used other Ergon products such as DEA, but might be a bit confusing at first for beginners.

The Main Menu

The main menu has nine options. Some of these relate to setting up the devices to be used and the remainder let you select which operation you wish to carry out on the floppy disc in question. The main options are :

Disc editing
Copying and verifying discs
Checking a disc
Formatting discs
Setting a discs parameters

The last option in the above list is the most important one and should always be chosen when a new disc is to be operated on. This lets the program know all about the floppy to be processed. As a minimum, all that need be done is to select DOUBLE, HIGH or EXTRA HIGH density and let the program find out the rest on its own. There is no currently known way for the density of a floppy to be automatically discovered (unless you know different ?) so make sure that you get it right before continuing. The program at this stage cannot handle MSDOS discs automatically.

Formatting Discs

This part of FDU allows you to format a floppy disc, rename it, format it with extra capacity or even, format it with less capacity than normal.

I am not at all sure of the safety of formatting a floppy with extra

capacity. A readme file on the FDU disc gives details of drives that cannot handle the 5% extra tracks that can be attained using FDU so a second option allows 2.5% extra tracks to be formatted. My drives were able to carry out the format and the extra space was usable, but, I still have my doubts.

Some drives could be damaged by trying to force the heads to a track that is not supposed to be there. Software distributors would be wise not to attempt to squeeze extra files into the extra tracks, just in case.

On the other hand, all drives are able to read from a disc that has been 'short' formatted. This option is very handy for software distributors as it reduces the amount of time that they have to wait to format a floppy. If you check out some of your master discs from various suppliers, you will probably find that many of them are not giving the full 1440 tracks (DS/DD) for example. Why format 1440 tracks when all you need is 200 ?

None of the above formatting options will work unless your disc interface supports the FLP_TRACK command.

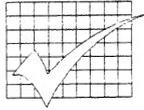
A disc being formatted takes its density and number of tracks information from that supplied by the user on the 'Setting a disc parameter' option from the main menu.

Renaming a disc is a simple matter of typing in a new name and letting FDU get on with it.

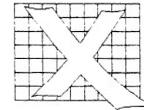
All formatting options can be abandoned by pressing ENTER on its own when asked for a media name.

Checking Discs

Checking a disc is a simple matter of letting FDU read every sector on



FDU



the disc. As it checks each track, a graphic indicator lets you know how well it is getting on. Tracks which have one or more bad sectors are indicated in red, while good tracks are shown in green.

FDU cannot tell whether the information stored in a disc track is good, bad or indifferent so don't expect miracles. A file can be corrupt simply by having 2 bytes interchanged, but the sectors the file lives on are readable so FDU considers the disc to be ok.

Copying and verifying

Once you have chosen the source and target drives on the main menu, you are all set to make single or multiple copies of the source disc. FDU copies whole discs at a time and does not copy single files. If you need to copy files, use WCOPY or something similar. FDU is for copying whole discs.

If a disc is found to have bad sectors, these are written to the output disc depending upon what information was read off of the source disc. This is good and it means that you can use the copy to play with the disc editor rather than trashing the only copy you have of the information.

Verifying can be turned off or on according to your needs, as can pre-formatting the target disc before the copy takes place. Both of these options increase the time to make a copy.

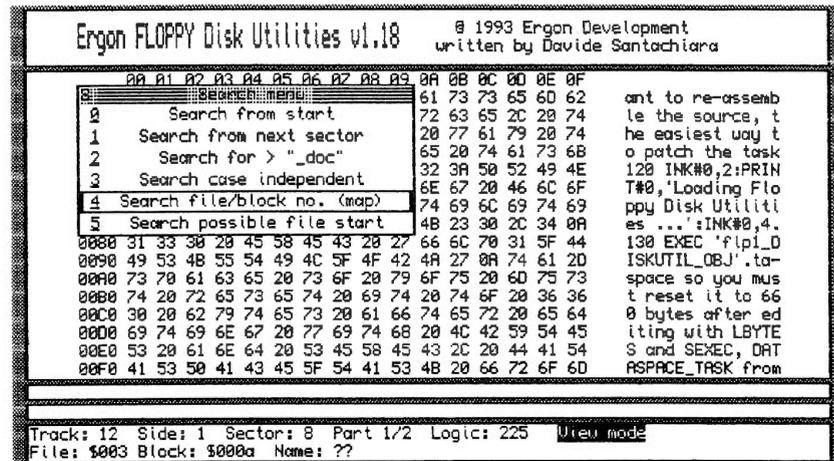
Internally, FDU optimises the order that sectors are read from the disc, depending upon whether it is running on a Gold Card or not. This helps to speed up the copying. Before the copy gets under way, FDU tells you how much of the disc can be held in memory and how many passes are required to make the required number of copies. Watch the prompts and make sure

you put the correct disc in when asked for it!

As the copy is progressing, FDU shows each track as it is read and subsequently written. Once again, green shows a good read or write,

before the start of the copy, it tells you how many disc swaps will be required. This information is also displayed when making multiple copies.

Two discs can be compared



red a bad one and white shows a good verify, if this option is switched on.

The target disc(s) should all be the same density as the source disc. In this option, the program doesn't seem to mind getting a MSDOS disc in the target drive as it is writing to it. As long as the target is formatted (or will be formatted by the copy) it will work ok. Bad sectors are noted on the screen as they are detected.

Copying a double density (720K or 1440 sectors) floppy took 148 seconds with everything turned off. Verifying the disc after writing it took an extra 66 seconds when that option was turned on. FDU copies the entire disc or as much as it has in memory, then verifies it. The program detected that a sector was bad on the source disc which probably increased the time taken to make the copy.

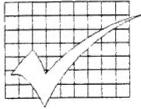
If you only have a single disc, FDU can held make backups as it will ask for the source and target discs to be swapped as and when required. Just

together, verified, without copying one to the other first - obviously, this will not be of much use unless the 2 discs are copies of each other. Of course, you could use this to check that one disc is a clone of the other and not just a WCOPY or something else along those lines.

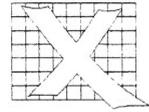
Disc Editing

This is the most complex and potentially the most dangerous part of FDU. Disc sector editing can be used to rescue lost data or to completely trash it beyond all hope of recovery - you have been warned!

Because of the nature of this facility, a fairly large section of the manual is devoted to a tutorial of its use. This tutorial is easy to follow, however, before trying out any of the editing functions, MAKE A BACKUP OF THE DISC YOU ARE WORKING ON AND USE THAT. It might also be a good idea to read the section in the manual that explains exactly how tracks, sectors, blocks and files are arranged on a floppy disc. A better



FDU



understanding of the background processes can help sometimes.

The disc editor allows you to directly edit sectors, search for some text on the disc, collect files from the corrupted disc and, if your directory has been lost, search the disc for all possible file starting sectors.

This last option is handy. If your directory sectors are bad, then how do you know where a file begins on the disc? The 'easy' answer is that when the file was first created, a copy of the 64 byte directory entry for that file was written to the first 64 bytes of the first sector used to store the file. This header was not an exact copy as it contains only the file name and its size, the size always being set to 64 bytes.

Why this disc layout is used, I have never seen written down, but as it is there, FDU makes good use of it to find potential files for recovery. Once written, this dummy header is never re-written or updated for that file.

Once a file start sector is found, its details are written to the screen ready for you to start collecting, or you can ignore it and try to find another file.

Collecting files is the relatively simple process of opening a file to hold the recovered data, the collector file, collecting all possible sectors of the original file from the bad disc and closing the collector file. A small amount of editing may then be required to remove extra rubbish from the final sector. The collection process writes complete sectors to the collector file, but if the original file did not fully fill up the last one, certain amounts of rubbish will be held in it and requires editing.

This is fairly easy to do for a file

that is pure text, such as a SuperBasic listing, but for other file types, it could be a problem. Quill files, for example, have their own 'rubbish' areas at the bottom of the file so it can be difficult to figure out where this good rubbish starts and where the nasty stuff starts. (One little utility that might help, but only with recovered Quill files, is Dilwyn Jones' STRIPPER program that converts a Quill file into plain text. This is available from the Quanta library on disc PS05. It doesn't care about the good or bad rubbish, provided that the first sector of the file is ok.)

Machine code files will be a problem, especially if you don't know about their length, dataspace, file type etc. FDU doesn't help much here, unless you load the directory into FDU - this does show the required information, but if the disc directory is trashed, much of it will be corrupted anyway. (I bet you wish that you had made a backup now don't you?)

Files can be collect automatically. Select the file from the list in the directory and the program will show a menu of all the block numbers for that file. You can deselect one or all of them so that it/they are not included in the collection then press ENTER. The 'where am I on the disc' indicator will then go absolutely mental as FDU zooms around the disc extracting sectors (in order) and writing them to the collector file.

Once this is done, and it might not take very long, the file is collected. Close the collector file and you are ready to collect the next file.

Collection can be done manually, by sector or by block (groups of 3 sectors) should you not be able to collect automatically.

It has to be said that manual disc

recovery is a long and very boring job. I have had to do it in the past, once for my own discs and a couple of times for a Quanta member (who shall remain nameless!) who had a couple of Archive discs trashed and no backups. You might think that backing up your work is quite boring, but just wait until you are in the position of having to attempt a recovery. Keep taking those backups.

Summary

Davide has done a good job of automating the disc recovery process and I find it hard to see how it could be made any easier. The formatting utilities work well, but I am a bit suspicious of the extra capacity functions. With floppy discs costing around 25p each these days, is it worth trying to get 5% extra storage and run the risk of damaging your drives? That said, they do work.

Disc copying is quite fast and the verifying option is quite nice as it gives extra piece of mind that the copy did actually work.

All in all, a good and useful program. You can obtain FDU direct from Ergon Developments in Italy, or if you wish, from Dilwyn Jones Computing in deepest, darkest Wales. Anything that uses DJToolkit can't be all bad, can it?

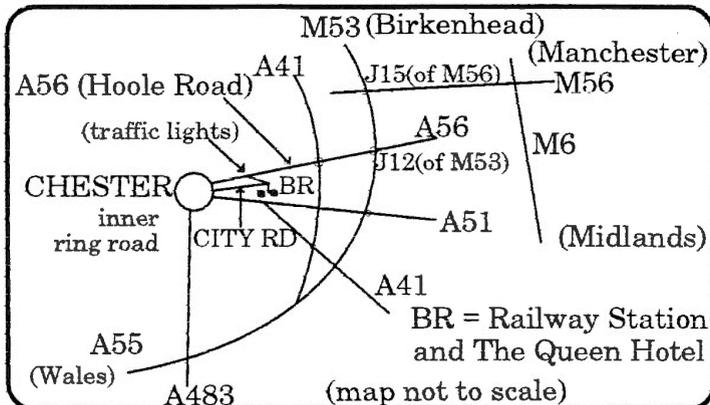
Norman Dunbar

Floppy Disk Utilities costs £18 and requires expanded memory. It can be obtained from either Ergon (Italy 39 522 70409), c/o Davide Santachiara, Via Emilio De Marchi 2, 42100 Reggio Emilia, Italy or Dilwyn Jones Computing (0248) 354023, 41 Bro Emrys, Tal-y-Bont, Bangor, Gwynedd, LL57 3YT.

CHESTER QL SHOW

SUNDAY 16TH OCTOBER 1994 10AM TO 4.30PM

The Queen Hotel, City Road, Chester (opposite railway station, at junction of City Road and Brooke Street, follow signs for station)



- Admission free!
- Car parking (with token from hotel)
- QL traders present
- Talks and lectures
- Easy to find venue
- near railway station
- Bring & Buy stand
- Meet other QL users
- Organised by DJC

ACCOMMODATION - Several hotels / guest houses near to the venue.

FURTHER DETAILS (e.g. hotels list): Dilwyn Jones (0248) 354023

Miracle Systems Ltd (0904) 423986, Mike Kenneally: (0625) 878207

(Thanks to Mike Kenneally (Club QL International) for general help!)

***DJC LOOKS FORWARD TO MEETING OUR CUSTOMERS AT
THE CHESTER QL SHOW ON 16/10/94!***

WATCH OUT FOR SPECIAL OFFERS & BARGAINS AT THIS SHOW!

We will also be at Clevedon Quanta Workshop 20/11/94
and at the QReview show, Romford, 28/1/95

THE POINTER ENVIRONMENT & QPAC2

POINTER ENVIRONMENT "IDIOT'S GUIDE" You may have heard a lot recently about the Pointer Environment and QPAC2, but may have been frightened off by reports that it is difficult to learn and to set up. True, there is a lot to learn, and this can make it seem difficult at first. But DJC has listened to QPAC2 users and done something about it! A new 20 page booklet by Norman Dunbar sets out to provide a simpler introduction to this most essential of QL software. Pointer Environment will become an essential part of every QL user's software setup sooner or later, especially when enhanced QL graphics hardware becomes available! Oh, the title - don't be offended, all is explained in this booklet! Available FREE on request with orders for QPAC2, or £2.00 to cover costs if you already have QPAC2.

QPAC2 - Shareware program to help you to set up boot programs for use with QPAC2. Expanded memory required. QPAC2 is available from QL PD libraries, or from DJC for £2.00

QPAC2 £39.95

QPAC1 ... £19.95

QPAC1 is an accessory pack of several small pointer driven programs which can be used with or without QPAC2

LINE DESIGN PROGS

It was used to produce and print this ad - say no more! £100.00
(printed from a QL using a HP Deskjet 500)

PROFORMA FONTPACK

ALSO AVAILABLE - new pack of 100 high quality fonts for use with LineDesign and Proforma £80.00

PROFORMA

'C' programmers can now access the drivers, graphics etc of Line Design! £100.00

DATA DESIGN database £60.00

DATA DESIGN API £20.00

PFDData £20.00 PFlist £20.00

ptb2pff (font converter) £60.00

DJC**DILWYN JONES COMPUTING**41 BRO EMRYS, TAL-Y-BONT, BANGOR, GWYNEDD, LL57 3YT,
GREAT BRITAIN. TEL+FAX: (+44) 01248-354023

Address Book & Label Printer	15.00
AMD Airplan	10.00
Banter **SAVE £5.00**	20.00
BASIC Reporter	10.00
Bible text disks	20.00
Cocktails Waiter	10.00
Convert-PCX	10.00
Cricket Secretary	12.00
Cue-Shell	40.00
Data Design 3	60.00
Data Design API	20.00
DBEasy	15.00
DBProgs	15.00
DEA disassembler	26.00
Deskjet-A5	12.00
Disa 2	40.00
Discover	20.00
DJToolkit	10.00
Easypr 3 part 1	40.50
Easypr 3 part 2	20.00
Easypr 3 part 3	20.00
Filemaster	12.00
Files 2	12.00
Five Game Pack	12.50
Flashback	25.00
Flashback Special Edition	40.00
Fleet Tactical Command	39.95
Flightdeck	15.00
Floppy Disk Utilities	18.00
4Matter & Locksmithe	23.50
Fountext94+Founted89 (Text87 driver)	39.00
FTC Data Printer	9.95
Fugitive	9.95
Genealogist 3	60.00
Genealogist 2nd Edition	30.00
Genealogist 2 to 3 upgrade	33.00
Genealogist 128k (budget version)	12.00
Grey Wolf	12.50
Home Budget	20.00
Image-D	10.00
Image Processor **SAVE £3.00**	12.00
Line Design 2	100.00
Locksmithe	14.95
Masterbasic	22.00
MDV Toolchest	14.95
MegaToolkit (disk)	25.00
MegaToolkit (eprom)	40.00
Micro Process Controller	64.50
MPC software	9.95
(Toolkit for use with Process Controller)	
Multi Discover	30.00

Music Manager	12.00
Network Prover	4.00
OPD Interchange	15.00
Open Golf	12.50
Open World	18.00
Page Designer 3	40.00
Painter	25.00
P.A.Y.E. Master (UK only)	30.00
PD2 Clipart	10.00
Picturemaster	15.00
Picturemaster Plus	20.00
Printermaster	20.00
Publishers Pack	199.00
QIndex	20.00
QL-PC Fileserver	24.50
QLiberator 3.36	50.00
QLiberator, budget version	25.00
QLibrary Manager	18.00
QLOAD and QREF	15.00
QPAC1	19.95
QPAC2	39.95
QPower regulator	24.95
QRactal	20.00
QTOP	29.95
QTOP2	29.95
Question Master	10.00
Quick Mandelbrot	15.00
Quick Posters	10.00
Quiz Master	10.00
Ramdisk	2.00
Return To Eden	17.50
Rob Roy Pack	10.00
R.P.M.	15.00
Scanned Clipart 1	10.00
Scanned Clipart 2	10.00
Screen Compression	10.00
Screen Dazzler	15.00
Screen Economiser	10.00
Screen Snatcher	10.00
S_Edit editor	20.00
SerMouse (+2.50 postage)	40.00
Sidewinder Plus	24.95
(screen dumps)	
Sidewriter **SAVE £3.00**	12.00
(sideways text printing)	
Slowgold	5.00
Solitaire **SAVE £5.00**	10.00
Save 1/3 on normal price!	
Spectrum Emulator Zm/128	28.00
Spectrum Emulator Zm/HT	40.00
Speedscreen (disk/mdv)	15.00
Speedscreen	
(Eprom+disk/mdv copy)	30.00

Spellbound	30.00
Spellbound Special Edition	50.00
Squidgy Round The World	12.50
SToQL	12.50
Super Disk Index	12.00
Super Disk Labeller	10.00
Taskmaster	25.00
Taskmaster hard disk upgrade	5.00
Text87 Plus 4	79.00
2488 Text87 drivers	29.00
Textidy	15.00
Text 'N' Graphix	15.00
The Clipart	12.00
The Gopher	12.00
3D Terrain	12.50
Toolkit 2 (disk/mdv)	19.95
Toolkit 2 (EPROM)	24.95
Trans24	10.00
Typeset94 (Text87 drivers)	29.00
Video Titles	5.00
Vision Mixer 1	10.00
Vision Mixer Plus	22.50
Winback	25.00

SPECIAL OFFERS VALID TO 31/12/94

FLOPPY DISKS & LABELS

FLOPPY DISKS DSDD 3.5"	0.40
FLOPPY DISKS DSHD 3.5"	0.70
ED DISKS - Call to ask for prices!	
DISK LABELS 3.5" roll 100	2.00
DISK LABELS 100 on tractor feed	2.50
ADDRESS LABELS roll 100	2.00
MICRODRIVE CARTRIDGES	2.50
Add 2.50 postage for floppy disks, or 0.50 postage if only ordering labels.	

MAGAZINES

QReview magazine UK per issue	2.00
QReview (Europe)	2.50
QReview (other countries)	3.75
IQLR from Vol 4 Issue 2 (each)	4.25

CATALOGUECall or write for a FREE copy of our
full 28 page QL catalogue.**DISCOUNTS (SOFTWARE ONLY) 5% FOR 2 PROGRAMS, 10% FOR 3 OR MORE, INCLUDING OFFERS!**

TERMS AND CONDITIONS. All prices shown in UK Pounds Sterling. Software sent post free to U.K., overseas add 1.00 per program (maximum 3.00). Bulky items - see above for postage. PAYMENT - Make cheques or Eurocheques (in Pounds Sterling, drawn on UK branch of bank or building society) payable to "Dilwyn Jones Computing". Payment by credit card (Visa/Access/Mastercard/Eurocard) also accepted. Minimum order is now 5.00 due to bank charges. Goods, whether individually identifiable or not, remain the property of Dilwyn Jones Computing until payment in full for them has been received by DJC. Under credit card company rules, we can only send goods paid for by credit card to the cardholder's registered address.

VISA
ACCESS
M/CARD
EUROCARD
all accepted!



SVENSKA QL-GRUPPEN



I'm writing this to tell you a little about the Swedish QL Group and QL in Sweden. When the QL was launched, 10 years ago, it was met with interest in Sweden. They say that we are particularly fond of establishing associations here in Sweden, I don't know if that's true, but soon there was founded a club around the QL. It is (and was) called "Svenska QL-gruppen" (The Swedish QL Group) and was founded in 1985. The first member list (from October 1985) shows a count of 33 members, most of them living in the neighbourhood of Gothenburg.

The main reason why the group was founded in Gothenburg was probably that the QL was used for education at the Chalmers University of Technology, in Gothenburg.

One year later we were about 160 members.

I think that I joined the group in the spring 1987, I would have joined earlier if I had known about the group. Since then I've been an active member of the Swedish QL Group. Now I am (unfortunately?) also secretary of the group.

When the QL was new, the only way to acquire a QL in Sweden was to go to England and buy a QL there. There was, however, soon lots of small enterprises importing the QL (English version) and you could buy your QL from about 10 different sources. The General Agent did not sell any QL's for a long time, they wanted to make a Swedish version, with our special letters Å, Ä & Ö on the keyboard, first. When they finally were ready valuable time had been lost (some people even bought "IBM's") and they found that the QL was not such a success in Sweden as for example the Spectrum had been. Besides I think they priced the QL too high, a QL bought from them was about twice the price compared to a QL bought in England.

The QL did soon find a place in education, as I said before, at Chalmers University of Technology, and the Swedish QL Group still has strong connections to Chalmers. The Ericson telecom company bought a large amount of QL's

(according to some rumours 1000 QL's) for their employees, and a club called "EDKX" was started. We tried to establish a cooperation with EDKX but I'm sad to say that we did not succeed. I don't know if EDKX are still active today. (By the way, I bought my first QL second hand from an Ericson man who was moving to the USA.)

At my office I convinced the manager that the QL was the cheapest way of learning to use computers, and we were able to put a QL on everyones table. Some of my former colleagues did not appreciate it, but most did. The main use of the QL was as a wordprocessor using Quill. I soon found it necessary to write a compendium on Quill, in Swedish, because many of my former colleagues had difficulties in understanding English, and the "User Guide" was not very easy to follow. I have been told that it is a very good compendium, but I don't think it will ever appear in an English version. The introduction of the QL in my former office was rather successful, but according to orders from a higher authority they have now been abandoned in favour of MS-DOS-computers.

I then spent most of my spare time for a year on writing a second compendium, this time on Abacus, but it still misses that "finishing touch".

For the last three years I have spent most of my "QL time" on preparing a big database on photographic chemicals, as photography is another big interest of mine. (I'm a chemist by profession.) I have also spent a lot of time on the construction of a new box for the QL, and maybe I will write a separate article on that some time in the future. It's my intention to make that into a "club project" in the Swedish QL Group.

I know of only two places where the QL is put to "professional work" nowadays. One is by a member of the Swedish QL Group, Inge Palmqvist, who has written a very big program in SuperBASIC for his profession (he is a bridge builder), and the other is by an authority in Norway, where Arvid Borretzen has a network



SVENSKA QL-GRUPPEN



(of QL-emulating Atari's). All other QL users I know are using QL as a hobby, or in education, and I'm sad to say that these days far too many are abandoning the QL in favour of IBM or Macintosh.

Since Sweden is a rather big country (by area) many of our members cannot visit our monthly meetings held in Gothenburg or Linköping (where we have an active sub-group), and thus we think that our magazine "Svenska QL-bladet" is our most important way of supporting our members. We do also have a program library for our members and do exchange programs with other QL clubs. We are also planning some easier hardware "projects", for those who like soldering. Once a year we arrange a bigger meeting, usually in Linköping and we also try to go abroad once a year to visit some meeting in Europe.

This autumn we will try to arrange a "Scandinavian meeting" which will be held in Gothenburg the 1st and 2nd of October. As I'm writing this I don't know how we will succeed, but I've already heard from our Norwegian friends (NASA) that they will try to come. We don't know about any QL clubs in Denmark or Finland, but we could always hope that the rumour of the meeting will reach Denmark and Finland too. *[The meeting is going ahead]*

We do, of course, expect most visitors to come from Scandinavia, but anyone with an interest in the QL and related things is very welcome. Most people in Sweden understand English. (Besides I think Gothenburg is very nice, but I'm born here so maybe I'm biased.)

The Swedish QL Group intend to show :

Two different I/O-cards for experimentation, one simple and one advanced.

Two different enclosures for the QL, one built of standard components, and one built from sheet metal.

How the QL is used in the education at Chalmers University of technology in Gothenburg.

Some examination tasks from Chalmers University of Technology.

A big technical program, written in SuperBASIC by one of our members.

A rather big database on photographic chemicals (in Archive).

Chalmers Mathematical and Graphical Toolkit (maybe already well known).

A home-made digitizer and a plotter made by one of our most prolific mechanical "hackers".

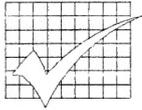
And maybe something more...

but as we are expecting some foreign guests we intend to give them the highest priority, and thus some of our own things may be cancelled.

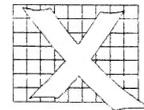
We have lost quite a number of our members during the last two years, and we don't know exactly why. We have also acquired a few new members. Our members are inherently lazy, and don't tell us what they want from the Group, and thus we can only guess what they want us to do. Many of our lost members have probably handed over their QL to a son or a daughter and we believe there is a need of support to the very beginner. In the future we will try to provide that and we are now starting a special column for beginners in our magazine "Svenska QL-bladet", and thus I think, it is very important that anyone giving away their QL to a relative also informs the the Swedish QL Group.

This applies, of course, to other countries also, but then the important information is for example in QUANTA, QReview. Since the QL nowadays is becoming a rare computer we have to stick together to be able to support each other.

Sone Thorén



QXL



The QXL from Miracle Systems has taken a lot of time and resources to be developed, rumoured to be over £50,000, and now is only beginning to reach its potential.

The problem with its early development lie not with the hardware but with the software. The first releases of the software SMSQ, more of which later, did not provide the full resources which a normal QL compatible could offer. The main stumbling blocks were the lack of a SuperBasic and compatibility with the older QL software. Now the software has been brought up to date with a new and powerful SuperBasic compatible language known as SBASIC and compatibility with older QL software has been made acceptable. Has the QXL come of age ?

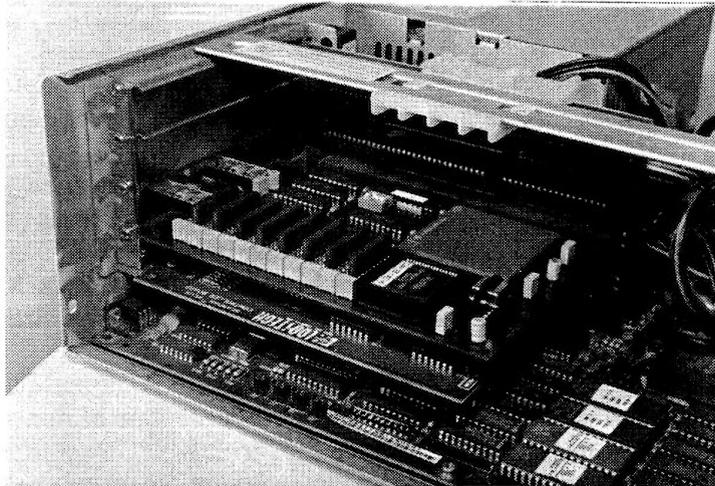
WHAT IS THE QXL?

The QXL is a small PC (Personal Computer) circuit board, called a half card (because of its small size) which plugs into a slot specifically designed to accept these type of cards on a PC. The slot is called an ISA slot which stands for Industry Standard Architecture. The card contains a 68040 processor (68EC040 to be precise) which runs at a speed of 20MHz (Mega-Hertz). This is a very powerful processor which provides the QXL with a very good speed rating of about 4 times the speed of a Gold Card. The RAM of the QXL is also contained on the board which can either be 4M or 8M as detailed in Miracle Systems latest advert. It is important that the RAM is on board the QXL as you can switch between the native PC mode and the QL mode without interrupting the QXL. This means you can be running both PC and QL applications at once. The QXL only uses the PC for its I/O system giving QL programs access to the PC's floppy disk, hard disk, keyboard, display, serial and parallel

ports. For further compatibility the QXL has QL style network ports.

To be able to run the QXL on a PC

into this slot (See the photograph on this page). The QXL comes pre-configured for a certain address which should work in most machines



the PC must have a spare 8-bit or 16-bit ISA slot (See below) and have a minimum graphics resolution of EGA (Enhanced Graphics Adaptor) so that it can display the normal QL Display on screen.

INSTALLATION

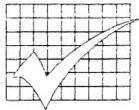
The picture on the front cover shows what you get when you buy a QXL. There is the QXL card itself together with the SMSQ software on disk and a small instruction sheet. The instructions are very simple and detail how to printout the rest of the instructions contained on the DOS disk supplied. These need to be printed off before the QXL is installed. Because of the frequent changes in the software a manual has never been produced but this is being rectified now. I have seen the pre-release version of the manual and it details all the new features of the QXL card and SMSQ with examples.

The installation of the QXL hardware is simply a matter of opening up the PC and locating an empty ISA slot and pushing the card

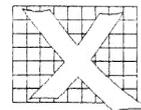
without further alteration. If this does need to be changed full instructions are given in the instruction sheets. These are very well written for the PC layman, they even detail the difference between an 8-bit and a 16-bit slot (A 16-bit slot consists of two connectors in line, one slightly longer than the other; an 8-bit slot consists of just one connector).

Once the hardware has been installed the PC can be turned on and the software to make the QXL run can be installed. This can be installed on the hard disk of the PC or it can be run from the floppy drive. To start the QXL you type in QXL which runs a PC EXE file to load and swap to the QXL card.

If there is no address clash with any other PC cards you will be presented with the familiar three QDOS Monitor SuperBasic windows (Mine worked first time). If there is a clash you can change the board address with the instructions given in the instructions. However this is not QDOS this is SMSQ.



QXL



To switch back to the PC you can press the key combination CTRL + SCROLL LOCK. This still leaves the QXL working and you can then swap back to SMSQ by typing in QXL/ which will restart the QXL from where you left off.

SMSQ

SMSQ is based on the SMS Kernel which has been modified to provide a degree of QDOS compatibility. The Kernel is larger in size than QDOS but it is still more efficient than the old QDOS Kernel.

SuperBasic has been replaced by SBASIC which runs much faster, in fact at speeds approaching that of compiled programs. SBASIC can now be EXECuted directly and multiple copies can exist.

SMSQ contains all the QL SuperBasic commands plus the TK2 commands although some commands have been extended to provide new enhanced features.

One nice feature is the ability of SMSQ to recognise "foreign" disks. This means you can directly read

Using Quill you can read these files without the need to first transfer the files on to a QDOS disk.

Two examples are shown on this page. The first is linedesign being run in SMSQVGA mode, this shows how impressive a large screen



SMSQ is also able to utilise the extended screen resolutions the PC is capable of. Four preconfigured versions of SMSQ are given to run in 4 different screen resolutions. These are:-

- SMSQQL - 512*256 in mode 4
- SMSQEGA - 640*350 in mode 4
- SMSQVGA - 640*480 in mode 4
- SMSQSVGA - 800*600 in mode 4

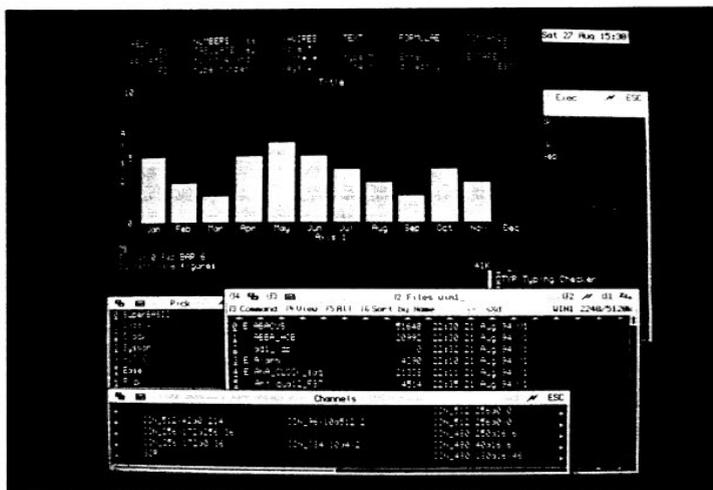
resolution can be. The second screen shot shows the same resolution. This has easel showing the normal QL screen size with several QPAC2 programs running in the extra screen areas.

I will be talking more about SMSQ, SBASIC and compatibility in the next issue of the magazine.

HARDWARE UTILISATION

The QXL has access to the hard disk of the PC by the command FORMAT "WIN1_20". This would create a file on the PC hard disk of 20 Megabytes. WIN1_ will then be just like any other device except larger.

At present you can only create one hard disk partition up to a maximum size of 63MB. I would like to see this feature extended to allow additional hard drives to be created to be named WIN2_ etc.



other formats of disk other than QDOS. A good example is the DOS disk that contains SMSQ, on this disk are several Quill DOC files.

Only Pointer Environment programs written for larger screens are able to utilise the extra screen size fully.

More next issue

Bruce Nicholls

QUBBESoft P/D

38, Brunwin Road, Rayne, Braintree,
Essex. CM7 5BU. Tel/Fax: 0376 347852
24hr - 7 Day Week Service.

PD & Shareware Software

QL Emulator AMIGA V3.21 (3 Disks)	£3.00p
C68 Compiler Runtimes (3 Disks)	£3.00p
C68 Compiler Source Code (2 Disks)	£2.00p
MicroEMACS V3.11 (2 Disks)	£2.00p
GNU Text Utilities (2 Disks)	£2.00p
Adventure '93 (2 Disks)	£2.00p
XLisp V2.1d (2 Disks)	£2.00p
STAR TREK Reference (2 Disks)	£2.00p
SPECIAL 1 Q.A.T.S.	£1.00p
SPECIAL 2 Molecular Graphics V5.11	£1.00p
SPECIAL 3 XLisp V1.4a	£1.00p
SPECIAL 4 QPACER	£1.00p
SPECIAL 5 'C' Tutorial	£1.00p
SPECIAL 6 C68 QPTR Comp V3.02	£1.00p
SPECIAL 7 CPORT Support Library	£1.00p
SPECIAL 8 QRactal Screens Demo	£1.00p
SPECIAL 9 QL Home Finance	£1.00p
SPECIAL 10 ZM1+ Spectrum Emulator	£1.00p
SPECIAL 11 Lib CURSES V1.20	£1.00p
SPECIAL 12 ELVIS Editor V1.6	£1.00p
SPECIAL 13 Text 'N' Graphics Demo	£1.00p
SPECIAL 14 'C' Programming Tools 1	£1.00p
SPECIAL 15 'C' Debugging Tools	£1.00p
SPECIAL 16 Ergon Dev Demo Disk	£1.00p
SPECIAL 17 Ger to Engl Word List	£1.00p
SPECIAL 18 QDesign Demo	£1.00p
SPECIAL 19 D.J.Walker Demo Disk	£1.00p
SPECIAL 20 DBAS Database Package	£1.00p
SPECIAL 21 FTC II Demo	£1.00p
SPECIAL 22 QL Hard Disk	£1.00p
SPECIAL 23 PSION Xchange	£1.00p
SPECIAL 24 SPECULATOR '93	£1.00p
SPECIAL 25 QLIPER Disk	£1.00p
SPECIAL 26 Line Design 2 Demo	£1.00p
SPECIAL 27 Compu-Clips 1	£1.00p
SPECIAL 28 Bison V1.21	£1.00p
SPECIAL 29 Flex V2.3	£1.00p

P.D. Disks No's 1 to 14 £1.00p each

QITALY Disk Mag Issue 15 to 25 £1.00p each

All prices Include Medium but not P&P. Please refer to P&P Guide. Medium is 3.5in DS/DD.

New QL Hardware

Fast-Net: Local Area Network, enabling connectic between QL & QL, QL & ST, ST & ST (ST must be running QDOS). Fully compatable with existing hardware expansion systems (Gold Card, Trump Card, Super Q Board etc etc). Runs approximately 9 times faster than existing QL Network. Plugs into QL or ST ROM slot via ribbon cable and tongue. Can utilise existing QL Network Cable. Software includes File Server. Full Documentation included. Minimum Configuration 2 Units £120.00p

Each Additional Fast-Net Unit: £ 65.00p

Trump Card: 768K RAM, Toolkit 2, Disk Interface, Dynamic RAM Disk, Printer Buffer. Includes A4 Manual £ 95.00p

ExpandeRAM 512: 512K RAM with through port to connect other peripherals i.e. Disk Interface, Eprom Board £ 45.00p

ExpandeRAM 0: 0K RAM, as above except User supplies RAM chips £ 20.00p

Twin 720K Disk Drives: Fully Cased with internal Power Supply Unit £100.00p

Twin 1.44mb Disk Drives: Fully Cased with internal Power Supply Unit £150.00p

Twin 3.2mb Disk Drives: Fully Cased with internal Power Supply Unit £200.00p

1 Year Warranty on all New Hardware

(P&P: UK + 5%- Max £5.00p, Euro + 10%- Max £10.00p, R.O.W. + 20%- Max £20.00p)

New Hardware in Pipeline

QUBIDE: IDE Hard Disk I/Face, allows connection of 16 bit IDE Hard Drive, upto 120mb, to the QL. Plugs into ROM slot. Software built in.

Payment Methods

Cheques (in UK pounds drawn on UK Bank), Postal Orders, International Postal Orders & EuroCheques (in UK pounds) payable to:- QUBBESoft P/D
Cash in UK pounds is also acceptable.

P&P Guide for PD & Shareware

Qty	UK	Euro	Z1	Z2	Qty	UK	Euro	Z1	Z2
	£ p	£ p	£ p	£ p		£ p	£ p	£ p	£ p
1-3	0.50	1.00	1.50	1.75	13-15	1.50	3.00	3.50	3.75
4-6	0.75	1.50	2.00	2.25	16-18	1.75	3.50	4.00	4.25
7-9	1.00	2.00	2.50	2.75	19-21	2.00	4.00	4.50	4.75
10-12	1.25	2.50	3.00	3.25	22-24	2.25	4.50	5.00	5.25



HELP DESK



I have a 720 sector disk drive. Is it 40 track or 80 track - how can I tell? (Mark Shaft, Leicester)

It could be either. Both a 40 track double sided or 80 track single sided drive would format a disk to 720 sectors. The easiest way to tell is to FORMAT a disk with the special 11 character names available on most disk systems. First of all, format a disk normally, e.g. with the command `FORMAT FLP1_1234567890` (a name of 10 characters or less). Note the number of sectors printed, which would usually be 720/720 sectors. Now reformat with this command: `FORMAT 'FLP1_1234567890*'` (note the asterisk as the eleventh character inside the apostrophes. If the resultant sector count shown is still 720/720, this means that formatting single sided made no change, so it is likely to be an 80 track single sided drive. But if the resultant sector count is halved (360/360 sectors), it was a double sided drive so must have been 40 track double sided. Most people with 720 sector drives assume they are 80 track single sided, but experience has shown that QL users are more likely to have 40 track double sided drives than 80 track single sided. If in doubt, avoid these types of drives since many software suppliers for example will not be able to supply on these two formats.

How can I add a mouse system to my QL? (John Staples, Nottingham)

There are four main types available. The original system was the QIMI, now available from the user group, Quanta (see page 8). It consists of a small circuit board which is installed inside the QL. You can then plug in an Atari style mouse. Generally speaking, it will work only with pointer driven software such as QPAC2. The second system is the Serial Mouse, by Albin Hessler. This consists of a disk of software and adaptor lead for a PC style mouse to plug into a serial port on the QL. Again, this is mostly used with pointer driven software, but more recent versions can copy the cursor and space/enter key presses of the keyboard too, making it useable to some extent with older cursor-key or joystick operated software. This is available in the UK from Dilwyn Jones Computing (see page 12). Thirdly, Adman Services (0952) 255895, 53 Gilpin Road, Admaston, Telford, Shropshire, TF5 0BG supplies another PC mouse and software written by Simon Goodwin. This one can operate like cursor key presses, though more recent versions are said to work with pointer driven programs too. The fourth type is a pure cursor key copier mouse which plugs into the CTL (joystick) ports on the QL, called the Mersey Mouse, produced by members of the Merseyside QL user group. You may also come across an older mouse system called ICE. This is no longer supported by software developers.

I have some programs where the instructions tell me to use 'Config' to

configure certain aspects of the software. What does this mean? (Colin Lamb, Somerset)

CONFIG is a program supplied with many 'pointer driven' programs, which are controlled by moving a small pointer (usually a little arrow) across the screen to point to an item on the screen. These programs usually have some special information built in to tell the program which disk drive to use, or what type of printer is supported, for example. By starting the program called CONFIG (usually with the EXEC or EXEC_W commands) and telling it to load the details of a program you specify, you can look at this information and change it as required. Since CONFIG makes actual changes inside the program itself, always do this on a copy of the original program (keep the original master disk safe in case something goes wrong and you have to make another copy).

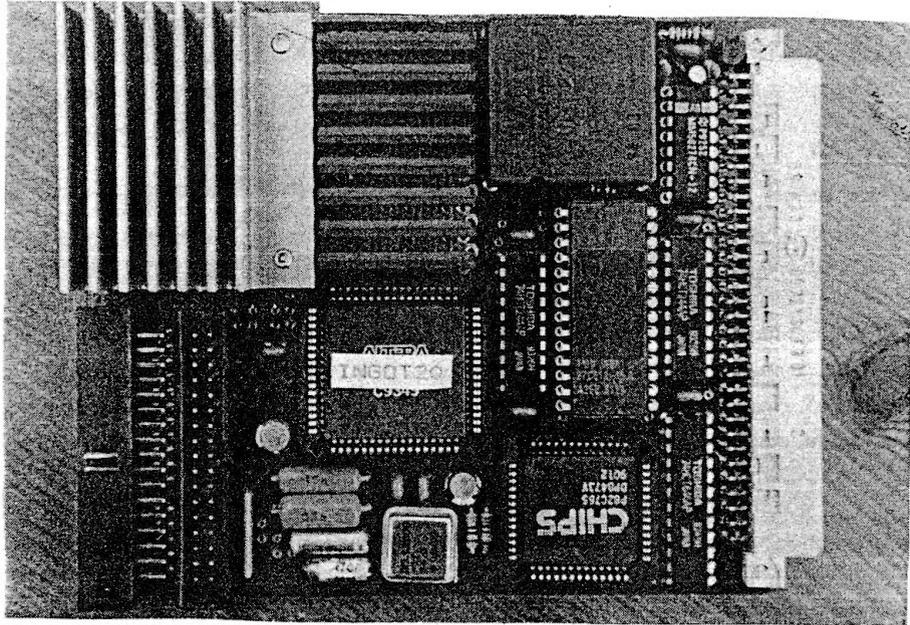
LATE NEWS

Software87 has produced a new version of Plus4, their wordprocessor. The new version (V4.1) contains several improvements on V4 and also has an integrated manual. A special version of Plus4 is also available for SMS2 users. For further details contact Software87 (071) 4859008, 33 Savernake Road, London, NW3 2JU.

MIRACLE SYSTEMS

SUPER GOLD CARD

"The Pathway to Future QL Development"



Briefly...

- * 3 Times Faster
- * 68020 processor
- * 4M bytes of RAM
- * CENTRONICS port
- * 2 Disk drive ports
- * 2 Year warranty
- * £375 (£325)

What is it ?

This is the first major revision of our highly successful Gold Card. We have replaced the 68000 with a 68020 so programs run about 3 times faster and have expanded the memory to 4M bytes.

Other improvements include a fast CENTRONICS printer port, 2 double disk drive ports, virtually crash-proof clock and a socket to optionally connect 5V. We also are providing a Centronics printer cable at no additional cost.

The deal...

The price is £375 inc. VAT (£325 outside EU) which covers postage, 2 year warranty and 14 day money back guarantee. We can upgrade your Gold Card for £225 (£200 outside EU), or trade in your Trump Card (£50), Super Q Board (£40), or other memory expansion (£25) against the purchase price.

Additionally, you can trade in your QL Centronics (£15) or Disk Adapter (£10) or both.

We are happy to accept payment by sterling cheque made payable to "MIRACLE SYSTEMS", or by quoting your MASTERCARD/VISA/SWITCH credit card number and expiry date (SWITCH card holders please also quote issue number).

Order From...

MIRACLE SYSTEMS LTD
20 Mow Barton
Yate, Bristol BS17 5NF
UK

Recycled Items...

Gold Card	£150
QL Centronics	£ 15
Disk Adapter	£ 10

(Recycled items carry a 1 year warranty.)

TELEPHONE/FAX: (0904) 423986

MIRACLE SYSTEMS

QXL

*See below for GOLD CARD
upgrade deal !*

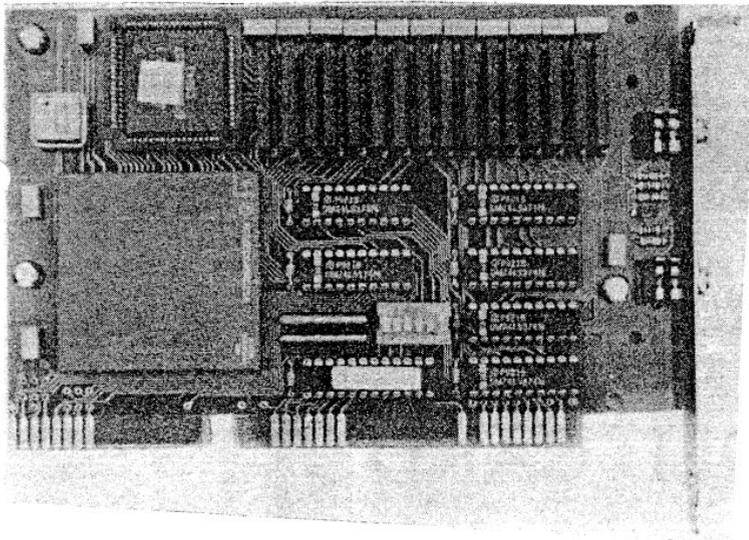
Now With

Call for other upgrade paths.

SBASIC

*Our NEW address:
MIRACLE SYSTEMS LTD
20 Mow Barton
Yate, Bristol BS17 5NF
UK*

SuperBasic Compatible Interpreter



- * 68EC040
- * 4M or 8M of RAM
- * Multitasking SBASIC
- * QL Network ports
- * Toolkit II
- * QDOS or MSDOS floppies
- * Uses PC's keyboard, floppy & hard disks, parallel/serial ports and mouse.

This is the card that plugs into a standard 8 or 16 bit ISA slot on a PC and allows the PC to run QL programs - FAST. A new QDOS compatible operating system from Tony Tebby called SMSQ, which is supplied on a disk, includes Toolkit II and gives you the familiar QL environment. SMSQ includes SBASIC a multitasking SuperBasic compatible interpreter.

Installation is simple; plug the QXL into a spare slot and copy 2 files from the supplied disk onto the hard drive and you're ready to go. From the DOS prompt type QXL and the PC will transform itself into a QL before your very eyes. If at any stage you wish to return to DOS just press CTRL-ScrollLock. You can later resume the QL session by typing QXL/ which takes you back to where you left off.

For POINTER ENVIRONMENT programs SMSQ can be configured to handle 3 screen resolutions in addition to the standard 512x256 QL screen. Your PC must have EGA or VGA graphics. EGA allows 640x350 whereas VGA also allows 640x480. Most SVGA cards will allow SMSQ to use 800x600 as well.

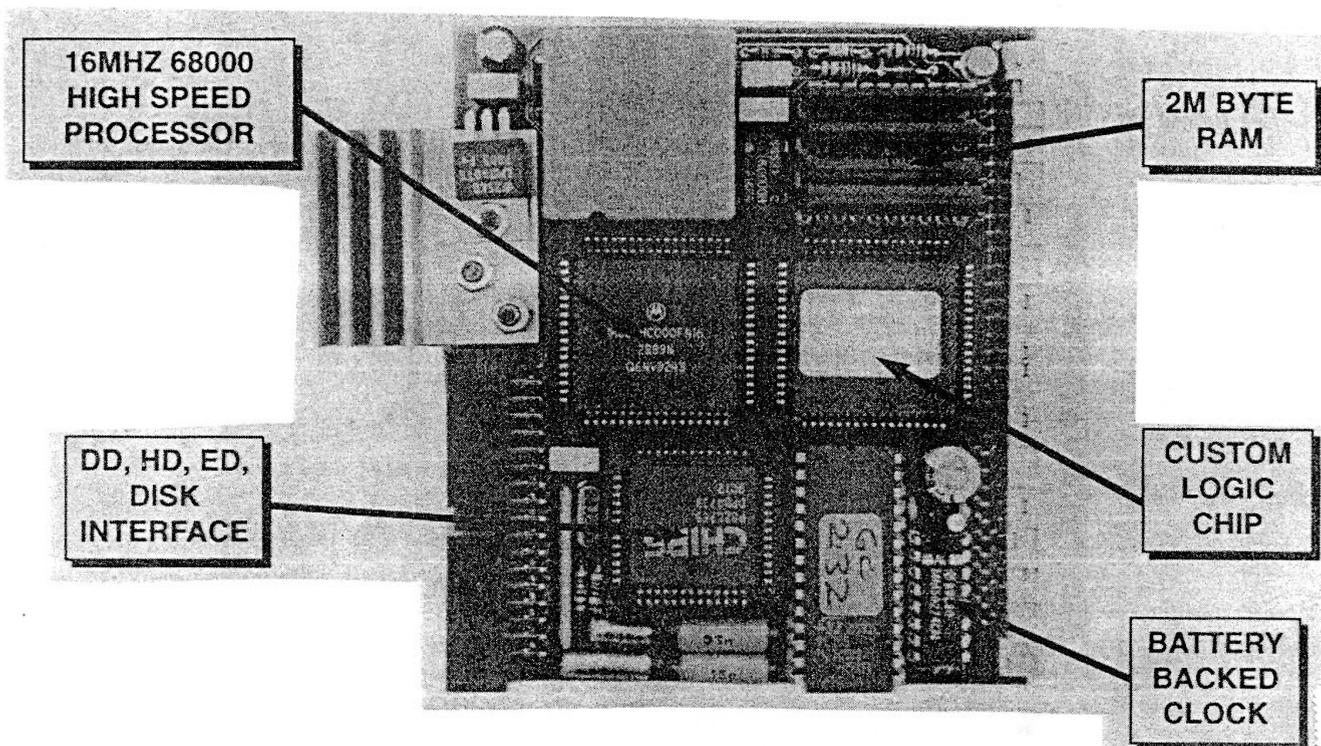
A QXL fitted with 4M bytes of RAM costs £380 including VAT (£330 outside the EU) and the 8M byte version is £495 (£430 outside the EU). Prices include postage, our 2 year warranty and a 14 day money back guarantee. Software updates are supplied free of charge and sent out automatically.

Alternatively send us your GOLD CARD plus £230 (£205 outside the EU) for a 4M byte QXL or send us your GOLD CARD and £345 (£305 outside the EU) for an 8M byte QXL.

We are happy to accept payment by sterling cheque made payable to "MIRACLE SYSTEMS", or by quoting your MASTERCARD/VISA/SWITCH credit card number and expiry date (SWITCH card holders please also quote issue number).

TELEPHONE/FAX : (0904) 423986

MIRACLE SYSTEMS



QL GOLD CARD

Recycled Gold Card £150 inc. (£125 outside EC)

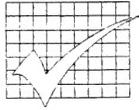
This is the expansion that has been revolutionising the QL. It is very easy to fit, it simply plugs into the expansion port at the left hand of the QL, and once fitted it will instantly increase the execution speed of the QL by about 4 times due to the presence of a 16MHz 68000 on board. There is 2M of fast 16 bit RAM of which QDOS sees a contiguous 1920K. The remainder is used for shadowing the QL's ROM and display memory and for the GOLD CARD's own code.

There is a disk interface which can access 3 mechanisms (4 with the DISK ADAPTER) of three different densities, DD (double density, 720K), HD (high density, 1.44M) and ED (extra high density, 3.2M) in any mix. The disk interface connector is the same type that was fitted to the Trump Card so most QL compatible disk drives can be used.

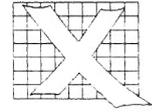
Please note: that DD drives still give a capacity of 720K per diskette.
Our DUAL ED DISK DRIVE allows the GOLD CARD to access DD, HD and ED diskettes.

Another feature is the battery backed clock. When the QL is switched on the contents of the clock are copied into the QL's clock so that the time and date are correct. The firmware in the ROM gives the GOLD CARD all the functionality of the Trump Card like TOOLKIT II and there is a sub-directory system for floppy and RAM disks.

Physically the GOLD CARD is about half the size of the TRUMP CARD and so fits almost all within the QL. Its current consumption is well under allowable maximum so no special power supply is required. The GOLD CARD comes with a 14 day money back guarantee and a 1 year warranty.



DEA



This article is a review of version 5.21 Plus 3 (what does that mean ?) of the DEA disassembler, written by Marco Ternerli.

Many reviewers get supplied with a free copy of the program under review and get to keep this copy once the article is done [our normal policy] - in most computer magazines that is. This review is of a copy of the program that I actually paid for with my hard earned cash. I bought the program to assist me in a fairly major job I am doing for one of the larger QL Software companies - look out for forthcoming releases. No names yet !

I already have a number of disassemblers, and while each one is good at what it does, DEA has to be the most up to date and complex beast of them all. I use each one where I think it is applicable, they are tools after all, to be used as and when desired - but the right tool for the job.

DEA comes with a 52 page manual that was written by an Italian, Davide Santachiara of Ergon. The translation has been improved by RWAP Services. The manual is not bad as manuals go, I have seen a lot worse, but I think it is a bit thin for the complexity of the program. Perhaps more tutorial could have been given for new users.

That apart, I have no problems with the manual, Looking at it in comparison with the actual program, I think that it would be quite a horrendous task to write a manual that was able to cover everything in depth and still be usable at the same time.

Enough said about the manual, on to the program. DEA is a Turbo compiled SuperBasic program, the file is approximately 194 kilobytes long and one can only try to imagine the size of the original source code !

As disassemblers go, DEA knows almost all there is to know about QDOS and the QL. It knows

about the system variables, SuperBasic variables, Things, QDOS vectors, Traps both QDOS and SMS and QPTR, ROMs, ROM image files, Error codes BP_INIT tables etc etc etc and when it comes across one in the code, it can track the parameters and use them to find where the structures used are held and disassembles these correctly as well. Basically, DEA does about 98% of the work involved in decoding a machine code program back to an assembly language source file.

There are a couple of things that upset DEA, one of these being self modifying code (nasty stuff) such as Turbo compiled programs, but on the whole, DEA can cope with just about anything you care to throw at it. Some programs, such as compiled C or Qliberated SuperBasic programs will need some work on your behalf after the source has been produced, but a lot less than with other Disassemblers.

DEA comes with a boot file that loads up 2 extension files and then EXECs the program. The first extension file is the usual Turbo Toolkit Runtimes and the second is a smaller toolkit containing a number of base conversion functions. If you normally have Turbo Toolkit loaded, you probably won't need the first file. If you have Toolkit 2 activated, you will not need the second.

Once into the program and past the initial screen, you are presented with a small menu that asks where to find the code that is to be disassembled, the options here are : In Memory, On a file or In a ROM. The option to quit is also allowed here.

DEA allows just about anything to be disassembled, with the above caveat on self modifying code. One very handy facility is the ability to get a directory listing, simply by typing in the device name and a question mark when prompted for a filename, as DEA doesn't use

the QMENU extensions.

DEA uses its own inbuilt menus and non-destructive windows. Menu options can be selected by moving the cursor up and down until it is positioned over the choice you want, then pressing the SPACE or ENTER keys. If the menu has an 'S' in the top left corner of its title, it will allow on a Single choice to be made. Multiple choice menus have an 'M' in the top left and these will not actually do anything until you get around to pressing ENTER. The SPACE key simply adds another choice to the list.

In both types of menu, a choice can also be made simply by pressing the letter or number shown at the start of each entry.

The main menu is first to be shown, and is a single choice menu. This gives you the choice of how the disassembly will proceed. Among the options, the one that DEA thinks is the most suitable will be highlighted for you - this helps reduce the number of key presses a user has to make in order to get from binary code back to source !

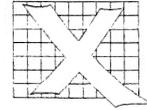
In most cases, the option has to be MULTI-LEVEL PARSING, as this is where all the hard work is done. In this option, DEA has up to 7 levels of parsing where it checks through the code it is disassembling to see where it can fill in the blanks. As it goes, the memory map is kept updated, provided you have this option turned on, of course.

The memory map is a small window which displays the progress of the disassembly. A line of pixels represents the entire code file, and each pixel represents a fraction of that code, under the map, an information section tells you how much code is represented by a single pixel.

As the decode progresses, more and more of the pixels are filled in and from time to time, change colour. White pixels are bits of the file being decoded, green ones



DEA



show which bits of the file have been completely decoded and black is still to be done.

DEA can follow the code through, as if it were executing it, or it appears that way, keeping a note of routines called in the code or in QDOS. Any routine called by the program that DEA has internal knowledge about, will also have any relevant data structures decoded as well. For example, a call is made to TRAP #2 with D0 = 1, to open a file, DEA knows that the trap requires a parameter in A0 and so it will look for the filename preceded by its word length and decode that automatically. Other internal structures are known about and decoded as required. This takes much of the work out of decoding a program.

Each of the 7 levels of parsing carry out a specific task. Some levels are carried out once only, others might be carried out many times. The seven levels are :-

Level 1, Main Parsing - this is performed once only and starts at the address you have specified as the start of the code to be executed, and performs a dataflow analysis on it following the calls to subroutines, branches etc in the program.

Level 2, Config Data - after level 1 parsing has been done, DEA will look for CONFIG data blocks in the program. If any are found, they will be decoded as they are found.

Level 3, Undecoded Label scan - If there are areas of code not decoded by any previous levels of parsing, this level tries to decode each as proper code. If it comes across an area that it cannot recognise as code, it assumes data, but more of this in level 6. DEA then carries on with these new bits of code and attempts to parse them as well. An undecoded label is an address in the code file that has been referred to in a previously decoded area, but which cannot be automatically understood. For example, some code carries out an

LEA.L address, A0 and has been decoded by DEA. At this stage, there may be no clues in the previous or following instructions to give DEA a hint as to what the code or data at 'address' actually is. DEA will try to decode it as assembly code, and if there are no errors, will assume that it was code - the user has not been asked for help yet.

Level 4, Skipped Address Scan - once DEA has finished at level 3 and has no more labels to examine, it searches for bits of the file that it currently has no information about. It tries, once again, on its own, to classify these areas as code or data. Any areas of code, lead to the parsing process getting going again and this could lead to some more skipped labels, so, DEA starts another level 3 scan. DEA will happily switch between level 3 and 4 until it can do no more.

Level 5, Search New word strings - Any area in the memory map that are still black could contain data. DEA now scans these areas for QDOS word strings. These being a word giving the length of the following string, followed by the characters of the string. This option can request user intervention. There is a menu option to ask for confirmation of any strings found. Assuming that at least 1 string has been found, DEA goes back to level 3 to start parsing again. If not, level 6 is entered.

Level 6, Retry Labels - this is similar to level 3, but this time, DEA is looking for data, not code. At this stage, DEA should have decoded a fair part of the file. If any areas are now decoded, level 3 is started again to try to get some more out of the file. If not, DEA could need some help.

Level 7, Search New Code Start - this is where the user finally has his or her say in the decoding process. DEA has done as much as it can, but now needs help. Any black areas in the memory map are examined and the screen shows a

decode as data on the right and a decode as code on the left.

The user then can move the cursor around in either code or data to show DEA where the next bit of parsing is to begin from. This is basically what you would be doing with other disassemblers which do not have DEA's intelligence. You identify to DEA what each of the unknown parts of the code file are, and DEA takes that information and then starts parsing again.

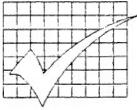
For each part that you tell DEA about, it will then parse that are according to your instructions. Code areas will cause the multi level parsing to be carried out until DEA gets stuck again. Once more, you supply a clue. This goes on until you abort DEA or there is nothing left to decode. You can now select to generate the full disassembly from the main menu.

In addition to being able to completely disassemble a program file, DEA can be used to extract certain routines from toolkits or from the QL's own ROM. As with a normal disassemble, you specify where to find the code and its start address etc. From the main menu, enter the Keyword Manager.

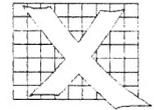
Another menu is shown and the most logical choice is highlighted, this being to 'Edit Keywords'. At this point, you type in the name of any Procedure or Function that you wish to disassemble or extract.

This is an area that caused me some trouble at first. DEA will ignore you if your code start address and length does not cover the start address of the various keywords that you wish to examine. To extract from the QL's on ROM, the start address is 0 and the length is 48K (DEA accepts the 48K or 49152 value, very handy).

Another problem I had, was that the prompt asks for a keyword to be prefixed with an 'at' sign (@) if already resident. If you do this, the keyword will not be included for



DEA



disassembly. What will happen is that you will be given its starting address in the ROM. This is useful for the main disassembly process, but confused me for a while ! (Read the manual properly !!!!)

Type in all the various procedure and or function names and tell DEA whether they are procedures or not. DEA assumes a function for any keyword that is not a procedure.

If you enter a keyword that starts in the code area that DEA is examining, you will be told its address and the name will be added to the list of keywords being looked at if you specify that it is to be kept track of.

DEA must be given the keyword in the proper case. 'dir' is not the same as 'DIR', beware.

Pressing ENTER only in response to a prompt for a keyword name stops the input and you are asked if you wish to parse the keywords you have selected. If you reply Y, the parsing setting menu is displayed with most of the options highlighted and, as usual, the most logical choice selected. Simply press ENTER to extract the keyword code. It really is this simple !

At the end of the parsing, you can choose from the Keyword Manager menu, to create a file of source code. This option is the most logical choice, and DEA is just waiting for you to hit ENTER or SPACE to select it. The output from DEA can be sent to the screen, a file or a device such as a printer.

The source code generated can then be reassembled to produce a smaller version of the original toolkit but you should be aware that this must only be for your own use - extracting other people's code for commercial use is in breach of copyright. You must have permission from the owner of the copyright to use their code for any purpose.

The generated source can be displayed on the screen, but, if it is a

long file, don't ! The ESC button does not seem to be scanned for, so the whole disassembly must be listed before you regain control.

This review covers only a few basics of the program. A full review would take up far too much space in the magazine and keep me busy for weeks.

I can say that I like DEA a lot, I am glad to have paid for it and it is very, very useful in what I do. It has its foibles, and the ones I have noted are :

Generating code to the screen, ESC cannot be used to abort the code generation. I don't particularly like the need to enable the 'ESC while parsing' option so that I can escape from a large decoding session. But once found, it is easily set.

When entering numbers for addresses, sizes etc, you can enter them in decimal, hex or as a number followed by K to signal kilobytes. The one place this doesn't seem to be liked is when altering the code length as DEA simply buzzes at you if you use the 'k' format.

Sometimes when decoding a file, during the level 7 parse, DEA shows a totally blank screen then puts up the menu for me to select a code or data style ! As nothing has been shown, how can I tell ?

I am sure that in a program as vast and complex as this one is, there could be a few bugs still to be found and ironed out. I think that anyone who is serious about developing code on the QL, or its derivatives, would do well to obtain a copy of DEA. If you do, be prepared to put a bit of effort into using it. It is not a simple program to use but it is well worth learning how to use it.

The manual is quite good and there is a small, perhaps too small, tutorial section which shows how to disassemble the runtime Turbo toolkit. DEA doesn't take long to do it either and very little effort is required. Another part of the

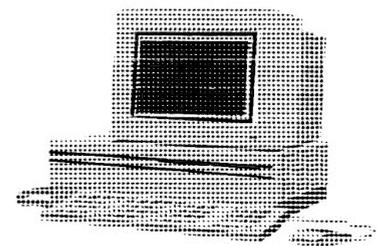
tutorial shows how to decode a file about which you have no knowledge.

There are some programs it doesn't like. I threw a C68 compiled program at it which caused it to get in a knot, but then again, did I tell it something about a piece of the file that might not have been quite correct ?

In order to get a further plug in for DJTtoolkit, DEA completely disassembled it in 40 seconds without needing any input from me. The final file was approximately 46 kilobytes long and matched up to the original, even down to bits in the file that are not called or used in any other part of the toolkit. DEA found them all, I'm impressed. I am off to see how it can handle Psion Chess.

Norman Dunbar

DEA costs £26.00 (+£1 for non UK countries) and is available from Dilwyn Jones Computing (0248) 354023, 41 Bro Emrys, Tal-Y-Bont, Bangor, Gwynedd, LL57 3YT, UK or Ergon (x-39-522-70409) c/o D.Santachiara, Via Emilio De Marchi 2, 42100 Reggio Emilia, Italy.



TF SERVICES

MINERVA

The ULTIMATE system upgrade

MINERVA RTC (MKII) + battery for 256 bytes ram, CRASHPROOF clock & I²C bus for interfacing. Can autoboot from battery backed ram. Quick start-up.

OTHER FEATURES COMMON TO ALL VERSIONS

DEBUGGED operating system/ autoboot on reset or power failure/ Multiple Basic/ faster scheduler-graphics (within 10% of Lightning)-string handling/ WHEN ERROR/2nd screen/TRACE/ non-English keyboard drivers/ "warm" fast reset. V1.97 with split OUTPUT baud rates (+ Hermes) & built in Multibasic.

1st upgrade free. Otherwise send £3 (+£5 for manual if reqd) . Send SAE, Minerva + disk/3 mdvs.

MK1... £40 RTC (MKII)... £65

GOLD CARD (incl SUPER) compatible

HERMES

A replacement QL co-processor for the QLs awful IPC 8049

- Do you get keyboard bounce?
- Do you find fast serial input unreliable?
- Do you want to connect a faxmodem at 19200bps and send and receive FAXES and/or data.

If you can say one YES , then you need HERMES

- 19200bps RELIABLE serial input - NO QCONNECT.
- Independent input baud rates - use serial mouse & print
- Stops keyboard bounce (unwanted repeat chrs)
- Improves 'fuzzy' and 'random' sound
- Provides extra input/output lines
- Key click

To fit, simply replace the QL 8049 or 8749 chip

£25 including manual/software

QL SPARES

Faulty QL board (no plug-in chips).....£9

Keyboard membrane£9	Circuit diagrams£2
68008 cpu£8	1377 PAL£3
JM ROM.....£10	Power supply£12
8302 ULA.....£10	8301 ULA.....£10
8049 IPC£8	MDV ULA£12

Other components/(sockets etc) please phone

I²C Interfaces

The I²C bus was designed by Philips to simplify interfacing. Minerva MKII clock is driven by an I²C chip, & a connector allows connection of other circuits. Our external circuits will interconnect without leads. Up to 5 interfaces can be powered off the QL.

Parallel Interface gives 16 input/output lines. Can be used for logic level output eg model train controllers. Input direct to motor drivers (eg L293/L298).....£25

Analogue Interface Each gives 8 analogue to digital inputs, and 2 digital/analogue outputs. For temp measurement, sound sampling etc.....£30

Data sheets. (analogue/parallel I²C chips).....£2

Control software/manual (Superbasic extns).....£2
(First interface purchase includes free 15D/9D lead)

QL REPAIRS

Fixed price for unmodified QLs, excl microdrives. QLs tested with Thorn-EMI rig and ROM software

£27 including 6 month guarantee

QBBS - UKs first QL scrolling Bulletin Board

Megabytes of files. Messages to/from UK/Belgium/Holland/USA/Germany for a UK phone call.

TANDATA callers add SIX zeros (000000) or wait for 3 seconds of modem tone if dialling manually.

0344-890987 (up to V32bis)

COMPUTER CLEANER

High Spec AC mains filters and spike removers

All in standard 3-pin devices and are plug-in
NO WIRING REQUIRED. In their 10th year of production - as old as the Sinclair QL

2-way adaptor...£14 3-way adaptor...£18
4-way trailing socket... £24

Prices include post & packing (UK only). Payment by Mastercard/Visa/Access/Eurocard/cheque/postal order/PO Giro transfer (58 267 3909). MAIL ORDER ONLY - no callers without ringing first. Ring for overseas prices.

Holly Corner, Priory Road, ASCOT, Berks, SL5 8RL

Tel: 0344-890986

Fax & BBS: 0344-890987



INTERRUPT



Using BP_LET within SuperBasic extensions.

As a follow up to my earlier articles on using the QL's Maths stack, here is a small example of how to use the BP_LET vector. For some reason, my articles seem to follow on from DJTOOLKIT (plug plug !) as a recent upgrade required that I find out more about this useful vector and now I will attempt to pass that information on to you.

BP_LET is mentioned in all the good QDOS manuals, magazines, books etc, but never seems to be explained, this article will hopefully do just that. I will also develop a very small (and very useless) procedure to show how BP_LET is used.

What does BP_LET do exactly ?

BP_LET allows you to modify the value of one, or more, of the parameters passed to a procedure or function. In most programming courses, this is treated as a no go area and is considered to be even more of a sin than using GOTO in a program.

A function should take some parameters, do something with them and return a single result without changing the parameters that were supplied. A procedure should take some parameters and carry out some form of action using those parameters. It should never change the parameter values.

BP_LET allows the rule books to be discarded and we can do anything we like.

We can write a procedure that takes one or more parameters, carry out a few actions using them and then return each parameter with a new value. That should really upset the purists !

As you may be aware, in SuperBasic, if you want to avoid having parameter values changed, you supply them in brackets as in the following :-

```
1000 REMark unchanging parameters
1005 :
1010 CLS
1015 first = 100.456
1020 second$ = 'Hello'
1025 third% = 2468
1030 :
1035 PRINT 'BEFORE'
1040 PRINT first, second$, third%
1045 process (first), (second$),
(third%)
1050 :
1055 PRINT 'AFTER'
1060 PRINT first, second$, third%
1065 :
1070 :
1075 DEFine PROCEDURE process(a, b$,
c%)
1080 a = a * 2
1085 b$ = b$ & ' Mum !'
1090 c% = c% / 2
1095 END DEFine process
```

If you type this in as it is above and run it, you should see the following on the screen :-

```
BEFORE
100.456 Hello 2468
AFTER
100.456 Hello 2468
```

As you can see, the parameters have not changed even although they were apparently altered within the procedure called PROCESS. What has happened is that the QL made copies of the variables values for use by the procedure and saved the old values while it was in the procedure. On exit, the changed values were discarded and the originals were restored, the end result is no changes.

Now remove the brackets in line 1045, the call to PROCESS, and run the program again. This time the result is :-

```
BEFORE
100.456 Hello 2468
AFTER
200.912 Hello Mum ! 1234
```

Note how the parameters have been changed ?

In the first example, the parameters were saved because the brackets around them fooled the QL into thinking that they were expressions and not variables, and as expressions cannot be changed, they were not. In the second example, the lack of brackets caused them to be treated as variables and the changes were carried out.

In assembly language, we can do similar things. Most procedures and functions are written so that any parameters supplied to it are unchanged when the procedure or function exits. This is like our first example.

We can use the BP_LET vector to cause the procedure or function to behave like our second example where the parameters are changed by the routine.

Calling BP_LET

When calling BP_LET, certain registers require to be set. These are :-

- A1.L Pointer to a tidy maths stack holding a result.
- A3.L Pointing to the name table entry for the parameter to be changed.
- BV_RIP(A6) Holding the top of the maths stack.

In order to set up a call to BP_LET, the maths stack must be tidy and holding a single value in a manner very similar to what it would be if this was a function just about to return to SuperBasic. (See previous articles on the maths stack for full details)



INTERRUPT



BV_RIP(A6) should be holding the address of the top of the maths stack as normal. Whether A1.L must be also holding the same value is unknown, but it is safe to assume that it should in accordance with normal rules.

A3 should be pointing at the name table entry for the parameter to be changed. From the name table, QDOS finds out what type of parameter this is and the result on top of the maths stack must match the parameter. If A3 points at a string, the maths stack must be holding a string etc.

All you need to do now is call BP_LET and on return check D0.L for errors. If an error occurs, it will

be BAD_PARAMETER (-15) and usually means that A3 was pointing at a parameter that did not have the same type as the value on the maths stack.

On return from BP_LET, D0 holds the error code, as described above, D1, D2, D3 are corrupted as is A0, A1 and A2. D4 to D7 and A3 to A7 are preserved. This means that if your routine requires to use BP_LET more than once, you will need to take care of all these registers.

BV_RIP(A6) is preserved so A1 can be reset from there if the maths stack is required again.

An example procedure

This will be a very simple example. A procedure will be written that will :-

1. Make sure that it gets 1 parameter.
2. Make room on the maths stack for a 4 byte string and its word size.
3. Call the MT_INF trap to get the current version of QDOS.
4. Stack the QDOS version.
5. Place the QDOS version into the supplied parameter.

The code on this page is written for the QMAC assembler, available from Quanta. It should be ok on others, but I am not a frequent user of other assemblers so I cannot be certain. The comments in the code should explain what is going on.

When you have assembled the code, it comes out around 96 bytes, load it into you QL as follows :-

```
addr = RESPR(96)
LBYTES      flp1_bplet_bin,addr
CALL addr
```

Obviously, you will replace the filenames I have used above with the ones that apply on your system. Now all you have to do is type :-

```
BP_LET a$
```

and whatever characters used to be held in a\$ are no more. A\$ now holds the 4 character version of QDOS, to see it, PRINT a\$. If a\$ did not exist at the time of calling BP_LET, it is created for you.

Should you make a mistake, and call 'BP_LET a%' instead, a couple of things will happen. First, a% will be set to 4, the word that is on the stack which holds the length of the string. Second, the maths stack would be left in an untidy state if this had been a function, a crash or

```
-----*
* A small demo of BP_LET - used to return the current QDOS version *
* in a string variable passed as a parameter. *
* Written for QREVIEW Magazine by Norman Dunbar *
* June 1994 *
* Copyright Norman Dunbar *
-----*
* Permission granted to anyone to use this code for any purpose, be *
* it public domain, commercial or anything else. *
-----*

BV_RIP    equ    $58           Offset for maths stack holder
BV_CHRIX  equ    $11A         Vector for claiming stack space
BP_INIT   equ    $110         Vector to initialise new procs
MT_INF    equ    $00         Trap key for MT_INF, trap #1
BP_LET    equ    $120         Vector for BP_LET !
ERR_BP    equ    -15         Code for bad parameter error

-----*
* The code starts here. *
-----*

        section code

start   lea.l   def_block,a1   Pointer to definition block
        move.w  BP_INIT,a2     Vector to be called
        jsr    (a2)           Do it
        rts                    Exit

def_block dc.w   1             Only one procedure
          dc.w   qdos-*       Where it starts
          dc.b   6,'BP_LET'   What it is called
```



INTERRUPT



```

dc.w 0 End of procedures

dc.w 0 No functions
dc.w 0 End of functions.

*-----*
* The procedure starts by checking that there is only one parameter *
* supplied. It must be a string, but BP_LET will check that for us *
*-----*
qdos move.l a5,d1 Pointer to last parameter
sub.l a3,d1 Pointer to first parameter
cmpi.l #8,d1 One parameter = 8 bytes in name table
beq.s param_ok

bad_param moveq #ERR_BP,d0 Oh dear, bad parameter
rts Bale out to SuperBasic

*-----*
* If we have a single parameter, A3 points at it position in the name *
* which is what we want. A1 is undefined. BV_RIP(A6) is a valid stack *
* so load up A1 with this address, claim some stack space and set the *
* stack pointer in A1 to point to the result space on the stack. *
*-----*
param_ok moveq #6,d1 6 bytes required for result
move.l BV_RIP(a6),a1 Preload stack top into A1
move.w BV_CHRIX,a2 Get the vector
jsr (a2) Get the space, no errors occur
move.l BV_RIP(a6),a1 New stack top
suba.l #6,a1 Make room for result
move.l a3,-(a7) A3 gets trashed by MT_INF trap

*-----*
* Now, call the MT_INF trap to get the QDOS version number and stack *
* it at A1. *
*-----*
mtinf moveq #MT_INF,d0 The trap we wish to set
trap #1 Go do it, no errors occur
move.w #4,0(a6,a1.l) Stack string size = 4 bytes
move.l d2,2(a6,a1.l) Stack the characters

*-----*
* Now the result is on the stack, restore A3 so that it once more is *
* pointing at our parameter's name in the name table. Then call BPLET *
* to put the value into our parameter. We are then done and quit. *
*-----*
bplet move.l (a7)+,a3 Restore our name table pointer
move.l a1,BV_RIP(a6) Set new stack top for return
move.w BP_LET,a2 Get the vector
jmp (a2) Call the vector, errors to SuperBasic

END

```

something equally as nice would be just waiting to happen. The four characters stacked for the string have been left lying on the maths stack. This is not a problem here, as we have written a procedure and procedures tidy the maths stack on exit.

If you call 'BP_LET a', using a floating point variable, you may well get a valid floating point number and the 6 bytes on the stack will be used to set its value. Some values can crash the QL if they make up a floating point value that is out of range of the QL's normal floating point range.

You must ensure that the correct type of parameter is supplied. Of course, you can check the parameter type within the procedure. On entry, A3 points to the first parameter in the name table and A5 to the first byte past the last, both relative to A6 of course. To check which type of parameter has been supplied, all you have to do is check the appropriate part of the type word, which is at 0(A6,A3.L) for the first parameter. The lowest 4 bits of the low byte of the type word hold the parameter type. These are :-

```

0000 = Null
0001 = String parameter
0010 = Floating point parameter
0011 = Integer parameter

```

So, an instruction sequence like the following should make sure that the correct parameter type is supplied.

```

move.b 1(A6,A3.L),d0
Get the type word, lowest byte
andi.b #%00001111,d0
Mask out unwanted bits
cmpi.b #1,d0
Test for a string
bne.s bad_param
Oops !

```

And that is about it. In the example procedure above, you may have



INTERRUPT



noticed that I never actually fetched the supplied parameter on to the stack. This is quite simply because I don't have to fetch the current VALUE of A\$ on to the maths stack, as I don't care about it. All I need to know is where is it in the name table and A3 tells me this.

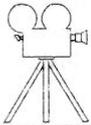
Another point to note. If I wanted to alter the values of more than one parameter, I would have used JSR (A2) instead of JMP (A2) then on

return, reset A1 to BV_RIP(A6), added 8 to A3, to point at the next parameter in the name table, stacked my next result (optionally checking the parameter type) and called BP_LET again. It is that easy.

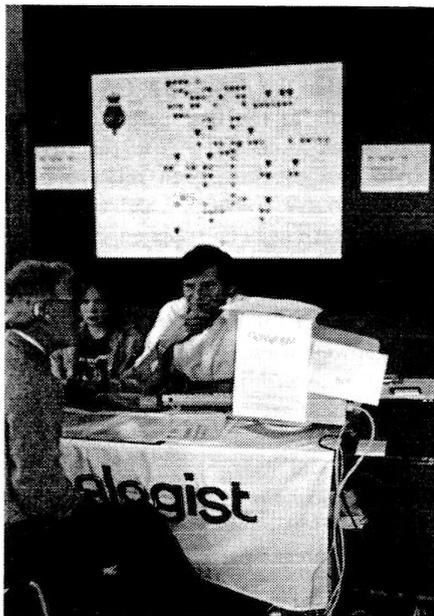
If there are any other 'bits' of QDOS that you wish to have explained, why not write to Bruce at the magazine and ask him, perhaps someone out there can come up with the information you require. I

have done articles on using and abusing the maths stack and now using BP_LET, there must be other stuff people want to know about or don't fully understand. If I can help, I will, if not, somebody else might be able to. If you don't ask, you don't find out, and besides, Bruce will be glad of the articles !

Norman Dunbar [Author of DJToolkit if you haven't guessed by now]



PICTURES AT A WORKSHOP



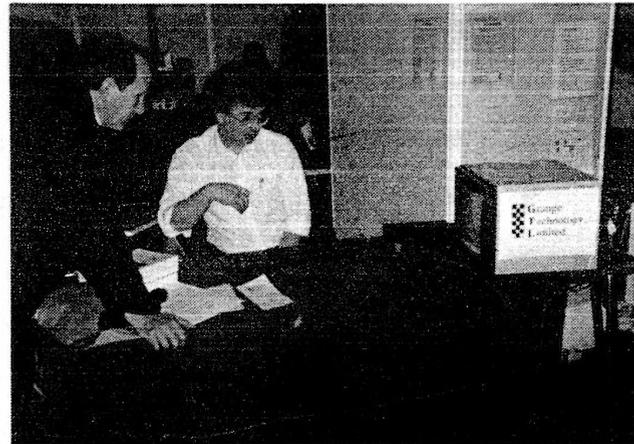
LEFT
Chris Boutal
(Geneologist)

RIGHT
Stuart Honeyball
(Miracle Systems Ltd.)

BOTTOM LEFT
Minerva I²C Demonstration
(TF Services)

BOTTOM RIGHT
Graham Thwaites
(Grange Technology Ltd.)

PHOTOGRAPHS COURTESY
OF
JIM BUIK



S.J.P.D.

Est. 1991.

36, Eldwick Street,
Burnley,
Lancashire,
England.
BB10 3DZ.
Tel. [UK] 01282 451854.

XLISP RELEASE 2.1d.
128K SPECTATOR.
SPECULATOR.
ZM1+ SPECTRUM EMULATOR.
XCHANGE VERSION 3.90.
ADVENTURE 93. (2 DISKS)
STAR TREK DISKS (2 DISKS)
COMPU-CLIPS.
MINI-CLIPS.
QL-REXX.
DBAS.
AMATEUR RADIO DISK.
FOUNTS FOR TEXT 87.
THOR COMPUTER DISK.
ESCHER GRAPHICS DISK.
DBMAN V3.6.

- * *ERGON DEVELOPMENTS DEMO.*
- * *QL HACKERS JOURNAL.*
- * *QeM COMMS PACKAGE.*
- * *Q.T.P.I. POINTER COMMS PACKAGE.*
 - * *QFAX.*
- * *LINEDSIGN CLIP-ART. (3DISKS.)*
 - * *CROSSWORD DESIGNER.*
 - * *PAGE DESIGNER 3 DEMO.*
- * *INTERNET GUIDE TEXT FILES.*
 - * *BISON VERSION 1.22.*
 - * *FLEX VERSION 2.4.6.*
 - * *INTERCAL.*
 - * *UNGIF RELEASE 73.*
- * = New Or Updated Disks.

THIS IS A SMALL SAMPLE OF
THE 100 + PACKAGES ON
OFFER.

I CAN NOW OFFER OVER 60 DISKS
OF CLASSIC LITERATURE. THESE
INCLUDES THE WORKS OF
SHAKESPEARE, HOMER, LEWIS
CARROLL & MILTON.

AMIGA QDOS EMULATOR.
RELEASE 3.21.
ONLY £3.50p.
FULLY INCLUSIVE.

PLUS 45 DISKS OF THE BEST
ASSORTED QL PUBLIC DOMAIN
& SHAREWARE

THE FOLLOWING PC SHAREWARE TITLES ARE
AVAILABLE ON 3.5" PRE-COPIED DISKS.

Z80 128K SPECTRUM EMULATOR.
XTENDER. ZX81 EMULATOR.
APPLE II EMULATOR.
C64 EMULATOR.
QDOS READER.
* *ATARI 400/800 EMULATOR.*
* *DRAGON 32 EMULATOR*
AMSTRAD CPC EMULATOR.
SHOWQS.
PSION 3 EMULATOR.
* *MACHINTOSH EMULATOR.*

MOST PC DISKS ARE £1.75p
FULLY INCLUSIVE.

QL Pre-Copied Disks Are From £1.75p Each, Inclusive Of Media And Return Postage.
I Can Also Copy Onto User Supplied Media At £1.00p Per Disk Plus Return Postage.

For A Catalogue Disk Which Details All PD & Shareware Offered And Also List
Second-User Hardware, Software, Magazines & Books Available. Send A Formatted
Disk And Return Postage Or 4 X First Class Stamps.



A Printed List Is Also Available, Send Large S.A.E. Along With
4 X First Class Stamps.

As of 29/09/94 14:20



FUNCTIONALITY

This listing is part of a set of useful procedures and functions which you can use in your library. All are in structured BASIC, so there should be no problem with line numbers in GO TO statements etc.

DECIMAL\$

A function to help with formatting numbers to given numbers of decimal places, e.g. to ensure money values always show 2 decimal places after the decimal point. It does not round off the value to the nearest decimal place, it merely ensures there is the right number of digits after the decimal point. Call it with something like PRINT DECIMAL\$(100) to print 100.00, or LET money\$='\$&DECIMAL\$(value)

ENABLE_CURSOR

Turns on a cursor in the given channel. You may have to use CTRL C before the cursor starts to flash. Useful routines when attempting to compile a basic program which uses INKEY\$, for example, to ensure that if you start a program with EXEC and use CTRL C to switch to another program, you can actually get back in! There is no cursor enabling command (apart from INPUT) in basic. But with an undocumented PAN command discovered by Simon Goodwin, it is possible to produce this facility. Both QLiberator and Turbo compilers have cursor enabling commands in their basic extensions files, so this is redundant if you have a compiler. This PAN command is not a documented feature of the QL, so although it seems to work on most current QL rom versions, it may be removed from future versions!

DISABLE_CURSOR

Opposite routine to above, to turn off a cursor in a given channel.

CONVERT\$

A function which converts a number from a given base to another number base. Useful for converting between binary, decimal and hexadecimal, for example. Handles base 2 up to base 36. Supply three parameters - the number to be converted (as a string), the base to convert from and the base to convert to. So to convert decimal 255 to binary, use PRINT CONVERT\$(255,10,2). Decimal numbers can be supplied as numbers, not strings is preferred. The routine does not add leading zeros to pad a converted string to a given length, but it should be easy enough to write a small piece of code to do this, e.g. LET a\$=CONVERT\$(value,from_base,to_base): PRINT

```

4170 :
4180 REMark return number as string with given number of
4190 REMark decimal places, does not round off values
4200 DEFine FuNction DECIMAL$(value,places)
4210  LOcAl r$
4220  r$ = value
4230  dot = '.' INSTR r$
4240  IF dot = 0 THEN
4250    r$ = r$ & '.' & FILL$( '0',places)
4260  ELSE
4270    IF LEN(r$) > (dot+places) THEN
4280      r$ = r$(1 TO dot+places)
4290    ELSE
4300      r$ = r$ & FILL$( '0', (dot+places)-LEN(r$))
4310    END IF
4320  END IF
4330  RETurn r$
4340 END DEFine DECIMAL$
4350 :
4360 REMark routine using undocumented cursor on trap to make
4370 REMark cursor flash in given channel
4380 DEFine PROCedure ENABLE_CURSOR(channel)
4390  PAN#channel,0,115
4400 END DEFine ENABLE_CURSOR
4410 :
4420 REMark this routine turns off flashing cursor in given channel
4430 DEFine PROCedure DISABLE_CURSOR(channel)
4440  PAN #channel,0,116
4450 END DEFine DISABLE_CURSOR
4460 :
4470 REMark function which returns a string containing given
4480 REMark number in other base specified (bases 2 to 36)
4490 REMark e.g. PRINT CONVERT$(255,10,16) gives FF
4500 DEFine FuNction CONVERT$(num,from,tobase)
4510  LOcAl sum,num2$,char,digit,conv$,remn,loop
4520  sum = 0 : num2$ = num : conv$ = ''
4530  FOR char = LEN(num2$) TO 1 STEP -1
4540    digit = CODE(num2$(char))
4550    sum=sum+(digit-48-7*(digit>=65)-32*(digit>=97))*(from^(LEN(num2$)-char))
4560  END FOR char
4570  REPEAT loop
4580    remn = INT(sum-(tobase*INT(sum/tobase))+.5)
4590    sum = INT(sum/tobase)
4600    conv$ = CHR$(remn+48+7*(remn>9))&conv$
4610    IF sum = 0 : EXIT loop
4620  END REPEAT loop
4630  RETurn conv$
4640 END DEFine CONVERT$
4650 :
4660 REMark make composite colour number (0 to 255)
4670 DEFine FuNction COMPOSITE(main,contrast,stipple)
4680  RETurn (64*stipple) + ((main ^ contrast)*8) + main

```

FUNCTIONALITY

FILL\$(0',required_length-LEN(a\$))

COMPOSITE

Avoids the hassle of calculating composite colour numbers from the main, contrast and pattern colour values. This combines them into a three digit number from 0 to 255, which is the normal QDOS colour value. For example, a black and white stipple of pattern 3: PRINT COMPOSITE(0,7,3) gives colour 248.

BREAK_STIPPLE

Breaks a composite colour number down into main, contrast and stipple pattern values. Try reversing the above by seeing what comes out if you enter the value 248.

SYS_VARIABLES

Many toolkits have functions to tell you the base address of the system variables area. On original Sinclair roms, the system variables were at a fixed address, 163840 (decimal). Then Minerva came along and showed us this was not always the case. Fine, if you have a toolkit or Minerva. But it would be nice to have a routine which could work on all QL versions and take advantage of the extended Minerva VER\$ function which can tell us the base address even if the system variables are moved in memory to allow use of the second screen. The Minerva rom basic version always returns 'JSL1', so we can use this in an IF statement to 'hide' the Minerva specific command from ordinary versions of basic, so that those with Minerva can find the system variables when the second screen is in use.

ADD_VAT

Adds a percentage to a given value, useful for Value Added Tax (UK) calculations. Function.

LESS_VAT

Function which returns a value minus the added VAT content, in other words, the VAT-free price.

ROUND_OFF

This function is similar to INT, but rounds off to the nearest value, for positive numbers, by adding 0.5

LOGS

A function which generates logarithms to other bases using the QL's LN function.

ODD

A simple function which returns 1 if a value is odd, 0 if even.

ROUND_PLACES

Rounds off a number to a given amount of decimal places.

```
4690 END DEFine COMPOSITE
```

```
4700 :
```

```
4710 REMark break down composite colour number to main colour,
```

```
4720 REMark contrast colour and stipple pattern number
```

```
4730 DEFine PROCedure BREAK_STIPPLE(value)
```

```
4740   main = value && 7
```

```
4750   contrast = ((value && 56) DIV 8)^(value && 7)
```

```
4760   pattern = (value && 192) DIV 64
```

```
4770 END DEFine BREAK_STIPPLE
```

```
4780 :
```

```
4790 REMark function to find base address of system variables
```

```
4800 REMark even in Minerva two screen mode, testing for 'JSL1'
```

```
4810 REMark and using an IF statement to 'hide' Minerva specifics
```

```
4820 REMark allows programs to take advantage of them where present
```

```
4830 DEFine FuNction SYS_VARIABLES
```

```
4840   IF VER$ = 'JSL1' THEN
```

```
4850     RETURN VER$(-2) : REMark Minerva system variables
```

```
4860   ELSE
```

```
4870     RETURN 163840 : REMark on AH/JM/JS
```

```
4880   END IF
```

```
4890 END DEFine SYS_VARIABLES
```

```
4900 :
```

```
4910 REMark function adds percentage to original amount
```

```
4920 REMark e.g. for VAT calculations
```

```
4930 DEFine FuNction ADD_VAT(value,percentage)
```

```
4940   RETURN value + (percentage*value/100)
```

```
4950 END DEFine ADD_VAT
```

```
4960 :
```

```
4970 REMark opposite routine to deduct VAT from total
```

```
4980 DEFine FuNction LESS_VAT(total,percentage)
```

```
4990   RETURN (total*100)/(100+percentage)
```

```
5000 END DEFine LESS_VAT
```

```
5010 :
```

```
5020 REMark round off number to nearest whole number
```

```
5030 DEFine FuNction ROUND_OFF(number)
```

```
5040   RETURN INT(number+.5)
```

```
5050 END DEFine ROUND_OFF
```

```
5060 :
```

```
5070 REMark return logs to other bases
```

```
5080 DEFine FuNction LOGS(value,base)
```

```
5090   RETURN LN(value)/LN(base)
```

```
5100 END DEFine LOGS
```

```
5110 :
```

```
5120 DEFine FuNction ODD(value)
```

```
5130   RETURN value-(2*INT(value/2))
```

```
5140 END DEFine ODD
```

```
5150 :
```

```
5160 REMark round off number to given decimal places
```

```
5170 DEFine FuNction ROUND_PLACES (value,places)
```

```
5180   RETURN INT(value*(10^places)+.5)/(10^places)
```

```
5190 END DEFine ROUND_PLACES
```

```
5200 :
```

Dilwyn Jones

IQLR

International QL Report

SUBSCRIPTION RATES:

<u>AREA</u>	<u>US \$</u>	<u>£</u>	<u>DM</u>
USA	\$20.00		
British Isles & Europe	\$34.00	£25.00	DM70.00
Canada	\$23.00		
Central/South America	\$34.00		
Rest of the World	\$40.00	£30.00	DM80.00

IQLR IS PUBLISHED BI-MONTHLY

"The Definitive Information Source"

Proven Performance Record

"Did you hear about QL
World's demise?"
"I guess it means the
QL is dead."

"Your remarks tell me that
you don't subscribe to
IQLR."

"If you did, you'd know
that the QL is alive and
doing quite well."

Readers in 33 Countries

"IQLR has kept me well
informed for almost four
years and it can do the
same for you."

REVIEWS

HINTS

TUTORIALS



ADVERTS

TIPS

NEWS

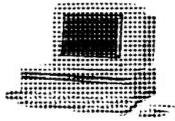
INFORMATION

PAYMENT METHODS:

We can accept £ or DM bank notes (currency) equivalent to the US \$ amount, Bank Checks in US \$ or International Postal Money Orders in US \$. As an added convenience, QL'ers in the British Isles and Europe may subscribe by sending either a Postal, Bank, Euro or Personal cheque for £25 to our European office listed below.

(Please Note: Personal Cheques in £ MUST be drawn on a UK bank. Personal Cheques in US \$ MUST be drawn on a US bank. Make ALL cheques payable to IQLR.)

IQLR * P. O. Box 3991, Newport, RI 02840-0987 * USA
Telephone/Fax : +1 401 849 3805
IQLR * 23 Ben Culey Drive, Thetford, Norfolk, * UK IP24 1QJ



MINERVA BOOT



There have been many alternative BOOT programs published in various magazines. Nevertheless, few of them utilise MINERVA's enhancements. Therefore this article is mostly for MINERVA users since it uses many of those enhancements. However, not wanting to appear ROMist, it will hopefully provide all users with some ideas on how to improve their BOOT programs.

When I began to write this suite of BOOT routines, my main concern was to load all the jobs as quickly as possible. With the JS, and earlier ROMs, this was difficult to do. Only one SuperBasic program could be run at a time. This meant that everything has to be installed in a linear fashion, or compiling certain routines to gain extra speed.

There are 2 other concurrent aims. These are to make the boot device invisible to the machine. That is, rather than assuming that all tasks would be loaded from FLP1_, the program would ask for one first. The code should also be ROM independent. This last aim conflicts with the desire for speed because it suggests that extra checks must be included in the code to cope with different set-ups.

I also prefer the long-forms of commands and spread the instructions out to improve readability. However, this adds to the length of the code, slowing loading (and processing?).

The first act of the BOOT program should be to install the TK2_EXTs and discover the boot device. There are in fact 2 possible ways to achieve this. If you have a MINERVA MKII, then you could put this code on the battery RAM; resulting in something like figure 1.

This code has 3 limitations. TK2_EXT is embedded on the battery RAM which will cause problems if you wish to demonstrate

your boot program on someone else's machine. The BOOT file must be renamed BOOT_MAIN, or similar, so that the BOOT program is not automatically loaded and run on start up. Also, there is only space for limited error checking of the device the user inputs. If you type 'WI1_' by mistake, this routine can not check for the mistake resulting in the BOOT program not being loaded.

There is a minor problem in that when the next part of the boot program is loaded, if viewing on a

Alternatively the battery RAM code with more error checks and included as part of the main boot program, looks like figure 2.

Please note that from now on, the line number ranges used are important and should be entered as printed.

Line 130 and 132 perform a little error-trapping. The reason for checking the length of DEV\$ to discover if it is seven characters long is because that is the length of DEV\$ if you are to boot across a

Figure 1

```
TK2_EXT
WINDOW 512,256,0,0:CLS
CSIZE 1,1
INPUT\,'BOOT DEVICE ( FLP1_ ) ? ';M$
IF M$<>'':PROG_USE M$
LRUN'BOOT_MAIN'
```

Figure 2

```
122 TK2_EXT
125 WINDOW #0, 512, 256, 0, 0: INK #0, 7: PAPER #0, 2: CLS #0
127 CSIZE #0, 1, 1: INPUT #0, \, 'WHAT IS THE BOOT DEVICE ( FLP1_ ) ? ';
DEV$ \
130 IF LEN( DEV$ ) = 4 OR LEN( DEV$ ) = 7 THEN LET DEV$ = DEV$ & '_'
132 IF DEV$ = '' THEN LET DEV$ = PROGD$: ELSE: PROG_USE DEV$
```

monitor, the original windows are restored, thus, our first act must be to reset the screen (line 120)

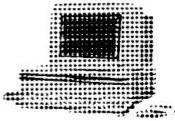
```
120 CLS: CSIZE 1,1
130 LET DEV$ = PROGD$
```

Since we have changed PROGD\$ to your choice, and have LOADED the main BOOT program, thus, resetting the variable M\$, we must return your choice to a variable; DEV\$ (line 130). Why must we do this? Rather inexplicably, the device that EXEC and LRESPR use for default devices are different. Therefore, it is simpler to use a variable to hold the default device and use it whenever we wish to LOAD, COPY, EXEC, or LRESPR a file.

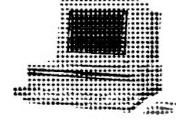
network (N1_FLP1_) and have omitted the final underscore. One could also check that DEV\$ contains a recognised device, i.e. WIN1_, FLP1_, MDV1_ etc.

We use channel #0 because Minerva MKII users may like to put the default devices into the battery powered RAM, which means that at boot time, the default devices are automatically loaded into the variables.

A main feature of this BOOT program is the use of Minerva's Multi-Basic feature to speed up the booting process. But what exactly can be run from a Multi-Basic? If a Multi-Basic has been run as an



MINERVA BOOT



EXEC'd program, then when it finishes, QDOS will delete it and any jobs that are owned by it; so you can only EXEC jobs from a Multi-Basic if you can find a way to stop the EXEC'd Multi-Basic from finishing (PAUSE -1 does not

work), and you do not mind the loss of at least 32K of memory.

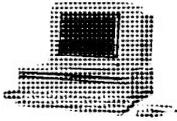
If you LRESPR an extension, then that extension is only available to that Multi-Basic and any jobs started from it. Again not very

helpful.

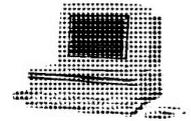
This leaves us with COPYing files and processing data. With a Gold Card fitted I started to copy help and parameter files to RAM disk. This task can be done from a Multi-

Figure 3 - Main Boot Program

```
137 LET M$ = VER$: LET V = 0
140 IF M$ = 'JSL1' THEN LET V = 1
160 LET x = RESPR( 2000 )
165 CSIZE #0, 0, 0: FORMAT #0, RAM1_500: CSIZE #0, 1, 1
170 IF V THEN
172     IF VER$(1) = '1.93' THEN
174         LET M$ = 'RAM1_MULTI_BASIC': COPY DEV$ & 'MULTI_BASIC', M$
176         ELSE: LET M$ = 'PIPEP'
178     END IF
180     EX M$;DEV$ & 'BOOT_COPY_TO_RAM>' & DEV$: EX M$;DEV$ & 'BOOT_ERROR_MESSAGES>' & x
190 ELSE
200     LET CMD$ = DEV$: MRUN DEV$ & 'BOOT_COPY_TO_RAM'
950     LET CMD$ = x: MRUN DEV$ & 'BOOT_ERROR_MESSAGES'
1200 END IF
1222 INPUT #0, \, "WHAT IS THE SECONDARY DEVICE ( FLP2_ ) ? "; PROG$
1223 IF LEN( PROG$ ) = 4 OR LEN( PROG$ ) = 7 THEN LET PROG$ = PROG$ & '_'
1225 IF PROG$ = '' THEN DATA USE 'FLP2_': ELSE : DATA USE PROG$
1227 INPUT #0, \, "NET STATION NUMBER ? "; M$: IF M$ <> '' THEN NET M$
1230 LET M$ = DATE$: LET M$ = M$( 1 TO 4 )
1235 IF M$ <> "1994" THEN
1240     MERGE DEV$ & 'BOOTS_DATE'
3000     ELSE
3010         LET Y% = M$
3020         LET M$ = DATE$: LET D% = M$( 10 TO 12 )
3030         LET M$ = M$( 5 TO 9 )
3040 END IF
3050 IF DEV$( 1 ) = 'N' THEN NET 2: NFS_USE MDV, N1_FLP1_, N1_FLP2_, N1_FLP3_, N1_FLP4_: PRINT TO 5, "MDV NOW SET TO N1_FLP"
3070 LRESPR DEV$ & 'EXTENSIONS_HERMSBAUD'
3080 CSIZE #0, 2, 1: PRINT #0, \ TO 8; " TODAY'S DATE IS : "
3090 CSIZE #0, 3, 1: PRINT #0, \ TO 8; D%; M%; Y%
3100 IF NOT HERMIS THEN
3110     CSIZE #0, 0, 0: INPUT #0, 'ARE YOU USING A PC KEYBOARD (Y) ? '; M$
3120     IF NOT M$ == 'Y' THEN MRUN DEV$ & 'BOOTS_NO_HERMES'
3130 END IF
4000 PRT_USE SER,SER
4020 CSIZE #0, 0, 0: CLS #0
4022 PRINT #0, " INSTALLED programs : "
4024 CSIZE #0, 1, 0: PRINT #0, \ " Qpac2" \ " VARIOUS SBasic EXTENSIONS" \ " FUNCTION-KEY DEFINITIONS"
4026 PRINT #0, \ \ " < alt Q > HAS QUILL" \ " < alt E > HAS THE EDITOR" \ " < alt S > HAS superBASIC" \ " < alt B > HAS multiBASIC" \ " < alt H > HAS HACKER "
4027 PRINT #0, " < alt A > HAS ASSEMBLER" \ " < alt X > HAS TURBO XCHANGE " \ " < alt D > has DATAdesign" \ " < alt L > SPELLs a file"
4028 PRINT #0, " < alt I > has Basic Reporter" \ " < alt t > SPELLs current task" \ \
4035 WINDOW 256, 200, 256, 50
4040 DO "RAM1_TASKS"
```



MINERVA BOOT



Basic. The reason for this is that where possible, I prefer the tasks that I install not to assume that associated files will be on FLP1_. I either use a config program or manually hack the program to look first to RAM1_. I consider this appropriate because I may not actually start using, say Text87, for an hour or so, and may have put a different disk in FLP1_ in the meantime, and then forget to put the BOOT disk in the drive. These files, therefore, include help-files, start-up defaults (as for TEXT87), and a Qtyp dictionary.

I have also adapted a program by Ian Malcolm which cures an obscure incompatibility between QIMI and MINERVA. This program only pokes memory so can safely be run

from a Multi-Basic. Thirdly, I have adapted a Quanta program (by Tony Tebby) called QDOS_MSG. This program allows one to change the error messages to whatever one wishes. It uses the TRA command, which I do not fully understand. The program reserves some memory, and this is done from SuperBASIC (line 160) and passed to the Multi-Basic, as described below. It is also possible to put ALTKEY and HOT_KEY commands in a Multi-Basic.

The check for ROM version (line 137 and 140) is done this way because Simon Goodwin says there is a bug in JS and earlier machines which causes problems when using VER\$ within IF-THEN lines, such as, "IF VER\$='JSL1' THEN".

The RAM disk has a size specified because I thought that this might allocate a continuous piece of memory to the RAM disk, and therefore, avoid memory fragmentation (particularly with a Trump card). However, I am not sure that this has worked as intended.

Line 160 reserves the memory necessary for the Quanta program mentioned above to alter the QL error messages. Lines 170 to 2000 are the main utilisers of MINERVA's abilities. If you have MINERVA, two programs are run from Multi-Basics. If you are running on an earlier ROM, then these two programs are merged. Thus the importance of line numbers. Wide spacing of line

Figure 4 - Boot Date

```

2020 IF DEV$( 1 ) = 'N' THEN
2040             LET TIMES$ = "N1_RAM1_TIME": LET TIME = POP_NEW( TIMES$ ): IF TIME > 0 THEN CLOSE #TIME: LET
A$ = DATE$( FUPDT( \ TIMES$ )) : DELETE TIMES$
2050             LET TIME = A$( ( " " INSTR A$( 10 TO ) + 10 ) TO ( " " INSTR A$ ) )
2060             LET TIMES$ = A$( ( " " INSTR A$ ) + 1 TO ) ( 1 TO ( " " INSTR A$( ( " " INSTR A$ ) + 1 ) TO )
- 1 ) )
2070             LET NAME$ = CHR$( 1 ) & "Jan" & CHR$( 2 ) & "Feb" & CHR$( 3 ) & "Mar" & CHR$( 4 ) & "Apr"
& CHR$( 5 ) & "May" & CHR$( 6 ) & "Jun" & CHR$( 7 ) & "Jul" & CHR$( 8 ) & "Aug" & CHR$( 9 ) & "Sep" & CHR$( 10 ) & "Oct"
& CHR$( 11 ) & "Nov" & CHR$( 12 ) & "Dec"
2080             LET D% = A$( 10 TO ( " " INSTR A$( 10 TO ) + 9 ) )
2090             LET M$ = A$( 6 TO 8 )
2100             LET Y% = ( M$ INSTR NAME$ )
2120             LET M$ = CODE( NAME$( Y% - 1 ) )
2130             LET Y% = A$( 1 TO 5 )
2140             ELSE
2150             CSIZE #0, 3, 1: PRINT #0, \ TO 7; " TODAY'S DATE IS ? "
2160             CSIZE #0, 1, 0: PRINT #0, \ TO 20; " IN THE FORM : 18 2 1990 ": LET TIME = 0: LET TIMES =
"0"
2170             INPUT #0, \ TO 27; A$: IF A$ = "" THEN LET Y% = M$: LET M$ = DATE$: LET D% = M$( 10 TO 12
): LET M$ = M$( 5 TO 9 ) : GO TO 2221
2180             LET D% = A$( 1 TO ( " " INSTR A$ ) - 1 )
2190             LET M$ = A$( ( " " INSTR A$ ) + 1 TO ( " " INSTR A$( 4 TO ) ) + 2 )
2200             LET Y% = A$( ( " " INSTR A$( 4 TO ) + 4 ) TO )
2210 END IF
2220 SDATE Y%, M$, D%, TIME, TIMES$, 0

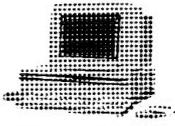
```

numbers is needed to allow for future changes.

An extra check has had to be introduced to check for MINERVA v1.97 which includes the Multi-Basic

code on the ROM and is accessed by "EXEC 'pipep'". Alternatively, Line 174 copies the Multi-Basic task to RAM disk. The file is used several times, and file access is quicker for a RAM disk. Using a

Trump card, the boot ran 2 seconds faster when I changed the boot program to use the "EXEC 'pipep'" rather than copy the Multi-Basic file.



MINERVA BOOT



Multi-Basics load the programs they are to run in this way:

```
EXEC Multi-Basic; 'program' & '>' & 'data or filename'
```

The data is put in a variable called CMD\$ and can be used or spliced as with any other variable in the Multi-Basic.

Line 1227 is not strictly necessary on non-networked machines, nor MINERVA MKII machines. The MKII battery RAM allows the NET station number to be stored and allocated on boot. It is here to provide compatibility with the Sinclair ROMs

Likewise with line 1230. MINERVA users might like to replace this line with "LET M\$ = DATE\$(TO 4)" which will produce the same value in M\$

Unfortunately, I know of no way of independently checking the machine date other than hard-wiring the program, as here (line 1235), or asking the user at boot time. You could check the complete date, but in my experience, using the year is sufficient. If the date is incorrect, the small program, called BOOTS_DATE, is loaded.

This first checks to see if the computer is being booted across a network. If so, then the date is gained from the source machine (assuming the date is correct there; perhaps that should be checked for...). This can be achieved by using the TK2 function FOP_NEW to open a new file. This returns the error code that was returned by the attempt to open a new file. If the error code is not zero then we have a problem. Assuming all is OK, then FUPDT will discover the date the file was last updated, which is just about the correct time since we have just created the file. The date is returned as a number so we must use the DATE\$ function to turn the

number into something more useful. This is an adaption of a routine that is in the Minerva manual. If one is not booting across a network, the user is prompted for the correct date. This is my most unreliable piece of code because it has rarely been tried in action (note the use of 'GOTO' in line 2170). It can be made much more efficient if MINERVA's date splicing enhancements are used.

line 3050 redirects queries from the Microdrives to the file server's devices. This allows re-use of, presumably, currently redundant device drivers (MDV) rather than struggling with rather long N1_FLPn.

A recent change can be seen

then change your mind after you have started printing a file you have to type in the Toolkit II command PRT_ABT to stop it printing out.

The final lines (4020-4035), remind the user of the ALTKEY (or HOT_KEY) definitions one has set up. Lastly comes the little used 'DO' command. The rationale for its use will be explained in the next issue.

The code that is run in Multi-Basics can safely be compiled. Why then bother with Multi-Basics when compiled code runs quicker? Compiled code is more difficult to change if I add software. Moreover, since the routine that takes the longest is copying files, compiling it would produce little speed improvement.

Figure 5 - Backup

```
31190 :
31195 :
31200 DEFine PROCEDURE BACKUP( A$ )
31203 PRINT "RENAME TEST_BOOT TO BOOT ?": LET Y = GET_KEY
31205 IF NOT Y THEN COPY_O "FLP" & A$ & "_TEST_BOOT" TO "FLP" & A$ & "_BOOT"
31207 SAVE_O "FLP" & A$ & "_TEST_BOOT"
31210 END DEFine BACKUP
```

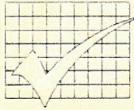
between lines 3100 and 3130. Originally I used a small program printed in QL World (RIP) to provide a keyboard beep. Last year I purchased Hermes. That required the first change to test if Hermes was present. The program provided with Hermes which tests for HERMES's existence is safe to use on non-Hermes equipped QL's. More recently, I purchased Jurgen Falkenburg's PC keyboard interface. Unfortunately, this is designed to be invisible to the QL. There is, therefore, no way to test for its presence except to ask the user...

The PRT_USE statement (line 4000) allows one to start printing and then carry on with other work on the QL without having to wait for the file to complete printing. The only downside with this is that if you

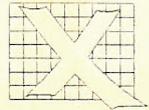
As an end-piece, here is a small routine to allow you to back up this program (figure 5). If one types 'BACKUP no', where 'no' is the drive you the file backed up to, e.g. BACKUP '1' will save the file on FLP1. Just in case you have made a horrendous mistake, the file is saved as 'TEST_BOOT'. You will be asked if you wish your previous version of 'TEST_BOOT' installed as the current BOOT program.

Plug: Visit your local Quanta group for help-ful advice and have most new programs demonstrated for you before having to shell out your money.

Andrew Pratt



SCREEN COMPRESSION



From the adverts of this program it is designed to allow the conversion of various graphic compression formats and to allow the screens to be saved in compressed form.

The package consists of four A5 printed instruction sheets and a 3.5" disk.

The program can be loaded by either LRUNing the boot program supplied or EXECing the program directly. Two versions of the program are supplied on disk one being a smaller but faster version that will work if you have the QLiberator extensions loaded. The program is fully configurable via the QJUMP Config program which is supplied. Once loaded you are presented with a screen showing all the options available.

The program is fully multitasking. I found this was useful as I frequently changed to SuperBasic to carry out a DIR on the disk that I was transferring information to and from.

The options available are large, but basically fall into three categories:-

- i) What information you are reading in.
- ii) What type of information you want to send out.
- iii) Whether you want to view or view and save the

information read in.

The latter option is relatively straight forward, that is not to say the others are complex just that there are many combinations.

Eleven compression formats are catered for :- Eye-Q, DJC, Painter, uncompressed, A.Quigley (QL World), R.Dwight (Quanta), CGA, PCX, Qplus, Moerel and

```

SCREEN COMPRESSION
1 - Source drive : FLPL
2 - Extension   : .SQP
3 - Screen mode : RBK      MODE 4      MODE 8      MODE 2/ST
4 - Compression : : : : : ALTO  DJC  PRINTER  QLU  QUANTA
                   EYE-Q  QPLUS GRAPHIC PCX  CDR  MOEREL

5 - Dest. drive : FLPL
6 - Extension   : .SQP
7 - Screen mode : : : : : MODE 4      MODE 8      MODE 2/ST CONVERT
8 - Compression : NONE  RS-LORED : : : : : PRINTER  QLU  QUANTA
                   EYE-Q  QPLUS  GRAPHIC PCX  CDR  MOEREL

9 - Invert colours : : : : : Yes
0 - START
W - VIEW
ESC- OUT

```

graphics toolkits. Some of these will be very familiar and some very obscure.

To use the program you move up and down the list of commands using the cursor keys and toggle between the possible options. The overall design of the program is one of ease of use.

When testing the program I initially came across some problems but after reading the instructions again these disappeared.

I found that the program was simple to use but wondered about its appeal to QLers.

The program does what it sets out to do, but to how many people actually want a program to convert between formats I am not sure.

The final twelve lines of the manual mention Dilwyn's routines SQUASH% and EXPAND%. More importantly, included on the disk are BASIC programs showing how to use these functions. These programs are clearly written and are explained very well. The programming style is clear and I found that I spent longer playing with these programs than with the actual Screen Compression program.

The fact that the program is easy to use makes it a good purchase, the additional basic programs supplied make it even better. So much better that as I will have to send my review copy back I bought the program from Dilwyn with a record two days delivery from when I sent the cheque.

The cost is just £10.00 which I feel is more than reasonable.

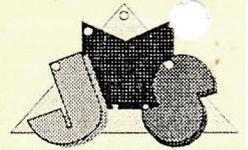
John Southern

Screen Compression costs £10 (UK), £11 (Overseas) and can be obtained from Dllwyn Jones Computing (0248) 354023, 41 Bro Emrys, Tal-Y-Bont, Bangor, Gwynedd, LL57 3YT.

Jochen Merz Software

Im stillen Winkel 12 • 47169 Duisburg • Germany

☎ and Fax: 0203-501274 • Mailbox: 0203-591706



proudly presents: SMSQ - the new QDOS compatible Operating System

SMSQ will work on any system which has a minimum of 1MB RAM and a faster processor than the QL: all ATARI ST/STE/TT (with or without QL emulator), GoldCard and SuperGoldCard and QXL SMSQ is not a modified QL-ROM-based system, it is completely re-written. This results in better performance and very many new features. SMSQ can be divided into three main parts: the Operating System, SBASIC and DV3. All three parts give advantages in all systems, and they will have a unique interface (again, after the way divided a bit between the Emulator with many features and other systems with less features).

The behavior of the operating system can be defined: if you want it to behave like QDOS (bad reaction under heavy load), then programs run faster. It can be set to be as fast as QDOS, but with a better reaction under load. For the ATARIs, 5 levels of memory protection can be defined. The operating system calls are, in general and on all systems, faster than QDOS. You can independently select the keyboard you're using, various translation tables (English, French, German, IBM and GEM), and the language for messages! The input line editing is extended to give you better editing control.

The SBASIC: You can now start as many interpreted BASICs as you want, all running independently. And you can add "private" extensions to these SBASICs, which go away when the job goes away. _BAS programs can now be EXecute, so you can't have to compile (filters) anymore. BASICs can be positioned everywhere on the screen (by call). SBASIC runs interpreted at about the same speed as compiled (QLiberated) BASIC, i.e. MUCH faster than SuperBASIC used to run. And, of course, a large number of improvements have been made, to give you an idea: you can directly enter hexadecimal and binary numbers / fast integer FOR and SELECT / improved error messages / BPUT, BGET now allow string to be parameters / END FOR, END REPEAT, REPEAT, NEXT etc. don't need a name anymore / LOAD, MERGE also accepts QSAVED files / SAVE does not need a filename, it saves the file under the same name it has been loaded / extended PEEK and POKE, PEEK\$, POKE\$, LBYTES, SBYTES etc. accept channel numbers instead of filename / improved slicing / machine-detection functions exist / ... and more! Default channel commands (ED defaults to #2, DIR to #1, BPUT to #3) now use channel #0 instead if they find that their default channel is not open - and if #0 is not open, it is opened implicitly!

Device Driver Level 3 - gives you flexible formats. You can access a DOS or TOS-Disk exactly like you access a QDOS disk, the system detects the format automatically. If you insert a DOS disk and type DIR flp1_ then you get a directory of it. You can use all the normal file commands in every program (not special extensions like ATR or ATARI-DOS) to access files to copy, delete, load etc. This allows you to boot from one disk, even if it is a TOS/DOS disk!

The WIN features are greatly enhanced on the ATARI version: you can now access ANY QDOS or TOS partition on any drive, even Extended Gem and Big Gem. You can now have more than one QDOS partition on one drive. You defined WIN2_ to be TOS-Partition C or D or ..., win3_ to be partition 7 etc., all this is VERY useful to transfer documents, graphics etc. between various systems. And, you can software-write-protect one or more harddisk partitions. Full harddisk control via WIN_Control Thing!

In addition, HD (1.4MB) disks are now supported on the ATARIs too! Full floppy disk control via FLP_Control Thing!

Finally, the various bits which don't belong into any of the three categories above: The serial and parallel port drivers of all systems have been put to the high standards of the ATARI drivers: more parameters, you can define in the device name whether this channel uses TRA or not (instead of toggling the global TRA all the time), dynamic buffers have been added. Full control of the port parameters with the SER_PAR_PRT Thing!

Other virtual drivers have been added or improved: named pipes (first in, first out, of course), History (last in, first out, new!! Even shared Histories are allowed) and DEV, the virtual directory device which allows search chains to be built.

The standard ATARI version supports the QL-Emulators/QVME, but it is possible to get a version which supports the monochrome 640x400 display too, which run on ATARIs without emulators (and the Stacy!!!) in addition. The resolution can now be changed at run time on the Extended Mode emulator (you could always do it on the QVME).

It has taken a bit longer than estimated, but for more facilities than originally planned were implemented. Latest news: you can execute SBASIC programs even out of the QPAC2 Files menu. You also get an SBASIC/QD F10 Thing interface inbuilt for free, so that you can use QD for writing, editing and starting (uncompiled) SBASIC programs!

As you can see, you get a system you were always dreaming off. We decided to offer it in steps, with the main part (as described above) being ready now! This means you can now benefit from what exist now - at the same price in steps that the final version would cost in total. Later on, a new Screen driver will come which will allow that printing will always continue, even to a window which is completely or partially overlapped by other windows. Visible areas of overlapped windows will be updated to the screen!!! Then we will have (everyone's wanting this for years) "background" floppy/harddisk operation, i.e. when files are read or written, the whole machine does not stop anymore completely. The third improvement stage will be a pointer-integrated SBASIC environment with Single-step/trace facilities and more.

	Feature	ATARI ST(E)/TT	(Super)GoldCard	QXL
This exists NOW!	New Operating System	NEW DM 199,-	NEW	already ex.
	Multiple, fast BASICs	NEW	NEW	already ex.
	Flexible Level 3 Drivers	NEW Level C, D or E	NEW DM 199,-	already ex.
	HD Disk-drive support (STE/TT)	NEW (if you own the QJUMP-drivers already) else DM 249,-	already ex.	already ex.
	TT Fast RAM support	NEW	impossible	impossible
	Monochrome Screen-driver	NEW + DM 50,-	impossible	impossible
And this will come later..	New Screen-driver	NEW + DM 50,-	NEW + DM 50,-	NEW DM 249,-
	"background" Disk/Harddisk	NEW + DM 50,-	impossible	NEW + DM 50,-
	BASIC-Development-Environment	NEW + DM 50,-	NEW + DM 50,-	NEW + DM 50,-
	Total price (when all is available)	with rebate, without monochrome DM 349,-	DM 299,-	DM 349,-

As a special bonus (we know that many users own more than one system), we offer a full package which contains versions for all three systems for only 33% more. This also applies to the upgrades, e.g. DM 66,66 instead of DM 50,-, which, we think is very fair (it just covers extra disks and manuals),

More news from Jochen Merz Software are coming: WINED, a pointer-driven editor for all sorts of media (Harddisk, Floppy disk) - and with its many features it offers far more than a plain sector editor! MENUCONFIG Level 2 - are you bored of having to re-configure all the new files you get every time you update? Select every item you want to change? CONFIG Level2 does it automatically for you. It remembers your last configuration and updates more recent version automatically! More news in the latest Jochen Merz Software Catalogue which you can get for free!

TERMS OF PAYMENT

Postage and package (Europe) DM 14,- (if total value of goods is up to DM 50,- then only DM 9,-). (Overseas) between DM 14,- (1 item) and DM 35,- (maximum). All prices incl. 15% V.A.T. (can be deducted for orders from non-EEC-countries). E&OE. Cheques in DM, £'s, Eurocheques and Credit Cards accepted.

