

QL Today

Volume 5
Issue 1
May/June
2000

ISSN 1432-5454

The Magazine about QL, QDOS,
Sinclair Computers, SMSQ...

help Preview Font quit

Arial

text for preview :
AaBbCcDdEeFf

font preview :

AaBbCcDdEeFf

Athletes
Atlas Overwate Normal
Aurora Bold

fontutils quit

pfb2pff

FontShow

FontPreview

do help ProWesS File Search quit Zzz

Search String QL

Extensions_txt Not

Directory WIN1_data <

Tree	First Occurrence	Binary
Spaces	Word	Case Dependant
QLUSERS_TXT : line 1 :	QL Email Users Database Update - 24/12/97	
QLUSERS_TXT : line 65 :	Daglish, Neil	<qltwirl@mata.vuw.ac.nz> Ne
QLUSERS_TXT : line 298 :	Terkelsen, Jrgen	<qltjt@dk-online.dk> Denn
IMSlst.txt : line 24 :	39 SMSQ A Colour	IMS SMSQ Updates (ATARI QL Col

wake do help PFlst quit Zzz

Footer \$N \$D printed with PFlst

tab distance 4

Landscape

fontsize 10

Two Columns

Font Courier

device default driver

driver

extensions

directory WIN1_data

EV/QBR.txt	delf_bas	IMSlst.txt	new_data_base_dba	RomDisq_ldp
Cardhst.txt	Eind90.1.98.txt	laptops_ldp	Opening_lib	SCART_ldp
cfg_bin	Eindhoven_prices_tab	moplate_ldp	postcodes.txt	style_aba
ChessClock.exe	Envelope_ldp	mf_scores	printer.dat	style.dbf
cologo_ldp	Envelope_ldp	monitors_T91	Q40Hove.txt	style_def
colrep_ldp	gnugo.dat			
default lay	IO_PTR			

wake do help Pwfile quit Zzz

action : Fileinfo link include subtree

sort method : alphabetic Commands

extensions : Not

directory : WIN1_pws_ <- Tree All

destination : RAM1 <- Exchange

file :

all : free on source 41MB on dest 1352kB

> doc	> mine	> pw
> ext	> pf	ument1_parg
> fnt	> prg	

help ProW

Back Load Reload Er

Title : General ProWesS documentation

ProWesS documentat

PROCS, Professional & Graphical Software
Dr. Frans Hermijckxlaan 13 /1
2650 Edogem
BELGIUM
tel : +32 (0)31 457 84 88 fax : +32 (0)31 458 62 07 e-mail : joachim@club.innet.be
www : http://www.club.innet.be/~year2827

- Introduction
 - What is ProWesS
 - ProWesS manual
 - Disclaimer & Copyright
 - Present, Past and Future
 - Installation

With
ProWesS
Cover
Disk!

Contents

- 3 Editorial
- 4 News
- 7 Quanta AGM, Manchester
Dilwyn Jones
- 11 TURBO & Pointer Environment
George Gwilt
- 12 Programming ProWesS in SBASIC - and why not?
Wolfgang Lenerz
- 15 Gee Graphics! (on the QL?) - Part16
Herb Schaafl
- 20 Some Notes on EPSON ESC/P2 Printer Codes
Don Atkins
- 21 My Black Box
Alex Wells
- 23 A busy Knight
Mark Knight
- 27 Q40 and Perfection
George Gwilt
- 28 QIMI Hints
Dave Westbury
- 29 You and Your Software - Just good Friends? Part 8 - What the User doesn't see.
Geoff Wicks

SERNET SPECIAL

- 31 Q40 - Q40 Sernetting
Derek Stewart
- 32 SERNET
Dilwyn Jones
- 33 A week-end with SERNET
Davide Santachiara
- 35 We always said: "it does work!"
Jochen Merz

- 36 Success - A Review
Tim Swenson
- 38 CAUTION: HOTKEYS - Don't Burn Your Fingers, Part 4
David Denham
- 41 Letter Box
- 41 Cover Disk & Postcard
- 42 Programming in Assembler - Part 7
Norman Dunbar
- 51 Last Minute News from JMS
- 52 Bytes Of Wood

Advertisers

in alphabetical order

Jochen Merz Software	9
Just Words!	29
QBOX USA	49
QBranch	24, 25
Quanta	39
RWAP Software	43
The Library	27
TF Services	47

QL Today

ISSN 1432-5454

German office & Publisher:

Jochen Merz Software
Im stillen Winkel 12
47169 Duisburg
Germany
Tel. +49 203 502011
Fax +49 203 502012
Box1 +49 203 502013
Box2 +49 203 502014
Mobile +49 1701 222277
email: JMerz@j-m-s.com
email: QLToday@j-m-s.com

English office:

Q Branch
20 Locks Hill
Portslade
BN41 2LB
United Kingdom
Tel. +44 1273 386030
Mobile +44 7836 745501
Fax +44 1273 381577
email: qbranch@qbranch.demon.co.uk
email: QLToday@j-m-s.com

Editor:

Dilwyn Jones
41 Bro Emrys
Tal-Y-Bont, Bangor
Gwynedd
United Kingdom LL57 3YT
Tel. +44 1248 354023
Fax +44 1248 354023
email: dilwyn.jones@dj.softnet.co.uk
email: QLToday@J-M-S.com

Co-Editor:

Bruce Nicholls
57 Shaftesbury Rd
Romford
Essex RM1 2QJ
United Kingdom
Tel +44 1708 755759
Fax +44 870 0568755
email: qltoday@q-v-d.demon.co.uk
email: QLToday@j-m-s.com

QL Today is published bi-monthly, our volume begins on beginning of June. Subscriptions begin with the current issue at the time of sign up. Please contact the German or English office for current subscription rates.

We welcome your comments, suggestions and articles. YOU make **QL Today** possible. We are constantly changing and adjusting to meet your needs and requirements. Articles for publication should be on a 3.5" disk (DD or HD) or sent via Email or into one of the JMS-BBS's. We prefer ASCII, Quill or text87 format. Pictures may be in _SCR format, we can also handle GIF or TIF or JPG. To enhance your article you may wish to include Saved Screen dumps. PLEASE send a hardcopy of all screens to be included. Don't forget to specify where in the text you would like the screen placed.

Article and Advertising deadlines are as follows:

Issue 1: 30 April	Issue 2: 30 June
Issue 3: 30 August	Issue 4: 30 October
Issue 5: 30 December	Issue 6: 28 February

QL Today reserves the right to publish or not publish any material submitted. Under no circumstances will **QL Today** be held liable for any direct, indirect or consequential damage or loss arising out of the use and/or inability to use any of the material published in **QL Today**. The opinions expressed herein are those of the authors and are not necessarily those of the publisher.

This magazine and all material within is © copyright 2000 Jochen Merz Software unless otherwise stated. Written permission is required from the publisher before the reproduction and distribution of any/all material published herein. All copyrights and trademarks are hereby acknowledged.

Welcome to the fifth volume of QL Today magazine. With this issue I am pleased to say that we have secured the services of a former QL magazine editor to assist us with the preparation of this magazine. With my move to a new day-job I have been struggling to find the time to do justice to QL Today, and having the help from someone like Bruce Nicholls should mean we can work to bring you a better QL magazine than ever! Bruce has been a QL software author (e.g. the Screen Dazzler program), a software publisher (Quo Vadis Design) and a magazine editor and publisher (QReview magazine). Welcome on board Bruce.

The 'Colour Drivers' (or GD2 to give the proper name) have taken a further step forward with the imminent appearance of a QL version. Marcel Kilgus has been using an early trial version as I write this, which presumably means he's familiarising with GD2 to eventually add the 'colour driver' facilities to QPC. I am looking forward to that - I am one of the keenest of users of QPC2! I also look forward to the Aurora versions, as I have a MinisQL Aurora eager to avail itself of colour drivers!

Production of Q40s hit a slight snag earlier this year after only about 28 machines had been issued, due to an unknown suspected hardware problem on some boards. Let us hope this is soon resolved - the Q40 may be a low volume production machine, but is important to the QL scene as the only true new native QL hardware of late. Jonathan Dent continues to work on the TCP/IP stack/SoQL software for QDOS. This too is very important to the future of the QL, so best of luck with its completion Jon.

The QL's presence on the internet is firmly established. Most of the well known names on the QL scene now have web sites and email addresses and Quanta took a step towards boosting its internet presence recently by creating a new Web site (www.quanta.uni.cc) and standard email

addresses for contacting its officials, all of whom were re-elected to their posts at the April AGM. Keep up the good work, guys!

I went to the Quanta AGM in Manchester, England, in April, and although it was not the best attended Quanta event ever, it was still an enjoyable day. It was nice to see so many eager QLers still beavering away on such a variety of systems, although it's perhaps sad that there are not many of the original Black Box QLs at these events any more. More positively, it's a sign we're moving on perhaps with plenty of Auroras, some Q40s and many other computer systems running QL emulators in evidence. See my show report elsewhere in this issue for news of some interesting projects by Simon Goodwin for example.

QLers in Europe and North America have their own shows from time to time of course and long may that continue - the regular shows in Eindhoven (The Netherlands) and in the USA are well established.

The next big event in the show calendar in Britain will be the big international event at the Horizon Centre in Portsmouth in October - see Roy Brereton's information article elsewhere in this issue. This is a large, modern centre to house a 2 day QL event. All of the QL traders hope to be present and plans are in hand to help QLers worldwide find accommodation there for the weekend. The south coast of England is ideally situated for those travelling by ferry from Europe, and not too far for those flying in to the major British airports. This is a major event to celebrate the QL in the new millennium and hopefully QLers worldwide will fly in to be together for the first big event of its kind in the history of the QL.

By then, I'll be married and I'll hope to meet as many of you as possible there!

NEWS

Q-Celt Computing

New Clipart CD-ROM for the QL Emulators

We launched our latest QL CD-ROM at the QUANTA AGM show in Manchester at the end of April. It is a collection of LineDesign Clipart, contained in a 600Mb QXL.WIN for use on all the emulators that support it, such as QPC or Q-Emulator. The file contains well over 5,000 individual pieces of Clipart on a wide variety of themes, most of which have never been seen before on a QL as they have been newly converted by Dilwyn Jones and myself, from Adobe Files in the PC Domain, and also some favorites. The Clipart is neatly categorised in directories, for ease of use. If you use LineDesign a lot, you will find this collection very useful - and even if you don't, why not buy LineDesign? It is much cheaper now as ProWesS (needed to make the new versions of LineDesign work) is now freely distributable. There is even a Demo copy of an older version of LineDesign (v. 2.06) on the CD to use as a viewer for the Clipart files. The CD-ROM costs just £15 Sterling including full Postage and Packing, and is available now.

QUANTA Library CD-ROM Project

Q-Celt have also been approached by QUANTA to convert their entire QUANTA Library of programs onto a CD-ROM. I was approached by Roy Brereton at the Manchester AGM and have agreed to start work on it as soon as Roy can send me the full library on HD disks. The library is approximately just over 100Mb, which will be available both as a QXL.WIN for use on the emulators, and each of the QUANTA disks will also be available as a ZIP file, outside of the QXL.WIN for use by people who don't have a QL Emulator, but may have access to a PC with a CD-ROM at work for example, and they can therefore copy the ZIP file they want to a floppy and unzip it at home on their QL. It is hoped to have this available by the QL 2000 show at the very latest, hopefully a lot sooner.

Work continues on our Religious CD Collection, containing text files of various Bibles and other religious lexicons, from the various religions of the world, some interesting reading! Should be available by the time you read this.

We also hope to be releasing the QL Emulator CD soon too, an Alpha test version was on dis-

play at the AGM in Manchester, having been burned to CD-R just the night before. We are still testing it, and are looking for people to test the various emulators to see if they will run directly from the CD, particularly the uQLx, AmigaQDOS and Q-Emulator for Macintosh versions. Contact us if you're interested on Q_CELT@hotmail.com

Quanta's new Web Site

Quanta's chairman, Robin Barker, has set up a new Web site for the group. Point your browsers at <http://www.quanta.uni.cc> to view the pages. In addition, new email addresses have been created for the officials, to provide a standardised means of contact with the officers by means of easy to remember email addresses which should not change should new officers be elected in the future.

Chairman	quanta_chairman@uk2.net
Secretary	quanta_secretary@uk2.net
Treasurer	quanta_treasurer@uk2.net
Editor	quanta_editor@uk2.net
Membership	quanta_membership@uk2.net
Software editor	quanta_software@uk2.net
Head librarian	quanta_librarian@uk2.net
Other enquiries	quanta_support@uk2.net

The site includes a form to tell the site about your QL-related Web site and thus enables the site to include a link to your site.

You can also use the site to place classified adverts/notices/announcements, which may be placed (and deleted!) via a form from the Web page. You get an email back to confirm the submission and a password for when you later wish to delete the advert!

The QL Email Users Database is now updated to provide better service. To help prevent the possibilities of us being spammed (i.e. sent junk emails), the actual database is currently only available online from Quanta Support - in time, password protected access will be introduced for Quanta members.

New Sprite Editor for the Q40

A beta version of my sprite editor for Q40 is available from

<http://www.altern.org/grimberty/ql/index.html>

(or directly by email on request, expect a zip of 36 kbytes and some delay).

Even if it is not yet finished, I would like to have some feedback (and constructive criticism/review). There is absolutely no documentation on how to use it, so feel free to take note of your troubles and intuition for later documentation (you can contribute to it too !).

In short, it runs only on Q40 with SMSQ/E (I did not test Classic), and only in 1024x512. It outputs static sprites in assembly for Qmac. It can input sprites from binary files, as long as they are not dynamically patched.

It can edit mode 4, mode 8 and mode 33 sprites, each up to 68x60. It manages only all-or-nothing masking for each pixel.

Grimbert Jérôme (jgrimbert@atos-group.com)

TCP/IP (SOQL) News

Jon Dent reports that at the time of writing (beginning of May) he had nearly got the TCP part of his internet software for QDOS working. His first test application solicits the expected reply when trying to connect to a host. Jon then built a crude POP3 application to see if he could actually exchange data and thus test out the TCP in all possible states. After getting that working, progress came to a temporary halt due to problems with his Internet Service Provider, and also coming under work pressures because of computer virus problems. Jon says he hopes to have better news of progress to report by the next issue and just possibly a first Beta release.

Graphics format support

Claus Graf wrote: I have finished pqiv 0.14 and the new feature is TIFF support. Please, check it out. Download is possible from www.q40.de.

RWAP Software News

Following the review of Starplod in Volume 4 Issue 5, I can supply a copy of this program which will work on Q40 and Aurora. I supply Starplod on a disk with 6 other PD adventures and a PD game for just £2.00

Q-Help is now v1.05 which incorporates various enhancements including a FIND command in both screen modes and the search is no longer case sensitive.

Nemesis Mk II is now v2.03 and is easier to install on hard disk systems.

The address given in the demo version of D-Day Mk II supplied on the last cover disk is now out of date - see my adverts for the current address. The full version of D-Day Mk II now costs only £10.00

Daniel Baum Website

Believed to be the only QL related Web site in Israel, Daniel Baum's Web site has recently undergone a facelift. The site features pages about the QL, its history and current developments, pages about Daniel's QL software and about Frederic van der Plancke's QXLWIN Explorer software. www.angelfire.com/il/dbaum/

George Gwilt Software

Now available from www.itimpi.freemove.co.uk/ are George Gwilt's extensions for QDOS to support hardware Floating Point. These extensions add to all variants of QDOS (including Minerva, SMS2, SMSQ and SMSQ/E) save/restore of the FPU context when task switching occurs, and also a generic QDOS port that caters for all the different Motorola FP Support packages. FPU requires a 68020 or better processor. The same web site also includes a new version of the GWASS assembler, for all of the Motorola processor family up to 68060.

News from Dave Westbury

New JPEG viewer for QDOS/SMSQ in any mode (4/8/Aurora/Q40) can be found at

www.soft.net.uk/dj/software/other.html

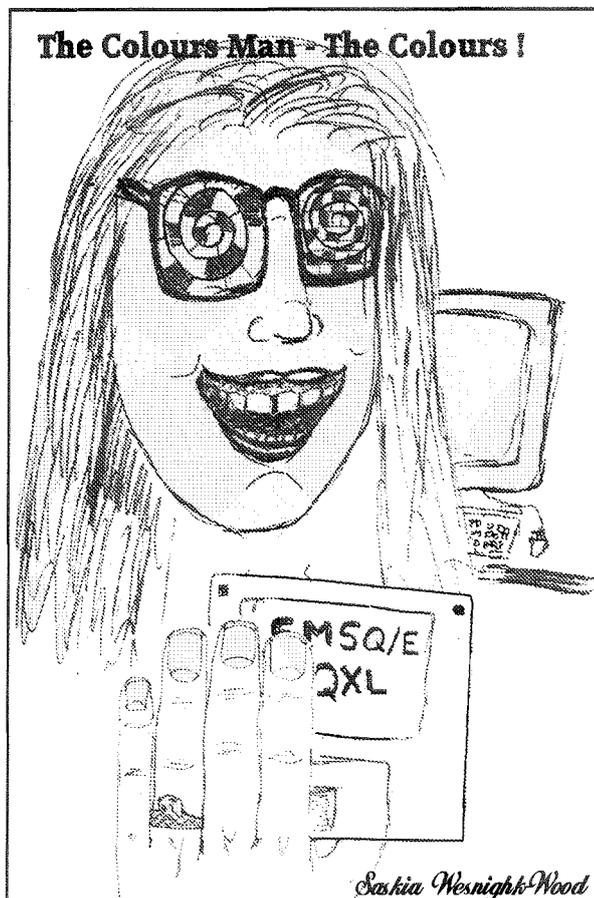
It is called **Photon.zip 11.3KB**

For those who don't like readme's just use:

```
EX Photon; 'filename'
```

in desired mode (4/8/33). Can also display in Aurora 16/256 colours (add \a0 or \a1 to end of filename), or run in new driver mode 16 (256 colours).

Wallpaper/dithering/screen aspect ratio scale/monochrome supported.



Cartoon

Just Words

QL-2-PC TRANSFER VERSION 3.00

Version 3 of QL-2-PC Transfer has now been released. The main new feature is the conversion of QL word processing files to HTML, the language used on the internet. HTML files can also be converted to ASCII. In addition subscript and superscript are now supported, as well as page breaks in Quill. It is also possible to define a highlighting style to generate italics from Quill files. Finally a bug has been corrected that was preventing the transfer of Quill files larger than about 10,000 words.

A demonstration version of the program is available from Just Words! or can be downloaded from our web page.

Upgrades from version 2 are free of charge. Most users will already have received their free upgrade. If you have not received it, please send your master disk to Just Words!

email: geoffwicks@hotmail.com, website:

<http://members.tripod.co.uk/geoffwicks/justwords.htm>

Jochen Merz Software

Hottest news is the release of the QXL high colour drivers with SMSQ/E V2.98. Many Q(X)lers are waiting for several years for these drivers. They are now available as an upgrade to SMSQ/E for the QXL. You have the choice of QL compatible modes and high-colour mode (65536 colours), and you can switch between the modes during run-time.

Other add-ons (you may have heard about them for the Q40 colour drivers) are the ability to use background colours and background images.

In general, the software interface between the QXL and the PC in which it is plugged has been vastly improved - you should get faster serial port rates and also faster parallel port output.

Check the JMS advert for details and prices.

SMSQ/E V2.98 is also available for other systems (free of charge) but they will not add major additional features (no high-colour yet for GoldCard/SuperGoldCard systems) nor for ATARIs. However, you can use the new true colour specifications (which is then translated into 4 colours) and you can use the background colour and background image feature.

Your next question may be: and what about hi-colour drivers for QPC? Of course, Marcel is already working on them. When he first got the sources, he thought it would be ready before the next Eindhoven meeting in August. Now it seems that it will be ready much earlier!

Q Branch

Q Branch is moving! Q Branch have moved their HQ to

Q Branch

20 Locks Hill

Portslade

BN41 2LB

United Kingdom

Tel. +44 1273 386030

Mobile +44 7836 745501

Fax +44 1273 381577

We have separated the business from the shop because Roy Wood will now be working at a full time day job. The Bank Volt will still exist as a PC outlet but all of the Q Branch functions will be moved back to Portslade including the distribution of QL Today. It is best to call early evening (6pm - 8pm UK Time) if you want to speak to us. Dilwyns Fontpack is finally released. 8 disks of PD fonts for ProWesS and a printout of the full fontsets together with an installation manual.

We will be handling the UK distribution of the QXL SMSQ/E colour drivers. We do have a few second user QXLs in stock. Call for details.

News from the QL Users list

We have picked some interesting news for you:

Andrea Carpi wrote, that Ludovico Camarda (a QL User who lives in Germany) has registered the following domain names: **Sinclairql.com**, **Sinclairql.net** and **Sinclairql.org**.

Ludovicos e-mail address is lcamarda@esoc.esa.de

He is looking for help to administer these sites.

The question "why bother with the "sinclair" part?" came up, suggesting that something starting with "Q" would be better (we do have a Q40 site, by the way). "theQLsite.com" was another suggestion.

Darren Branagh replied: "The main reason for a longer domain is that they are cheaper - shorter domain names tend to have higher price tags, as they are snappier, and lots of companies use TLA's (three letter acronyms). However, I have reserved www.thesinclairql.com and have been successful in getting it, but I like theQLsite.com, so may change!!"

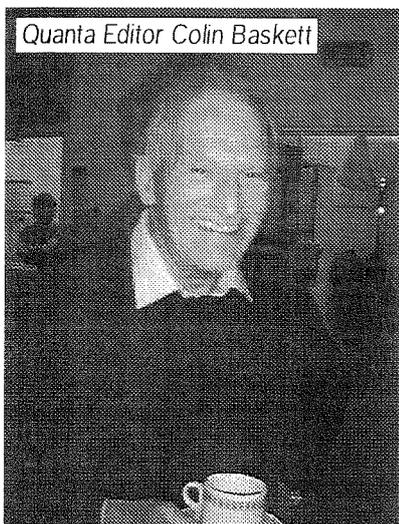
If we carry on like this we may have more QL related domain names than users - Editor! This was just an extremely short summary of the very many interesting news you can get by subscribing to the QL Users Email list - why not subscribe now? Send an email with the message **subscribe ql-users** to majordome@nvg.ntnu.no

Quanta AGM, Manchester

Dilwyn Jones

A cold but sunny day welcomed Quanta members to Davyhulme in Manchester for the Annual General Meeting.

The scout group HQ provided the venue where users and traders converged. The usual fight for table space took place at the beginning between Jochen Merz, QBranch, Q-Celt, Quanta and Geoff Wicks. Notable by their absence due to illness were Rich Mellor of RWAP Services and Bill Richardson. Tony Firshman also could not make it.



Quanta Editor Colin Basket

Down one end of the hall there was an extensive display of Sinclair computers from ZX80s, ZX81s, Spectra, QL, Thor, Jupiter Ace... you name it, it was probably there if it had even the most tenuous of Sinclair connections.

Quanta announced a new Web site set up by chairman Robin Barker. Those who are online can access the Quanta page at <http://www.quanta.uni.cc/>

Jochen Merz released a new Wolfgang Lernerz game called The Wall, which is a puzzle where you are faced with a wall made of coloured tiles, which you have to destroy by

clicking on a tile with at least one like-coloured tiles next to it, but not diagonally. Jochen was assisted on his stand by Bernd Reinhardt.

Geoff Wicks was proudly showing off version 3 of his QL2PC Transfer software, now with some HTML functionality.

QBranch's stand was dominated by Q40s, with interest high now that the long awaited colour drivers are available. Having successfully supplied 28 Q40s by then, production hit a hiccup when it was realised that for some reason some failed to run certain software despite having the same operating system version, while others worked OK, so production was temporarily held up while the matter was investigated. Several copies of the Linux-68k CDR were to be seen around the room confirming the view that the Q40 is of interest to many as a comfortable

Linux platform - many QLers now have an interest in two or more operating systems, with Linux becoming

increasingly popular among us. Linux-68k is able to access the high colour modes, said Roy Wood, who also demonstrated Claus Graf's PQIV picture viewer. Roy also said that Mark Knight and George Gwilt will eventually release a version of Turbo compiler and Pointer Toolkit for SMSQ/E. The Poin-



Show organisers Sarah and John Gilpin

ter Toolkit will add the facility to compile pointer driven programs - a feature missing from earlier releases of Turbo.

Q-Celt brought an interesting bundle of goodies over from Ireland, including novelties as black IBM keyboards for QLers to make their systems more QL-like. Quote of the day: "it's good, but scratch off those three letters and it'll be much better." No comment. Darren Branagh's stand was somewhat tainted by all the MS and Intel bits, but made up for it by



Bernd Reinhardt in conversation with Quanta chairman Robin Barker

selling a couple of CDs for QL-compatibles, including the new CD of over 5,000 items of Line Design clipart. Darren plans a range of these CDs and an alpha test copy of the QL Emulators CD was on display, although not yet ready for release. A cunning piece of Irish logic led to a special offer card on his wall phrased as: "Mouse Mats - Last One - price each=...."

Darren's journey back to Ireland sadly got badly marred by drunken English yobs who decided to play an impromptu



Bernd Reinhardt and Jochen Merz trying to sell some French Flags

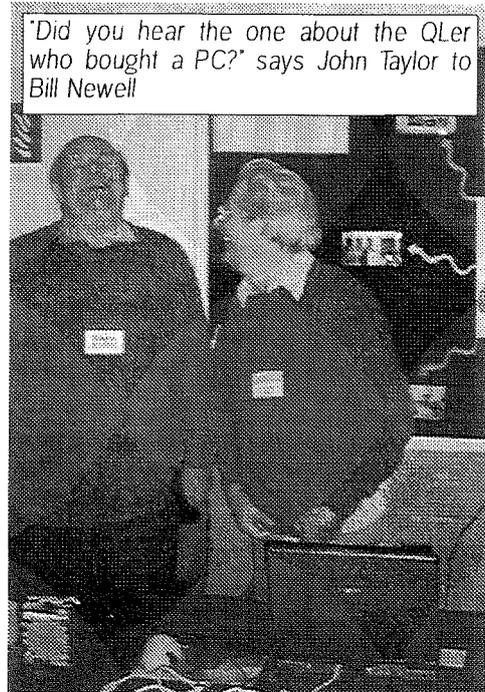
with a Qubide and Super Gold Card. An external drive and power supply stack complemented the system, making it much smaller than an equivalent PC tower case system, and making it recognisably a QL of course. I have tried to persuade the

owner of the system, Alex Wells to write an article for us about it.

During the AGM at the end of the afternoon, members voted to keep the existing committee members as nobody had elected to stand against them for election. Quanta's trading accounts for the previous year showed a small trading loss of '572 for the year, although the bank account had not gone into the red at all, and some saving had been made through lower auditors' fees. The Chairman (Robin Barker) said that the issue for the dual platform proposals could not be discussed at this AGM and time would be set aside at the QL2000 function in October for this. The Horizon Centre in Portsmouth had been decided upon as a

Classic for both Amigas and Q40s - Simon demonstrated it in SBASIC on a Q40 with commands as simple as COPY flp1_sound_file TO sound. It also provided a means of resampling sounds to provide special effects facilities. Simon also told me he was working on QDOS software to allow him to download pictures from his digital camera - the answer to Bill Waugh's article in the last issue! Although to be fair, Simon did say that even if he did finish the software, it may be specific to the kind of camera he has. Simon was also interested to meet Dominic Morris, author of aTCP/IP stack in just 7KB for the Z88 who was present at this meeting.

Later

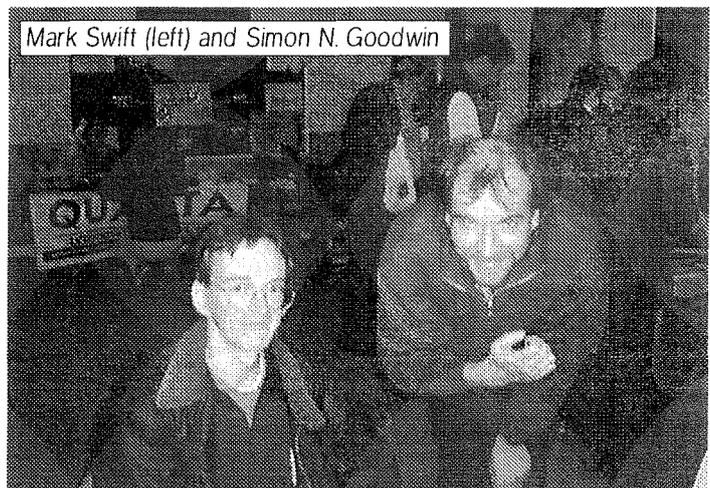


"Did you hear the one about the QLer who bought a PC?" says John Taylor to Bill Newell

game of rugby on the ferry and injured an elderly lady, and Darren got caught up in an ensuing scuffle.

Mark Swift, author of QDOS Classic for the Q40, travelled from Blackpool and met up with Simon Goodwin for the first time - they had long corresponded and spoken by phone and once introduced in person, they set about implementing a sound device driver for QDOS

in the day. I saw an interesting construction job where an Aurora had been built into an old QL keyboard case along



Mark Swift (left) and Simon N. Goodwin

JOSILEN MERZ SOFTWARE

Im stillen Winkel 12 D-47169 Duisburg
Tel. 0203 502011 Fax 0203 502012
<http://www.j-m-s.com/smsq/index.htm>

SMSQ/E Version 2.98 Update

Yes, it exists! The colour drivers not only work on the Q40, but also on the QXL card. 65536 colours are possible. The QXL interface has been improved too, the keyboard driver has been rationalised, and the speed of the serial ports and the parallel port increased. Depending on your PC, up to 115kBaud should be possible.

This Version is available for the other systems too. They cannot handle more colour, but "understand" the new colour commands and convert them to 4 or 8 colours. You can also define "desktop" background colour and image, even on the ATARI- and GoldCard-Versions.

Prices for Update/Upgrade

for ATARI ST, STE, TT	free
... with additional pages for the manual	DM 16,-
for GoldCard/SuperGoldCard	free
... with additional pages for the manual	DM 16,-
for QXL with add. pages for manual	DM 79,90

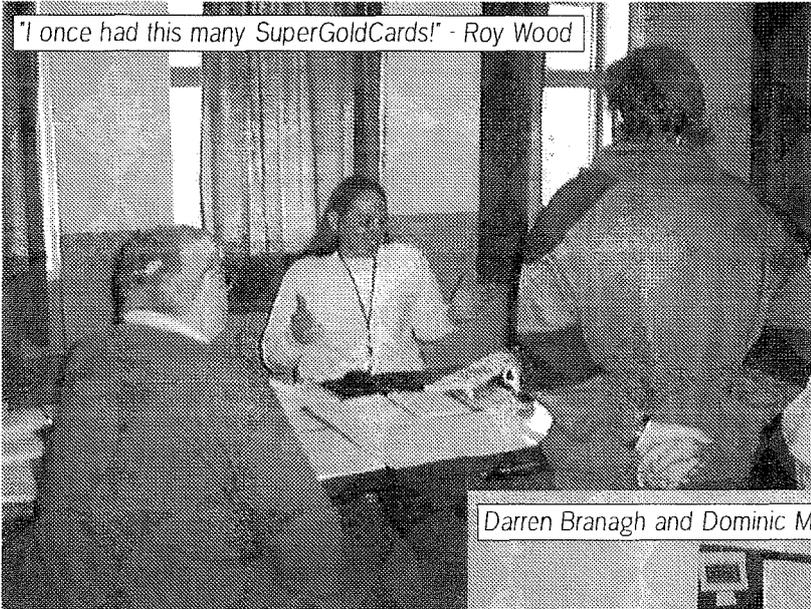
The new colour drivers for QPC2 will be coming soon!

TERMS OF PAYMENT

Postage and package [Germany] DM 8,99 (if total value of goods is up to DM 50,- then only DM 5,99). [Europe] DM 14,50 (if total value of goods is up to DM 50,- then only DM 9,50). [Overseas] between DM 14,50 (1 item) and DM 35,- (maximum). All prices incl. 16% V.A.T. (can be deducted for orders from non-EU-countries). E&OE. Cheques in DM, EURO, Eurocheques and Credit Cards accepted.



"I once had this many SuperGoldCards!" - Roy Wood



an international meeting. Arrangements were being sought with a quality local hotel to provide a significant saving on accommodation costs if a good number of members chose to stay there for the weekend. Discussion turned to the TCP/IP stack software from Jonathan Dent, with the committee suggesting that financial assistance could be offered to Jonathan Dent to produce the software and it would be made

Darren Branagh and Dominic Morris in conversation



venue for QL2000 and although somewhat expensive to hire for a 2 day event because of overnight security costs etc the committee had decided to authorise the event since it was a one off special event for the QL in the year 2000, specifically nominated as

Simon Goodwin gets his hands on a Q40 to test the sound device driver.



available via Quanta software library. Some concern was expressed that this might then preclude non-members from use of this software on their QLs. Simon Goodwin asked if rather than direct finance, if any material assistance could be provided, such as a range of 'QL' hardware for the software to be tested upon.

As you can see, QL shows are usually quite interesting. You can see things you've never seen before (or for a long time), you can meet people you never met before, and, generally speaking, help turning any QL show into a success. So come along to the next QL show near you - and if you do you can help making sure there will be another QL show near you!

TURBO & Pointer Environment

George Gwilt

When Mark Knight said that it would perhaps be useful to have a means of producing Pointer Environment programs written in SuperBASIC which could be compiled by Turbo, I agreed that that would be nice, but thought that my system of approaching PE via C68 was probably enough to be going on with...

When Mark went on to say that he himself had not as yet been able to produce such a SuperBASIC program, I immediately rose to the bait. Months later, building on my experience with my C68 system, I managed to set up what I call TurboPTR.

There are already various systems for tackling the ticklish PE problem. There is first and foremost QPTR. This consists of the software required for PE plus documentation plus a large library of macros for those using assembler language and the Qmac assembler plus a set of extensions to SuperBASIC to allow PE programs to be written that way. There is also a set of header files and routines written by Tony Tebby enabling an intrepid programmer to force an entrance to PE via C68. Jonathan Hudson's QEYES is just such a C68 product. Finally there is Easyptr which is no doubt easier than QPTR's BASIC.

Why then do we need more systems?

1. Assembler programs using Qmac's macros (or even mine - for use with my own assembler, GWASS) are incredibly difficult to write and debug. Just place an item too far to the left in a window and the whole program collapses with no hint as to what has happened. And the placing of items with the sizing of windows and sub-windows has to be just right. So let's try C68 or SuperBASIC instead!

Assembler is difficult

2. As far as I am aware all the non-assembler systems so far available (apart from my C68 one), are based on the production of what's called a "working definition", and not the "window definition" itself. I must explain that the original concept of the PE system was that each "window" which can be moved or resized is defined in a way that enables many different layouts to be produced from it depending on the size required. Such a definition starts out with the maximum size and ends up with a set of repeated sections each defining a different layout and each with a smaller size. The software provided in PE picks the appropriate repeated section for any given size and will set up the "working definition" required by an operating program. The simplest form of "window definition" allowing a resizable window contains only one repeated section. This gives the minimum size of window, but with a "scaling flag" added. In this case, given a required size the PE software will set up an appropriately sized window. This is definitely easier than doing it yourself - as you have to do if you are going to set up the "working definition" directly.

Resizing is tiresome

3. A SuperBASIC program, unless you have SMSQE or some such system, has to run on its own. You can't for example have two SuperBASIC pro-

grams running concurrently on a QL with a JS rom. It is thus useful to be able to compile such a program. Since Turbo on the whole compiles quicker programs than QLiberator, it would be nice to have this possibility.

No Turbo compilation Why do existing systems not use my approach?

1. Tony Tebby, in introducing his C68 system, says that C68 does not cope easily, or at all, with relative pointers. In a C68 running program it is true that all addressing tends to be absolute instead of Program Counter relative. Also, it is not immediately easy to set up the relative address between one item and another. Since the "window definition" is held together by a set of relative pointers Tony Tebby decided to go straight to the "working definition". Inside this, all relative pointers have been replaced by absolute addresses, and so it is amenable to C68.

"C68 can't cope with relative pointers"

2. It is just conjecture on my part, but I suspect that the problem of having to deal with relative (word) pointers in the "window definition" has deterred those producing the SuperBASIC version of QPTR.

Relative (word) pointers are difficult?

How does my approach work?

1. For C68 I have written a special C function called "getsze" which has to be called at the start of any C68 - PE program. This function replaces pointers to a set of absolute addresses by word-sized relative pointers, and - Hey Presto - without any more trouble the window

definition is set up! You might wonder why the name of the routine is "getsize". In fact this is because it also performs another vital service, which is to calculate the amount of space needed for each of the possible manifestations of window - one for each of the repeated sections.

2. TurboPTR builds up the "window definition" by filling several spaces each separately allocated from the Heap. One practical difficulty with word pointers is the rule that if the target address is more than 32768 bytes away from the word pointer the word pointer must instead point to a long word relative pointer to the actual target. That's fine if there is a long word available within striking distance. If not - disaster! In TurboPTR I have ensured that in every part of the structure there will be enough long words near the word pointer

to satisfy the maximum need.

Incidentally, in TurboPTR the problem of size required for a "working definition" is solved in much the same way as for C68, but it is totally hidden from the programmer inside the SuperBASIC extension words used by the system.

TURBO

I was quite relieved and ready for a rest when at last TurboPTR appeared to work. And then I read, in the December 1999 edition of QUANTA, Mark Knight's statement that, with Dave Gilham, I would be doing programming for a new version of Turbo.

Once again I couldn't resist the challenge and, with the help of a QL World article by, plus encouragement and advice from, Simon Goodwin, I have eventually produced a version of Turbo which works in PE. It works on my Q40, my QXL and

my Gold Card, either with SMSQE or with JS rom.

I have also managed to compile TurboPTR programs, but only after extensive changes to them and to TurboPTR itself. These were due to three things:

- a. Compiled PE programs differ from SuperBASIC ones.
- b. Turbo requires stricter standards than SuperBASIC.
- c. Turbo falls over sometimes if an expression is too complicated.

For example

```
k = fn_a(fn_b)
```

may have to be replaced by

```
k = fn_b : k = fn_a(k)
```

FUTURE

I intend to see if Turbo, already a formidable weapon, can be sharpened up even more. Perhaps I'll even be able to produce an optimiser to run between Parser and Codegen as adumbrated on page M-5 of the Turbo v2.0 Manual.

Programming ProWesS in SBASIC- and why not?

Wolfgang Lenerz

You cannot have been even mildly interested in the QL over the last few years and not have heard about Prowess. However, I have noticed quite a few times that people don't really know what it is and what to do with it. So here is a small explanation of Prowess and how to use it in your own programs:

First of all, a certain number of concepts should be set out as Prowess may seem a little intimidating at first. In fact, it is dead easy. The expression "complex but not complicated" really applies here: complex, because there are many new things to learn, but not complicated, because what you learn is not difficult to learn in itself.

Some General Thoughts on ProWesS

Prowess, like the Pointer Environment's WMAN, is a window manager. In other words, it doesn't do anything else than help you to set up windows,

print them on the screen, change their content and pass the user's action(s) to the program. You will still have to program the rest of your program yourself! Whether you prefer WMAN or Prowess is largely just that - a personal preference.

Generally speaking, in favour of Prowess over WMAN is the fact that Prowess uses Proforma, which lets you use nice vector fonts, i.e. fonts which can be of 'practically') any size and still look nice. WMAN, on the other hand, is limited to the normal QL fonts. There was some talk a few years ago of a vector fonts system for normal QL windows (and thus for WMAN), but this seems to have been abandoned.

The disadvantage of Prowess over WMAN is due, in my eyes, to the large specification of system resources Prowess and Proforma need: without a relatively fast machine, and a large amount of free memory, both are really unusable. But this is simple the ransom to pay for having things like scaleable fonts, which have to be drawn very carefully on the screen.

Programs running under Prowess still need the Pointer Interface itself (though not the WMAN window manager since that is replaced by Pro-

wess). The Pointer Interface handles very low-level stuff such as determining in what window the pointer is located. The window manager, on the other hand, is there to make sure that windows can be drawn nicely.

Thus Prowess, like any window manager, is only responsible for the window part of your program. If, for example, you create a program to copy files from one directory to the other, you will create a window where the user can determine from what directory files will be copied to and from. This part of your program will be handled by Prowess. The actual copying routines, on the other hand, are independent of the window manager, and have nothing to do with Prowess.

Programming for Prowess thus means designing and creating your window(s). This, in turn, is achieved by creating Prowess "objects", possibly changing them, "activating" the main object, and then removing all of the objects once they are no longer needed.

Objects

The entire window itself is an object, which in turn, contains other objects, which can contain yet other objects - and so on. You, the programmer, never really manipulate the window itself, only the objects of that window. Normally, the first object you would create would be the "outline" of the window - all other objects are then fitted, or poured, into that outline object.

The outline of the window is just a sort of container whose purpose is to contain (or "own") all of the other objects (some of which are very powerful). Thus, the outline object is the owner of all the other objects. The combination of the outline object together with all that it contains, is known as a **system**. A program can own several systems. For example, you might have one system, which corresponds to the main window. At any time, you might open (pull down in WMAN parlance, or activate in Prowess) another window, for example to show some options. This would be another system. However, all systems are built on the same model: one outline object followed by objects within the outline.

In other words, the first thing to do when programming under Prowess is to create the outline object. Once this is done, you can create other objects for, and contained by, it. The entire suite of objects is a system. Once you have created a system, you can activate it. Activation means that the window will be drawn on the screen, and Prowess then processes the keystrokes/mouse clicks. Indeed, once the window is drawn on the screen, Prowess waits for the user to hit/do ob-

jects (such as menu items) or press certain keys. The information on what happened is then given to your program, which is thus informed about a HIT or DO on the corresponding menu item, so that it can react accordingly.

Changing (Parts of) Objects

It may happen that you want to change an object. Let's suppose that an object contains a string which tells the user what a default directory might be. The user now changes this default directory. It would be useful to change the object, or its contents, so that it reflects the new default directory. It would be fastidious to (i) remove the object, (ii) create a new one and (iii) activate it. Thus it is possible to change an object, or at least many aspects of an object: in the example above, it might be possible to change the content of the object, (i.e. the string with the default directory) and also other aspects, such as the font to use to display this string.

Quering Objects

Likewise, sometimes it can be necessary to ask an object something about itself. Even though you created them, you don't always know everything about the objects (ha, if it ain't magic...)! So, you can query an object about itself. As an example, you might have an object that shows you a directory, and allows the user to change that directory. At the end, you will want to know exactly what directory was chosen by the user - you then query the directory object and ask it what its current directory is set to!

Removing Objects

When you no longer need an object, you can just remove it. It then no longer exists. It is interesting to know that, when you remove an object, you also remove all the objects that are owned by it!

Programming ProWesS in SBASIC

It is possible to write programs that use Prowess. Prowess more or less assumed that most programs using it would be written in 'C' or assembler, a Basic interface was not initially foreseen. However, this has since been provided (if you can program in C or assembler, then you probably don't need this series, you should have a look at the (pretty good!) documentation provided, and at the sample programs).

So it is possible to write programs in SBASIC that make use of Prowess, just as it is possible to write programs in Basic to use the Pointer Environ-

ment. To be true, the only programs I know of that are written in SBASIC and make use of Prowess are the example programs that come with it, and Agenda (a shameless plug for one of my programs - ahem!).

Please note that I did write programming in SBASIC, and not Basic or SuperBasic. Indeed, it is important to note that the SBASIC interface that I wrote for Prowess will only work with SBASIC as contained in the SMSQ (/E) operating systems - it will not work in normal SuperBasic. Compiled programmes (using QLiberator) will work on all machines (there will be a special section on compilation later). Please note that on machines other than those running SMSQ(/E) YOU MUST NEVER ATTEMPT TO RUN AN UNCOMPILED BASIC PROGRAM USING THE PROWESS SBASIC INTERFACE. If you do, your machine will almost certainly crash. Any damage to you, your computer or your data will be your own fault, and nobody will accept any responsibility for this. YOU HAVE BEEN WARNED!!!!

(A word of explanation for the more technical minded: this restriction is not due to the SBASIC interface itself. Simply stated, the interface calls the routines provided by Prowess - and these make use of the register A6. As you probably know, use of this register is strictly forbidden for programs using the normal SuperBasic interpreter. The SMSQ(/E) Sbasic, as well as QLiberated programs, do not have this restriction).

So, how do you write programs in Sbasic that make use of the facilities provided by Prowess? Simply with the new keywords that are provided by the Prowess Sbasic interface (normally in a file called PWbasic_rext).

Thus, a review of the new keywords is perhaps the best way of approaching the problem. I'll also explain new concepts as and when they arise.

1 - PWcreate: Create an Object: owners, types and tags

The PWcreate function is used to create any and all objects. As was mentioned above, creating an object is one of the main steps in ProWesS programming. It is also generally the very first step - you can't really do anything with (or rather in) Prowess until you have created some objects. So this step is paramount, and, for something so important, it is actually achieved quite easily, by using the PWcreate keyword. This is a new function with the following syntax:

```
my_object = PWcreate (owner, type [{, tag }])
```

This will create an object, and the SBasic variable **my_object** will be (or rather point to) that object. From now on, whenever you need an object for doing something, you can use **my_object**. This is actually no different from using any other variable returned by a function in SBasic. Sometimes, you will come across the term "Object ID" which is just this variable, too!

The parameters passed to the function can look more daunting than they are. Alright, first of all, the square brackets '[' indicate an optional item, as usual. The curly brackets '{' indicate an item that may be repeated any number of times, so that `[[tag]]` means that any number of "tags" (whatever they may be) can follow the type parameter.

What this function does is create the object "my_object" according to the parameters you have given. You now have a Prowess object!

The parameters for the **PWcreate** function are logically structured and correspond to some important concepts within ProWesS.

The owner parameter

The **owner** of my_object is the object that will own my_object. Thus, this parameter, i.e. owner, must be a valid Prowess object. As was mentioned above, objects usually belong to other objects. With this parameter, you tell ProWesS who my_object will belong to, i.e. what object will own the newly created object "my_object".

Of course, the very first object you create can't belong to anybody, as there is no object yet that could own it, so the owner is... 0. If you now create a second object, and pass **my_object** as owner, then the owner of that second object is **my_object** as returned by the first call to **PWcreate**.

The type parameter

This parameter of the **PWcreate** function tells the software what **type of object** you are creating. Indeed, in Prowess, there are many different types of objects, which will do very different things.

Normally, the first object you would create would be the outline of the window, which, as mentioned above, is just a sort of container enclosing all the other objects in the window. So the type of this object would be that of an outline. Another object could be, for example, a menu item for this outline, or an "infotext" item.

Each type has one special "type word" associated with it, which is predefined and carefully explained in the Prowess manual, even though we will come back to them later on.

Thus, if you want to indicate that the object is to be of the type outline, you would use the type `"PW('TYPE_OUTLINE')"` (the exact meaning of this will be explained later).

The *tag* parameter(s)

The various types provided by Prowess are very different from each other and will achieve very different results. But there are only a few of them and it seems logical that you will have to define some more details for each of them. This is why the type parameter can itself be followed by other parameters - for some strange reason, these parameters are called **TAGS** and define exactly what characteristics the type should have.

For example, if you create an object of the type `loose_item` (which is a loose menu item) you, the programmer, should be able to determine whether this object is to contain a string or an icon, and if it is a string, what the string is. In other words, when creating this loose menu item object, you would follow the type by a 'tag' saying that the item is of the type string, and the tag would then be followed by the string.

Tags are always dependent on a type. An important aspect of programming in Prowess is to understand that each type has its own tags, even though they may achieve something similar, like setting a text for the object. The description of each type also contains the description of the tags that this type supports.

The tags are thus parameters for the types. If you remember, in the introduction it was mentioned that Prowess is not complicated but complex,

since there is a lot of information to absorb. Most of that information concerns the tags for each type, and what they mean and do. To be quite frank, I personally can never remember what type can use what tags, and what each tag does. I just keep a copy of the manual handy and look it up whenever I need to. If you want to learn them by heart, though, feel free to do so.

In most cases, all tags that can be used when you create an object, can also be used when you change an object, even though there are a few exceptions. On the other hand, all tags used for a change can always be used during the creating of an object.

An example

Here is an example of the `PWcreate` function. It creates an outline (this is the type) which has a quit item (one of the tags for this type). The owner of this outline will be 0.

```
my_outline = PWcreate (0,PW('TYPE_OUTLINE'),  
PW('OUTLINE_QUIT'))
```

Here, the owner is 0. The type is `"PW('TYPE_OUTLINE')"` and the tag for this type is `"PW('OUTLINE_QUIT')"`. With this command, you have created a Prowess object. This object is an outline. This outline object will contain a special menu item (with the string "Quit"). If you now activate this outline, the window will magically appear on the screen!

We will see how to activate this object in the next instalment of this series.

Gee Graphics! (on the QL?) - Part16

Herb Schaaf

A Point and a Line in 3 Space

In GG#15 we explored 2 skew lines in 3 space, but skipped over a simpler(?) question; what is the shortest distance between some Line in space and any single Point? I looked in vain for a solution in several textbooks. What I did find was a way to measure the angle between two lines in 3 space based on their direction co-

sines. After some thought (and sleep) it occurred to me that solving the 'Point to Line' had probably been left as an 'obvious' and/or 'trivial' 'exercise for the reader'.

Here is an algorithm: Create a line segment between the point in space and one point on the Line in space. Then solve for the angle between the line segment and the Line. Then use the length of the line segment with the Sine of the

angle to find the shortest distance from Point to Line, and then again with the Cosine of the angle to work out the location along the Line that is closest to the Point.

If the angle is zero then the Point is actually on the Line and the shortest distance is also zero. If the angle is 90 degrees then that one point on the Line is also the closest point to the Point in space.

Given one starting point on a Line there are at least 3 ways to fix a Line in 3D space:

- (1) a second fixed point on the Line
- (2) direction numbers (or offsets)

(3) direction angles (or cosines)
Take a look at the PROCedure setup_line3d which tries to work out values for all three

cases when given one known case.
As far as I can tell the program Point_to_Line_3D_bas seems to be working. Next time I hope

to show the solution graphically in a set of orthographic views.

```
100 REMark Point_to_Line_3D_bas
110 REMark HL Schaaf April 30, 2000
120 REMark to go with GG #16
130 :
140 CLEAR
150 explain
160 get_inputs
170 setup_line3d ans MOD 3
180 show_input_data
190 solve_Pt2Line
200 show_results
210 REMark views_in_3D
220 :
230 DEFine PROCedure setup_line3d (case)
240 LOCAL i
250 REMark cases are 0, 1, 2
260 DIM line3d(2,3)
270 REMark must have at least one point = point1
280 REMark if given direction cosines use case = 0
290 REMark if given 2 points use case = 1
300 REMark if given offsets (direction numbers) use case = 2
310 REMark elements 0, [ 1 to 3 ] for direction cosines
320 REMark elements 1, [ 1 to 3 ] for point1 x, y, z
330 REMark elements 2, [ 1 to 3 ] for point2 x, y, z
340 REMark use element 0, 0 for sum of COS^2 == 1
350 REMark use element 1, 0 for length of line segment
360 REMark use element 2, 0 for success in setting up ?
370 :
380 REMark must have one point on line, get it now
390 FOR i = 1 TO 3
400   line3d(1,i) = Pt_1_of_Line(i)
410 END FOR i
420 :
430 REMark case 0 with direction cosines
440 IF (case == 0) THEN
450   FOR i = 0 TO 3
460     line3d(0,i) = dircos(i)
470 REMark use segment with unit length to set point2
480     line3d(2,i) = line3d(1,i) + dircos(i)
490   END FOR i
500   line3d(1,0) = space_btwn(line3d(1,TO),line3d(2,TO))
510   line3d(2,0) = 1
520 END IF
530 :
540 REMark case 1 with two different known points on same line
550 IF (case == 1) THEN
560   FOR i = 1 TO 3
570     line3d(2,i) = Pt_2_of_Line(i)
580   END FOR i
590   find_direction_cosines
600   line3d(2,0) = 1
610 END IF
620 :
630 REMark case 2 with offsets for direction numbers
640 IF (case == 2) THEN
650   FOR i = 1 TO 3
660     line3d(2,i) = line3d(1,i) + Offsets(i)
670   END FOR i
680   find_direction_cosines
690   line3d(2,0) = 1
700 END IF
710 :
720 IF NOT(line3d(2,0)) : PRINT "What case ? ":STOP
730 END DEFine setup_line3d
740 :
```

```

750 DEFine PROCedure find_direction_cosines
760   line3d(1,0) = directions_from(line3d(1,T0),line3d(2,T0))
770   FOR i = 0 TO 3
780     line3d(0,i) = direction_cosines(i)
790   END FOR i
800 END DEFine find_direction_cosines
810 :
820 DEFine PROCedure explain
830 WINDOW 512,256,0,0 : PAPER 0 : CLS
840 WTV : MODE 4 : PAPER 2: INK 7 : CSIZE 0,0 :CLS
850 PRINT \,"This is an attempt at solving the problem"
860 PRINT , "of finding the distance and direction from a "
870 PRINT , "Point in 3D space to a Line in 3D space."
880 PRINT\,"You are invited to create a Line by first entering"
890 PRINT , "the x, y, z values of some Point #1 on the Line."
900 PRINT\,"You then have a choice of methods that will define the Line:"
910 PRINT , " 1 - Set x, y, z, for another Point #2 on the same Line"
920 PRINT , " 2 - Use x, y, z Offsets for Point #2 relative to Point #1"
930 PRINT , " 3 - Use Direction angles measured from the x, y, z axes"
940 PRINT , , "(two suffice, the 3rd will be calculated)"
950 PRINT\,"Then enter x, y, and z for a 3rd point in 3D space."
960 PRINT\\ , , , "Touch [space bar] to continue"
970 PAUSE
980 DIM Pt_1_of_Line(3)
990 DIM Pt_in_3D(3)
1000 DIM Pt_2_of_Line(3)
1010 DIM Offsets(3)
1020 END DEFine explain
1030 :
1040 DEFine PROCedure get_inputs
1050 CLS
1060 get_point Pt_1_of_Line, " for Point 1 on the Line"
1070 get_line_fix
1080 get_point Pt_in_3D, "for a 3rd Point in 3D space"
1090 END DEFine get_inputs
1100 :
1110 DEFine PROCedure get_point(point_vector,name$)
1120 DIM a_point(3)
1130 PRINT
1140 INPUT"Please enter the x "&name$,a_point(1)
1150 INPUT"Please enter the y "&name$,a_point(2)
1160 INPUT"Please enter the z "&name$,a_point(3)
1170 FOR i = 1 TO 3 : point_vector(i) = a_point(i) : END FOR i
1180 END DEFine get_point
1190 :
1200 DEFine PROCedure get_line_fix
1210 REPEAT get_choice
1220   PRINT , "please choose method for fixing Line in 3D space"
1230   PRINT , "by touching appropriate number key"
1240   PRINT \ , "1 - Setting x, y, z, of Point 2 on the Line"
1250   PRINT , , "2 - Using x, y, z Offsets from Point 1"
1260   PRINT , , "3 - Using Direction angles;"
1270   PRINT , , "measured from the x, y, z axes"
1280   PRINT , , "two suffice, the 3rd will be calculated"
1290   ans$ = INKEY$(-1)
1300   ans = CODE(ans$)-48
1310   IF ans >0 AND ans <4 : EXIT get_choice
1320 END REPEAT get_choice
1330 :
1340 SElect ON ans
1350   = 1 : get_point Pt_2_of_Line, "for Point 2 on the Line"
1360   = 2 : get_point Offsets, "offset for the 2nd Point on the Line"
1370   = 3 : get_direction_angles
1380   = REMAINDER : get_line_fix
1390 END SElect
1400 END DEFine get_choice
1410 :
1420 DEFine PROCedure get_direction_angles
1430 PRINT,"Setting any two direction angles will cause"
1440 PRINT,"the third one to be calculated"
1450 min_ang = 0
1460 DIM dirang(3)

```

```

1470 DIM axes4(3)
1480 axes$ = "XYZ"
1490 FOR ax = 1,2
1500   pick_axis
1510   get_angle
1520 END FOR ax
1530 DIM dircos(3)
1540 FOR i = 1 TO 3
1550   IF axes$(i)="0" THEN
1560     dircos(i) = COS(RAD(dirang(i)))
1570     dircos(0) = dircos(0) + (dircos(i))^2
1580   ELSE
1590     last_axis = i
1600   END IF
1610 END FOR i
1620 dircos(last_axis) = SQR(1-dircos(0))
1630 dirang(last_axis) = DEG(ACOS(dircos(last_axis)))
1640 dircos(0) = 1
1650 END DEFine get_direction_angles
1660 :
1670 DEFine PROCedure pick_axis
1680 REPEAT get_axis
1690   PRINT\,"Choose an axis, ";
1700   FOR i = 1 TO 3
1710     IF (axes$(i) < "0") THEN
1720       PRINT axes$(i);
1730       IF (i < 3): PRINT " or ";
1740     END IF
1750   END FOR i
1760   PRINT,"by touching the appropriate key"
1770   REPEAT valid_key
1780     axis = ((CODE(INKEY$(-1))) MOD 32) - 23
1790     check$ = CHR$(axis+23+64)
1800     IF (check$ INSTR axes$) : EXIT valid_key
1810     PRINT "Please choose a valid axis"
1820   END REPEAT valid_key
1830 IF axis > 0 AND axis < 4 :EXIT get_axis
1840 END REPEAT get_axis
1850 axes$(axis) = "0"
1860 END DEFine pick_axis
1870 :
1880 DEFine PROCedure get_angle
1890 PRINT ,"Enter the angle from the positive ";check$;" axis"
1900 PRINT ,"in Degrees from ";min_ang;" to ";180-min_ang,
1910 INPUT ; dirang(axis)
1920 IF ax = 1 THEN
1930   min_ang = 90 - dirang(axis)
1940 IF min_ang < 0 : min_ang = -min_ang
1950 END IF
1960 END DEFine get_angle
1970 :
1980 DEFine FuNction space_btwn(pt1,pt2)
1990 LOCAL i
2000 sum_sqrs = 0
2010 FOR i = 1 TO 3
2020   sum_sqrs = sum_sqrs + (pt2(i) - pt1(i))^2
2030 END FOR i
2040 RETURN SQR(sum_sqrs)
2050 END DEFine space_btwn
2060 :
2070 DEFine FuNction directions_from(pt1,pt2)
2080 DIM direction_cosines(3)
2090 segment_length = space_btwn (pt1,pt2)
2100 IF (segment_length) THEN
2110   FOR i = 1 TO 3
2120     direction_cosines(i) = (pt2(i)-pt1(i))/segment_length
2130     direction_cosines(0) = direction_cosines(0) + (direction_cosines(i))^2
2140   END FOR i
2150 ELSE
2160   PRINT#0;"Identical points !!" : PAUSE
2170 END IF
2180 RETURN segment_length
2190 RETURN direction_cosines

```

```

2200 END DEFine directions_from
2210 :
2220 DEFine PROCedure show_input_data
2230 CLS
2240 PRINT "The input data for the Line are as follows:"\
2250 PRINT\ "Point 1 on Line", "x", "y", "z"
2260 PRINT "      ", line3d(1,1 TO 3),
2270 PRINT\ "Point 2 on Line", "x", "y", "z"
2280 PRINT "      ", line3d(2,1 TO 3),
2290 PRINT\ "Direction cosines are:"
2300 PRINT line3d(0,1 TO 3),
2310 PRINT\ \ "Direction angles in Degrees are:"
2320 FOR i = 1 TO 3
2330   PRINT DEG(ACOS(line3d(0,i))),
2340 END FOR i
2350 PRINT \ \ \ "The input data for the Point in space are:"
2360 PRINT\ "Point in 3D Space", "x", "y", "z"
2370 PRINT "      ", Pt_in_3D(1 TO 3),
2380 PRINT\ \ "Touch [space bar] for solution"
2390 PAUSE
2400 END DEFine show_input_data
2410 :
2420 DEFine PROCedure solve_Pt2Line
2430 DIM P1toP3_cos(3)
2440 Cos_of_Angle = 0
2450 P1_P3 = directions_from(line3d(1,TO),Pt_in_3D)
2460 FOR i = 1 TO 3
2470   P1toP3_cos(i) = direction_cosines(i)
2480   Cos_of_Angle = Cos_of_Angle + direction_cosines(i)*line3d(0,i)
2490   IF Cos_of_Angle == 1 : Cos_of_Angle = 1
2500 END FOR i
2510 Rad_Angle = ACOS(Cos_of_Angle)
2520 Pt_to_Line_distance = P1_P3 * SIN(Rad_Angle)
2530 Foot_offset = P1_P3 * COS(Rad_Angle)
2540 REMark place on Line closest to Point in 3D space
2550 DIM Ft_of_Pt2Line(3)
2560 FOR i = 0 TO 3
2570   Ft_of_Pt2Line(i) = line3d(1,i) + (Foot_offset * line3d(0,i))
2580 END FOR i
2590 REMark cross check for agreement ?
2600 Foot_to_3DPt = directions_from (Pt_in_3D,Ft_of_Pt2Line)
2610 IF NOT(Foot_to_3DPt == Pt_to_Line_distance) THEN
2620   PRINT #0; "Distance check error ?"
2630   PRINT#0; Foot_to_3DPt;" - ";Pt_to_Line_distance;" = ";
2640   PRINT#0; Foot_to_3DPt - Pt_to_Line_distance
2650 PAUSE
2660 END IF
2670 DIM Pt2Line_dir(3)
2680 FOR i = 0 TO 3
2690   Pt2Line_dir(i) = direction_cosines(i)
2700 END FOR i
2710 END DEFine solve_Pt2Line
2720 :
2730 DEFine PROCedure show_results
2740 CLS
2750 PRINT\ "The distance from the point in space to the line is ";
2760 PRINT Pt_to_Line_distance
2770 PRINT\ "The closest point on the line to the point in space has"
2780 PRINT "the following x, y, and z : "
2790 PRINT\ , Ft_of_Pt2Line(1 TO 3),
2800 PRINT\ \ "The direction cosines from Point in space to the Line are:"
2810 PRINT\ , Pt2Line_dir(1 TO ),
2820 PRINT\ \ "The Angles in Degrees being "\,
2830 FOR i = 1 TO 3
2840   PRINT DEG(ACOS(Pt2Line_dir(i))),
2850 END FOR i
2860 REMark PRINT \ \ \ \, "Touch any key for views in 3D"
2870 PRINT \ \ \ \, "Touch any key to exit"
2880 PAUSE
2890 CLS
2900 END DEFine show_results
2910 :
2920 REMark end of listing of Point_to_Line_3D_bas for GG #16

```

Some Notes on EPSON ESC/P2 Printer Codes

Don Atkins

Recently I've been programming in Psion's Archive to print sheets of adhesive address labels on the Epson Stylus Colour 740. The hardest part was setting the printer parameters to get the printing in the desired position on the labels which, in my application, are 2 across by 8 down the sheets, i.e. 16 labels per sheet. Of course the stationery is designed in metric measure whilst the printer criteria use imperial - not the easiest combination!

I had downloaded the voluminous Epson coding manual so, theoretically at least, had all the information I would need but I found the almost complete lack of worked examples in the manual a considerable disadvantage, especially when it came to dealing in INT and MOD for setting top and bottom margins and page length.

One unexpected thing I found early on in my travails was that the 740 does not offer the option to remove any automatic line feed - it's not supported, although it apparently is in some earlier Stylus Color models.

I used the "Set Defined Unit" and "Set Page Format" coding to fix the top margin and page length. The Page Format normally requires sending 9 values to the printer which, when using Archive, translates into 17 because the chr(0) needs to precede every following value.

Setting the defined unit is straightforward enough with the ESC (U command and I chose 20/3600 (0.0055 inch) for my unit.

For the page format, which determines top and bottom margins, both measured from the top of the paper, it's necessary to calculate the variables th, tl, bh and bl using somewhat clumsily presented formulae.

Take as an example th which the manual shows is:

$$\text{INT} \left\{ \frac{\text{top margin} \times \frac{1}{\text{defined unit}}}{256} \right\}$$

but the defined unit is in the form $\frac{n}{3600}$

It's much clearer, in my view, to present the formula as

$$\text{INT} \left\{ \frac{\text{top margin} \times 3600}{n} \right\} / 256$$

then, if the top margin is to be 0.15 inch and n is chosen as 20, we have

$$\text{INT} \left\{ \frac{0.15 \times 3600}{20 \times 256} \right\} \text{ which gives } \text{INT} \left\{ \frac{27}{256} \right\}$$

so the integer, INT, is zero and the remainder, MOD, is 27.

Similarly with bh

$$\text{INT} \left\{ \frac{\text{bottom margin} \times 3600}{n} \right\} / 256$$

Then, if the top margin is to be 11.70 inch, for A4, we have:

$$\text{INT} \left\{ \frac{11.7 \times 3600}{20 \times 256} \right\} \text{ which gives } \frac{2106}{256}$$

so the integer is 8 and the remainder is 58, (2108-(8*256))

You always need to keep the maths in the vulgar fraction form with the divisor 256 on the bottom so as to get the correct MOD. Hence, for my coding to set a top margin of 0.15 inch and a page length of 11.7 inch the complete string is:

```
lprint chr(0)+chr(27)+"("+chr(0)+"c"+chr(0)
+chr(4)+chr(0)+chr(0)+chr(0)+chr(27)+chr(0)
+chr(0)+chr(0)+chr(58)+chr(0)+chr(8);
```

So much for the page format problem. Another difficulty I encountered was in setting 2 different left hand margins on each line of printing, corresponding to the left hand and right hand labels on the sheets, because it is necessary to first print a line on the left hand label then switch to the first line on the right hand label and so on right down the sheet.

If you try this with the 740 by using the obvious code, ESC l n, where n is the required number of columns, choosing n to suit, it doesn't work because the printer ignores the first left hand margin instruction and prints nothing until it reaches the righthand margin instruction!

To overcome this I used ESC l n to set the left hand margin for the left hand label, printed the string str1\$, then called a proc in the form

```
lprint rept(" ",40-len(str1$));str2$
to print the string str2$ on the left hand margin of
the right hand label.
```

Whether any of this is likely to be of interest or help to anyone else, I'll leave our editor to decide.

My Black Box

by Alex Wells

I came across a quite unique Aurora system at the Manchester Quanta AGM and persuaded its owner to write his side of its story in the hope that readers would find it interesting - Editor

This is not another historical survey of the QL scene, but it does start about 10 years ago. I had been wanting to try a spread-sheet and we did not have easy access to suitable machines at work then - we used PDP11 or VAX machines to do our sums. I bought a QL for £40 from Curries which seemed reasonable as they had started out at £400. I planned to play with the spread-sheet for a month and then throw the QL away, as it had acquired a terrible reputation with an incomplete operating system and those dreadful microdrives!

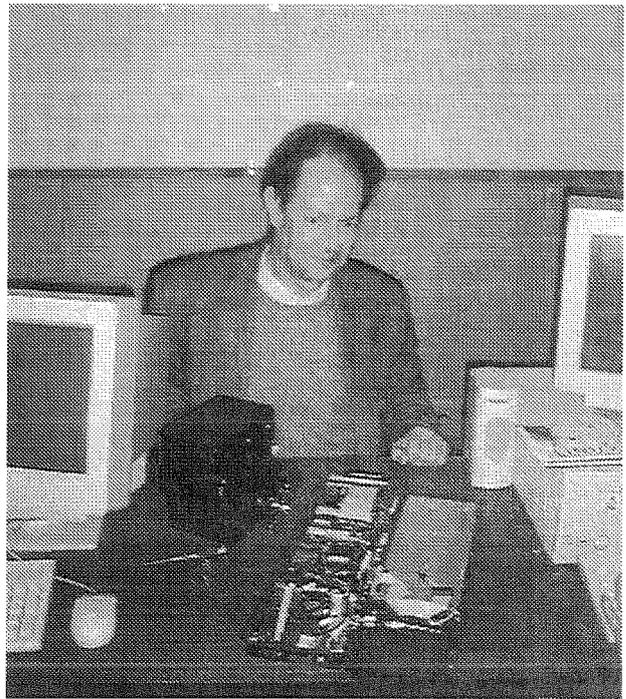
Spread-sheets and word-processors appeared on our terminals of course, and were very handy in a black-and-white cursor-key environment. We continued writing and compiling Fortran for our large-matrix jobs. At home I was finding it quick and easy to write Super(!)Basic - throw out the QL next month? The micro-

drives were neat, quick and reliable for the time - wait until they show signs of wear? I was sure this machine would not tempt me to tinker or mend it... But what was a toolkit for, and what was TK2?

What were QPac2 and Q-Liberator? Could a QL really talk to another with the network lead? What about a data-base for my music collection? How about trying 'C', Lisp?

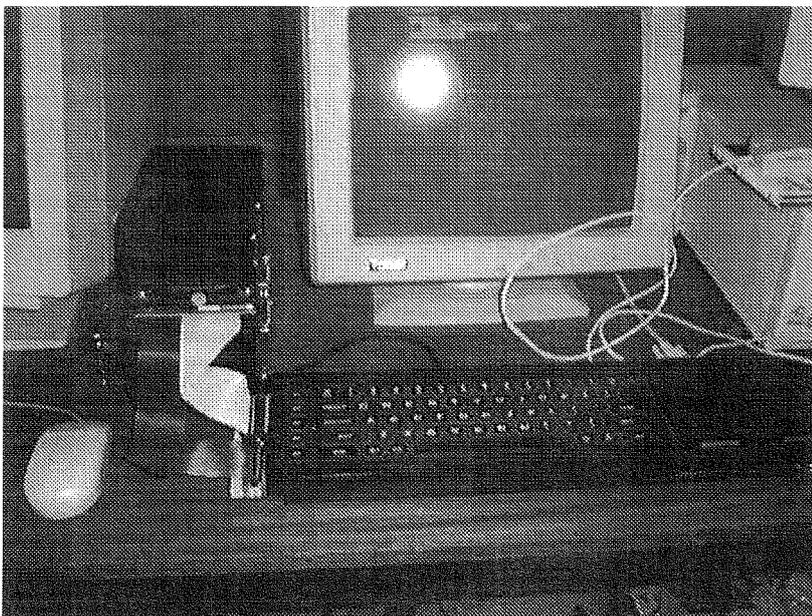
I met someone who told me about a Quanta meeting. Various mysteries were resolved at this meeting, thanks to all the friendly and helpful folks there. Quanta gained a new member, and I bought Computer One Forth. Hmmm! F4th. This may take some time - keep the QL a bit longer?

So here I am 10 years later with



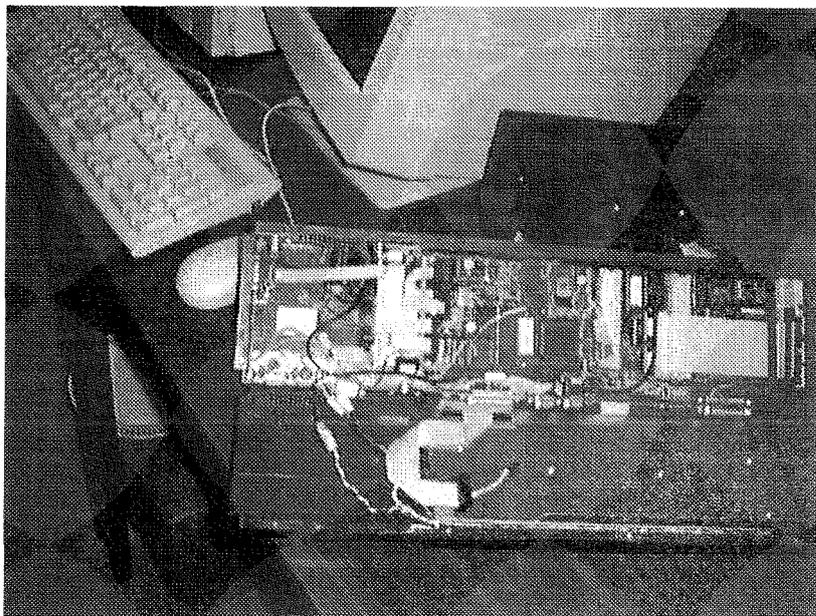
various QL clones running SmSq (with an 'E' on the end?). Q-Liberator is rarely needed. The quirky extra facilities like the Minerva 2nd screen are distant memories. The Aurora/Super Gold Card system still refuses to talk to another one with Sernet and a null-modem lead. There is also a Q-40 with SmSq, and I wonder whether I will ever become familiar with the Linux filing system. The great plus point is that they are very usable under SmSq. If you are wondering what the connection is between any of this and the title 'My Black Box', your patience will soon be rewarded...

When Aurora, Qubide and hard drives appeared I assembled a machine with a Super Gold Card and back-plane in a desk-top box. This box had already been used with a QL motherboard squeezed in. Each version took some time, and some drilling and bending was needed to produce a satisfactory arrangement with proper plugs and sockets on the back. However it is quite difficult to move the desk-top box around with a screen and keyboard. This was long before the MinisQL had been invented.



A lighter and more compact configuration was needed, using aluminium or light alloy boxes. The boxes could be cut to fit an old power pack and hard drive. Would all the other stuff fit inside the QL case minus QL motherboard?

After deciding not to include a single floppy inside the black box, the power-pack/hard-drive boxes had to carry the 2 floppies on top, which was not very elegant - power-pack too wide. The external fan is not an improvement either, but it does its job with a resistor in series to reduce the noise. The ribbon cables for these devices could be run together to the left-hand end of the QL black box. There is only about 5 inches of ribbon exposed. The power supply goes to the back of the black box, where it connects to the Aurora through a 4-pin hard-drive socket - just like the Q-40 power cord 2 or 3 years later! This is where we reach the point of these notes. Dilwyn took a photo of my favourite black box at the April Quanta meeting in Manchester. He threatened to publish it if he was short of material. As YOU have not sent him anything more interesting (and ANYTHING would be more interes-

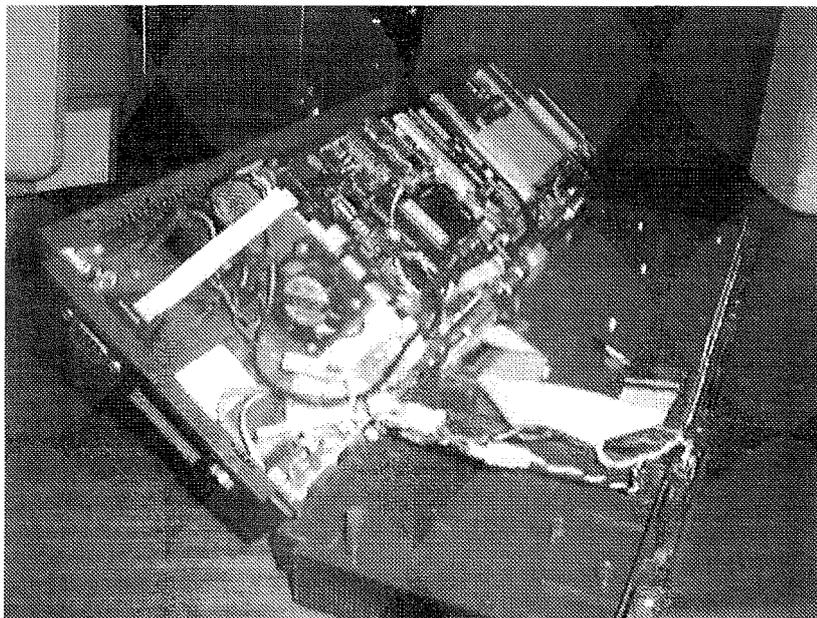


ting!) you are now being exposed to a description of a much-modified QL case.

The black box holds the Super Gold Card, Qubide and Aurora, with the SGC protruding at the user's left. The 5 volts for the SGC arrives through the Qubide, and the heat-sink is removed, so the Qubide ribbon can exit to the real world easily over the SGC memory. It terminates in a socket on the end of the box. The left-hand side of the black box also has the floppy drive ribbon and the original parallel printer ribbon from 'Miracle Systems' attached to the SGC.

The Aurora gets its power through the optional 4 pins, and holds a SuperHermes and Minerva. A ribbon cable connects the Aurora to the QL keyboard via a small collection of diodes mounted at the back of the QL lid. The diode board holds the edge connectors from a QL which mate with the shortened keyboard ribbons. The keyboard membrane and ribbons are not disturbed when I remove the top. If I want to use a conventional keyboard it can be plugged into the front of the black box. Another little board holds a set of 4 miniature switches on the right-hand end to control the Aurora display, replacing the internal jumpers. The normal output is a monitor at 768 x 384 (so I can read the screen with normal sized QL characters) but the 'Cub' display at 576 pixels wide can be quite useful.

SuperHermes is used primarily to give a 'high speed' modem connection, as I use a Qimi mouse connected to Aurora. SuperHermes keeps track of the total hours used. Minerva has an I2C-IO connection at the back of the box, and the RTC clock battery is nearby. Minerva usually knows which hard drive is in use at boot-up time



so the starting process is normally uneventful! (I tried the Qu-bide version 2 chips, and soon reverted to the v1.56 set.)

There is a miniature fan carefully installed in a bulkhead running across the box, but I suspect it is not needed. The QL speaker remains in place, and of course the microdrives were removed, giving a rather empty 'plenum chamber' at the right-hand end.

Before the bits were fitted in the black box, lots of extra holes were cut for the connectors. I remade all the split and broken screw-holes at the front, so it can be taken apart and re-assembled without the old plastic box problems. Fortunately the 4 long screws at the back of the case do not seem to cause any damage - repairing them would be tricky! The inside of the box was nickel-sprayed and varnished as the QL with SGC used to fall



over when very close to hostile devices.

When it was all finished with a tiny reset button, net plugs, serial ports 1,2 & 3 (serial-1 is connected to 9 and 25 pin 'D' plugs - I only use one at a time!) there were 16 connectors, 2 switches and 4 LEDs around

the black box, plus the keyboard on top.

And the most amazing thing is - it works!

Last July it only took an hour or so to pop the Q-40 into a conventional case - 'plug and play'! But: do I want it to be more portable ?

A busy Knight

Mark Knight

I have done a lot of work on my QL in early 2000, during December and early January because I was too ill to go out much, through February and early March because that was just the way I decided to spend my spare time. Some of this work has been database processing for a local business, using a combination of Editor and a custom SuperBASIC program (compiled with Turbo to make it go at a usable speed). I took delivery of the data on DOS disks, transferred it to the QL, processed it and then copied it back to DOS disks. Working on this database was boring to say the least, but I couldn't back out after signing the contract and I need the cash anyway. (One of the rea-

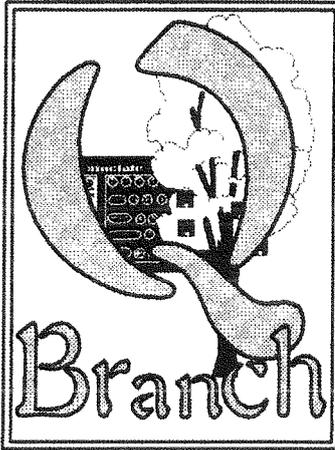
sons for this is that I recently became temporarily addicted to shopping, often for things I didn't need. Although I don't consider any of the things I have bought to be a waste of money, I think I should have waited longer for some of them... terrible, this addiction, I'm now so desperate for money I have to work with PCs... Anyway I'm sure you don't want to read about my personal problems). To relieve the boredom I have also been working hard on updating the Editor again, with the result that a new version has been shipped to The Library for distribution to anyone interested. This version of Editor is known as version 3.02K, and has lots of advantages over the first of my releases, Editor 3.00K.

I have can now announce with reasonable confidence that Editor 3.02K runs on all QL com-

patible platforms. It has been tested on QLs with Trump Cards, Gold Cards, Super Gold Cards (with and without Aurora) on Atari ST and TT running SMSQ/E, on QXL and Q-40, on QPC and QPC2 (on screens up to 1280x1024 pixels) and it has even been tried and found to work on an old CST Thor. No problems reported to date, and though there can be no rock-solid guarantee it is highly likely that your system will run the new version.

I have updated Editor's screen handling and both Editor and its configuration program can now use the full screen on systems with up to 4096x3072 pixels - that should allow plenty of room for growth... I have also made it possible to enter or configure the help, character set and boot command file-names to the full 36 characters allowed by the operating

Q Branch



PROGRAMMING

QD 98	£ 53.00
QD + QBasic	£ 69.00
QD + Qliberator + QBasic	£ 110.00
Qliberator	£ 50.00
Master Spy v 3.3	£ 30.00
QPTR	£ 37.00
Easyptr pt 1 & 2 (together)	£ 33.00
Easyptr pt 3 (C library)	£ 16.00
QMake	£ 18.00
QMon / JMon	£ 22.00
Basic Linker	£ 22.00
DISA 3	£ 37.00
QMenu	£ 16.00

Text 87

£ 79.00

Typset 94	£ 29.00
Fountext 94	£ 39.00
2488 drivers	£ 29.00
Epson ESC/P2 drivers	£ 26.00

Text 87 is the only QDOS / SMSQ wordprocessor capable of handling the full screen on the Aurora / QXL / QPC systems. New drivers are currently being written.

UTILITIES

FiFi 2	£ 22.00
QSup	£ 32.00
QSpread v2.04	£ 66.00
Cueshell 2	£ 30.00
Qload / Qref	£ 15.00
Disk Mate 5	£ 16.50
QPAC 1	£ 20.00
QPAC 2	£ 40.00
QTYP 2	£ 30.00
QLQ	£ 32.00

'Just Words' by Geoff Wicks

THESAURUS, STYLE CHECK, SOLVITPLUS 3

£ 10.00 ea / ANY 2 PROGRAMS £ 18.00 / ALL 3 PROGRAMS £ 25.00
(Includes Pointer and non-pointer driven versions)

(P.E. versions need Hot_text, WMAN and PTR_GEN or SMSQ/E to run)

Upgrades from previous versions £ 2.50 + S.A.E. New Manuals £ 1.50

QL2PC - New program !!

Convert text files from QL to PC formats and much more !

Only £ 10.00 Now with HTML support !

Spelling Crib : PD program £ 1.50 + SAE
or Free if you buy all three programs

Q Branch is on the move ! From the end of this month we will be returning our HQ to Portslade and our old telephone number. The Bank Volt, which has been our home for the last 2 years will be closing down. This is a bit sad for me because every now and then we have had customers who have come especially to buy QL software but I will be taking a full time job with a PC parts supplier and Q Branch will effectively become my evening job. This does not lessen my commitment to it and we are not giving up. If you need to call us please do so between 6pm and 8pm in the evening. QXL colours. The colour drivers for the QXL have now been released. These are the first colour drivers for any system other than the Q 40 and mark the start of the process of integrating the new Graphic Interface into the other systems. We do still have a few second user QXLs if you want to try these out.

Dilwyn s Fontpack is finally available. 8 Disks and hundreds of pages of printout. Just what you need now ProWesS is free !

Finally the Q 40 colour drivers are at v 2.98 and the support disks now have Dave Westbury s Photon Program for viewing JPEG files included.

We are currently out of stock of the SuperBasic Reference Manual
Place your order now to get one as soon as it is reprinted

The SBASIC / SuperBASIC Reference Manual

The complete definitive guide to BASIC programming in QDOS / SMSQ including three disks of PD toolkits, example procedures and an electronic index.

compiled by Rich Mellor, Franz Hermann and Peter Jaeger

Over 500
pages !

£ 40.00

+ postage

Q Branch

Feeling out on a limb?
Reach out for Q Branch.
Suppliers of Quality QDOS/SMSQ products
Hardware and Software.

Tel +44 (0) 1273-386030 fax +44 (0) 1273-381577

email qbranch@qbranch.demon.co.uk web : http://www.qbranch.demon.co.uk

Q Branch

NEW ADDRESS 20 LOCKS HILL
PORTSLADE
E. SUSSEX. BN41 2LB. UK.
NEW NUMBERS
Few

ProWesS

ProWesS (now free !)	£ 1.60
DATAdesign	£ 24.00
Fontutils	£ 30.00
File Search	£ 12.00
PFlist	£ 12.00
Dilwyn's Fontpack	£ Call
LINEdesign v 2.16	£ 24.00
PWfile	£ 18.00

Paragraph

The ProWesS word processor

Demo version £ 1.50 + postage
Full Registered version £ 18.00



Version 2.02 available now !

SMSQ/E

Gold Card / Atari / QXL Version
£ 76.00

Various Atari versions : call for details
THE QXL COLOUR DRIVERS ARE AVAILABLE NOW !
RETURN OLD MASTER DISK FOR UPGRADE - £ 35.00

QPC 2 is here !

£ 90.00 (£ 70.00 SMSQ/E Owners)

Upgrades from QPC 1 £ 30.00
(return master disk)

Special offer !

Get Cueshell for only £15.00 with any copy of SMSQ/E

Hardware

We have a small stock of second user items. Auroras / Qubides / Gold Cards / Qplanes / superHermes etc. call us to get details of the items available. These are going fast so call soon.

Super Gold Cards	£ 125.00
QXL	£ Call
Recycled superHermes	£ 80.00 *
Recycled Gold Card	£ 60.00 *
Recycled Aurora	£ 75.00 *
Aurora	£ 90.00
Qubide	£ 55.00
Qplane	£ 25.00
Aurora cables	£ 3.00
Aurora rom adaptor	£ 3.00
'Arfa Braquet'	£ 8.00
'Son of Braquet'	£ 18.00
The 'Braquet'	£ 16.00
MC plate	£ 6.50

* when available.

14" and 15" monitors for the Aurora - Call.

Q Branch Programs

The Knight Safe 3	NEW VERSION!	£ 35.00
upgrades from previous versions		£ 5.00
Q - Route v1.08C		£ 25.00
Route finding programme		
Q - Count	NEW VERSION!	£ 25.00
Pointer driven home accounting		
The Fractal Collection !		£ 35.00
Stunning fractals on your system !		
Dilwyn's Fontpack		
Eight disks and hundreds of printed pages of fonts for ProWesS systems. Most of these have been released as public domain fontas but we have included full printouts to make it easier to use.		
		£ 15.00



There are so many extras available for the Q 40 that it is hard to list them here. If you are interested in one of these boards than contact us and we will supply the details. the basic prices are shown here.

Q 40 board with 16Mb RAM & I/O card	£ 330.00
+ SMSQ/E	£ 30.00
Extra 16 Mb RAM	£ 30.00
Tower Case	£ 45.00
All prices do not include shipping.	



We can accept payment by VISA, Mastercard and Switch. You can also pay by Eurocheques made out in Sterling or a Sterling cheque drawn on a UK Bank. Prices include Post and Packing in Europe.



system so hard disk users no longer have trouble getting Editor to look in a subdirectory for files.

Editor 3.02K also has no known bugs at the time of writing this text, all the bugs I knew of in 2.05, 2.10, 3.00K or 3.01K I have fixed. There have been a number of bugs in Editor known to me for years and there were more listed in the REMark lines at the start of the version 2.10 source. I set about finding and fixing these, though in one update I discovered I had introduced a nasty bug myself - well that's what beta test volunteers are for. Anyway, the result is no bugs left on my list. When I shipped 3.01K to Phil Jordan for inclusion in The Library I confidently wrote in the text file included that there were no known bugs, which was true. The very next day a letter from George Gwilt reported two bugs and another letter from somebody else reported one of the same bugs; a phone call two or three days later reported a bug in the configurator; that's programming for you. OK, in a program the size and complexity of Editor there must still be some bugs, so if you find any of them let me know quickly and I'll have a go at fixing those too. Editor sorting is faster and more flexible, screen updates are slightly quicker and there are a number of new commands. One of these is a command added by Chas Dillon to add up a column of numbers, enhanced by me so that if adding integers it will give full 10 digit accuracy if the total is within the 32-bit signed integer range. You can also enter characters by decimal or hexadecimal code and alter the Editor default device at runtime, previously the default device was only altered by configuring the program.

One change that was small in terms of programming effort but is extremely useful has added another filetype to Editor; "Structured" files. This is for editing 'C' source, Assembly language, Perl or any other programming language that doesn't require line numbers at the start of each line (SuperBASIC and SBASIC files should be edited as "Unformatted" files, as before). There aren't many changes to the behaviour of Editor for this filetype, but they do make a significant difference when Editing my own Assembly language files and I hope others find them useful.

To keep pace with Editor I have changed the configurator and the EdtPrt (printer driver) program to use high resolution screens, and changed the keyboard handling slightly in EdtPrt to make it more Editor-like. The old configuration program could not configure Editor in a subdirectory, the new version can. I don't intend to make further changes to EdtPrt as I don't use it, and I will leave it to somebody who is more familiar with the program to do more work. I wrote Q-Page (available from the QUANTA library) because I didn't like EdtPrt, so I use Q-Page. Q-Page is smaller, much faster and better suited to my own needs, though it does not support the masses of command features that EdtPrt does.

If you used Editor a lot but stopped after you obtained an Aurora graphics card or a QXL or whatever, you may like to see if the updated version of your old friend suits you. I have sent it to The Library, it is FreeWare and if you want the source code you can obtain that too. Please don't send requests or orders direct to me I can't handle the volume of

mail that would result from this, Phil Jordan is set up for it, I'm not. Order details for The Library will be elsewhere in QL Today.

For a while I intend to accept bug reports and suggestions and to continue updating Editor. As the source code is available other keen QL programmers may fix more bugs or enhance the program further; if you do this please, please send me your source listing and a clear explanation so I can amend my own version of Editor and pass the benefits on to other users. All amendments will be properly credited. As well as bug reports if you have suggestions for further changes please pass them on. I certainly don't promise to implement suggested new commands or features but they will be considered.

If you want to comment, to make suggestions or to report further bugs please write or 'phone me directly:

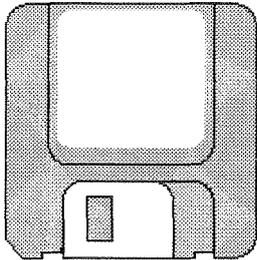
Mark Knight,
304, Portobello Road,
Notting Hill,
LONDON,
W10 5TA.

Or 'phone: (020) 8932 6987
(Note new telephone code and number as a result of changes to London code. Do NOT call before 9:00 a.m. or after 9:30 p.m.)

Mark Knight April 2000.

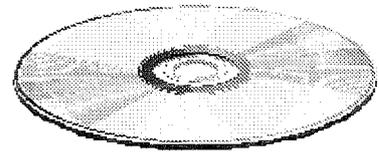
THE LIBRARY

QL SOFTWARE FROM



THE PAST, PRESENT

AND INTO THE FUTURE.



NOW INCLUDING THE

QUBBESOFT LIBRARY

Formerly from Ron Dunnett

Available on DD or HD Disks, 100 or 250 Zip and shortly on CD

P. Jordan 42 Hawthorn Crescent, Cosham, Hants, England. PO6 2TP Tel. (44) 023-092370574

Q40 and Perfection

by George Gwilt

As SMSQE for the Q40 develops, some strange things happen when programs are run, especially when the extended colours are brought into play, by the command DISP_SIZE.

I have recently moved onto SMSQE v2.97, which comes more firmly to grips with various ways of using the 65536 colours available. The most recent problem I encountered was the curious fact that Perfection refused to print anything anywhere. This annoyed me since I rely heavily on Perfection's printer drivers to produce output just as I want it, even when the source has come from somewhere else - QD, The Editor or even from an alien program such as Microsoft Word.

My annoyance was so great that I actually set about finding what was wrong. As is often the case, when I had found the error, I was surprised that the program worked in the first place. It now seems odd to me that Perfection ever prints anything anywhere! Once the error is found a way to correct it is usually obvious and I now once again have a viable Perfection for my Q40. The SuperBASIC program below might be of

value to other users of Perfection v 6.10. It requires the program to be altered to be called "perfection" and to be in ram1_, though an alteration to the definition of "dev\$" can allow it to be in any other directory. The file itself is changed, so, if you want the original one to be preserved, you'd better save it somewhere first. The program also assumes that TK2 is loaded.

For those interested, and also for those who might be able to update Perfection from its source code, I report that the error stemmed from the (perhaps unwitting) assumption that the high word of register D1 is left zero after a call to SDCLEAR, even though the manual distinctly says that the final contents of the register are "undefined".

Listing of SuperBASIC Program to Correct Perfection v 6.10

```
100 REMark *****
110 REMark * To correct Perfection *
120 REMark *           V 6.10           *
130 REMark *****
140 :
150 dev$='ram1_':REMark ** alter this as
    needed **
160 ch%=FOPEN(dev$&'perfection')
170 IF ch%=0:Wrong 1:STOP
180 IF NOT Perfection610:Wrong 2:STOP
```

```

190 IF Alter_Ok
200 a$=dev$&'Perfection altered'
210 ELSE
220 Wrong 2
230 END IF
240 PRINT#0,a$:CLOSE#ch%
250 STOP
260 :
270 DATA 54,46,49,48:      REMark *    "6.10"
280 DATA 96,0,91,14:      REMark *    BRA      end
290 DATA 112,32:          REMark *    MOVEQ   #32,DO
300 DATA 78,67:           REMark *    TRAP    #3
310 DATA 114,0:           REMark *    MOVEQ   #0,D1
320 DATA 96,0,164,236:    REMark *    BRA      back
330 :
340 DEFine FuNction Perfection610
350 LOCal b%,c%,x
360 IF FLEN(#ch%)<>82604:RETurn 0
370 RESTORE 270
380 FOR x=25 TO 28
390 BGET#ch%\x,b%
400 READ c%
410 IF c%<>b%:RETurn 0
420 END FOR x
430 RETurn 1
440 END DEFine
450 :
460 DEFine FuNction Alter_Ok
470 LOCal x,b%,c%
480 RESTORE 290
490 FOR x=59292 TO 59295
500 BGET#ch%\x,b%
510 READ c%
520 IF c%<>b%:RETurn 0
530 END FOR x
540 RESTORE 280
550 Push 59292,4
560 Push 82604,10
570 RETurn 1
580 END DEFine
590 :
600 DEFine PROCedure Push(ps,num%)
610 LOCal x,b%
620 FOR x=ps TO ps+num%-1:READ b%:BPUT#ch%\x,b%
630 END DEFine
640 :
650 DEFine PROCedure Wrong(k)
660 INK#0,2:STRIP#0,7:CSIZE#0,2,1
670 SElect ON k
680 =1:PRINT#0,"I can't open "&dev$&"perfection"
690 =2:PRINT#0,"Perfection either altered or not v6.10":CLOSE#ch%
700 END SElect
710 INK#0,4:PAPER#0,0:CSIZE#0,0,0
720 END DEFine

```

QIMI Hints

by Dave Westbury

This information is also available on Dilwyn Jones' Web site, along with a circuit diagram of the QIMI interface, saved as a QL PIC file.

<http://www.soft.net.uk/dj/qldocs/qldocs.html>

QIMI interrupts are enabled by read or write to address 114622 decimal (01BFBE in hex).

Mouse buttons are read from 114588 (\$01BF9C)
bit 4 = right button
bit 5 = left button
0=button on

Mouse movement is read from 114620 (\$01BFBC)
bit 0 = up/down direction
bit 2 = right/left movement
bit 4 = right/left direction
bit 5 = up/down movement
0=left and down, 1 = right and up

You and Your Software - Just good Friends?

Part 8 - What the User doesn't see.
Geoff Wicks

One line of code in my program Solvit-Plus took over 30 hours to write. If this statement conjures up a vision of me sitting snug in my anorak nerdishly tapping on my keyboard into the early hours, then you would be wrong. For most of the time I was eating, drinking and making merry and the poor QL was doing all the hard work.

The line of code it produced is not particularly spectacular:

```
strg$="eianrtsoldcugmphbfvkzywxqj"
```

The QL was going through long lists of words for the main European languages and working out the relative frequency in which each letter appears - "e" is the most common, "j" is the least common. In some searches Solvit-Plus looks through the search word to find the least common letter, and then examines only words containing that letter. This can more than treble the speed of some searches, but the user is not

aware of it, because it is used only in the slowest searches.

Most programmers can tell a similar story. They put a lot of effort and original thinking into part of the program, but the user remains blissfully unaware of the time and effort involved.

It can also work the other way round. Your program may work, but the code and layout may be so bad that it is just as well the user remains unaware of its content. A programmer who writes sloppy, spaghetti type code will soon find it backfires on him. A bad dream will start at the first bug report, and turn into a full scale nightmare when the program needs upgrading. This article is not about your programs being friendly to the user, but about your programs being friendly to you.

When I was a boy people referred to the three R's in education - reading, writing and arithmetic. This article is about the three R's of a well laid out basic program - REMarks, readability and reference.

REMark statements should liberally pepper your programs. These help you to remember what each section of the program does. You could, for example, put a REMark statement at the start of

HIT AND MISS?

Writing QL software is often a tricky business, because it is not always clear what users want. We try to make it less of a hit and miss affair, by listening to customers and non-customers. Hence a quick upgrade of our QL-2-PC Transfer program to generate HTML (Hit and Miss Language to the cynics!). We have also added subscript, superscript and Quill italics.

Sometimes we don't get it quite right. The market for QL-2-PC transfer was slightly different from what we expected, so just a little warning. To run this program you will need to possess the pointer environment files (ptr_gen and wman). You will also need a disk utility such as Xover, DISCover or QLIBM_exe.

We don't want the transfer of files to be a hit or miss affair for you, so you can always contact Just Words! for help even if you don't want to buy the program. Be warned, however, we shall still tempt you with our demo disk!

QL-2-PC TRANSFER - STILL ONLY A TENNER!

Geoff Wicks, 28 Ravensdale, Basildon, Essex, SS16 5HU, United Kingdom.

Tel: +44 (0)1268 - 281826 : email: geoffwicks@hotmail.com

Web: <http://members.tripod.co.uk/geoffwicks/justwords.htm>

every procedure describing its purpose and use. You could place them at strategic places in a program to remind you about the meaning and purpose of a variable. They are also helpful in SElect ON - END SElect loops to remind you what each item in the loop does.

In my Style-Check program the processing of a text file is done in SElect on - END SElect loops. You will find that it contains lines like:

```
= 65 to 90 : REMark Upper case letters
= 97 to 122 : REMark Lower case letters
= 44, 58, 59 : REMark Punctuation marks.
```

REMark statements are particularly helpful in pointer programs where you have to use numbers for your menu items:

```
= -4 : REMark L - Load document
= -5 : REMark I - Information screens
= -6 : REMark S - Save file
```

The **READABILITY** of most documents is enhanced by a simple rule, "Use plenty of white space", and this also applies to a basic listing. How do we create white space in basic?

One of the most useful lines in basic does nothing at all. It is simply the line number followed by a colon:
1000 :

If you place this line at the end of each procedure or function, you will quickly see where one procedure or function ends and another begins. You can go a stage further and use a colon line to separate different parts within a procedure. The menu loop is on obvious place. Separate the code for each menu item and you will find your menu loop easier to follow.

Another important use of white space is in indentation. Make sure your code is indented in all types of loops (e.g. SElect On, REPEAT, FOR n =):
1000 DEFine PROCedure countdown
1010 REPEAT loop
1020 If count=3 : EXIT loop
1030 count = count - 1
1040 END REPEAT loop
1050 END DEFine

If you indent your code, you will find it is much easier to follow, particularly if you are fond of using nested loops.

The good news is that you do not have to indent the code yourself as there are many programs to do it for you. These include Mark Knight's File

Utilities on QUANTA library disc UG13 and a listing by Dilwyn Jones published in QL Today Volume 3 Issue 2 page 54.

The final R is **REFERENCE**. A long basic program will have numerous variables, procedures and functions, and it is easy to lose sight of the structure of the program, particularly if you have not looked at it for sometime. To understand your program you need more than a simple listing.

You could, of course, manually make a note of every variable and the program's structure, but how much easier it would be if this could be done automatically.

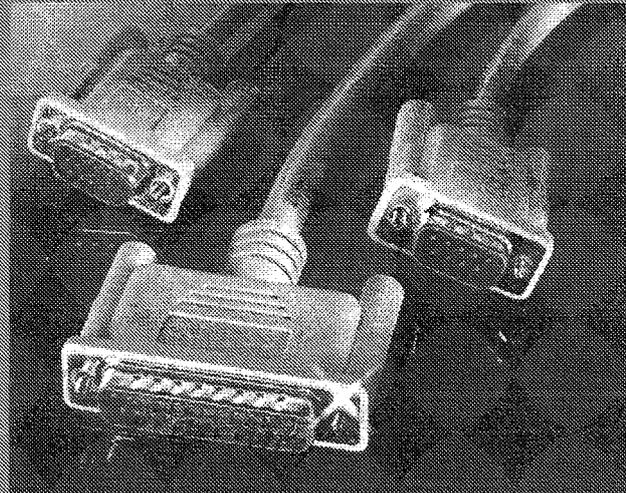
Here is more good news. There is a program that does this. QREF analyses and prints the details of a basic program. It first of all lists every procedure giving the line number where it starts and a list of other procedures called from within it. Next it prints out the structure of the program giving the level at which each procedure is used. Then it prints out every variable, making a note of its type and every line in which it is used with the lines in which it is assigned highlighted. It also prints out every command and function used and the number of times it is used. Finally it gives a warning of items defined but not used and of estimated data space requirements.

Dilwyn Jones introduced this program to me about five years ago when I first started to write commercial software and I have found it a valuable aid every since. It saves much time when someone has reported a bug or I am upgrading a program.

What Qref unfortunately does not do is help you with pointer programming. If you use Easyptr you will know each menu item, and each information window, of which there are often many has an identification number and parameters. At the moment these have to be noted manually. It would be a useful addition to Easyptr if these could be printed out.

Next time: Professional howlers.

This series is near its end. The next article will look at some professional "in-house" software, and the lessons that can be learnt from this. The final article will look at the QL commercial market. Meanwhile if there is a topic you think I should have covered, but have not done so, please let me know.



SERNET SPECIAL

USERS TELL US ABOUT THEIR EXPERIENCES WITH SERNET, HOW THEY GOT IT TO WORK AND WHAT NEEDS TO BE CONSIDERED. MORE TIPS ON SERNET? TELL US, PLEASE!

Q40 - Q40 Sernetting

Derek Stewart

Connection of Q40 via a standard Maplin null modem cable, using Sernet v2.2 Gives faster networking than the QL network at the maximum I/O card baud rate in the Q40. Which is 115200 baud = 11.Kb/sec approx.

To install Sernet v2.2 on a machine for serial networking, configure Sernet v2.2 to access the required serial port in question.

Connect the null modem cable to the specified serial ports of the machines in question. I usually configure 4 versions for each of the Q40's serial port, naming them as SERNET_SER1_REXT,... etc

The commands to connect two Q40 or other machines are as follows, please bear in mind that the baud rate must be the same on each machine. So the following table can be used for the machines I have in use:

		BAUD RATE
Q40	Q40	115200
Q40	QXL I	57600
Q40	SuperGoldCard Aurora (Ser1/2)	19200
Q40	Atari ST Mode 8 Emulator	19200
Q40	Trump Card QL	9600

The Trump card requires some serial port extensions to use sernet, it is SMSQ/E specific, but the transfer rate is slow. The QXL has a very low data transfer rate, but will connect at 57600 baud.

I have not got the Superhermes SER3 port connected to Sernet. Probably because the baud rate of the SER3 port is set in the command string and not by the BAUD command.

To Serial network two Q40 machines at 115200 baud, the following sets were used:

Q40 station 1 (server)

```
BAUD 2,115200 : REMark Set SER2 to 115200
                baud
LRESPR "SERNET_SER2_REXT" : REMarkSernet
                configured for SER2
SNET 1 : REMark Set Sernet station 1
SERNET : REMark Start Sernet Server
```

Q40 station 2

```
BAUD 1,115200 : REMark Set SER1 to 115200
                baud
LRESPR "SERNET_SER1_REXT" : REMarkSernet
                configured for SER1
SNET 2 : REMaark Set Sernet station 2
```

Then on station 2, use DIR S1_WIN1_ will give a directory of the station 1, WIN1_

All SBASIC commands work as usual using the S1 prefix like the N1 for the QL LAN.

Using QPAC2 Files thing, transfers files very quickly from the Sernet Server Q40, as it must use the extra ram to cache the file transfer.

The Q40 at present runs the Holborn View BBS connected by sernet to a Atari Mode 8 Emulator, the connection rate is only 19200 baud or 1.92 Kb/sec. And uses a standard null modem cable from Maplins.

SERNET

Dilwyn Jones

SERNET was developed from Phil Borman's MidiNet software. Bernd Reinhardt modified the software to use the serial ports instead of the MIDI ports. An article describing the original SERNET appeared in QL Today Volume 1 Issue 2, page 36.

Several people have reported problems getting SERNET to work. I was one of those initially until I upgraded to a newer version and found the correct wiring for the machines concerned.

So far I have not yet made up a cable to connect a standard QL to anything, but since PCs and Auroras are likely to use similar serial ports, the following cable diagram should help you get simple 2 machine setups to work.

Cable wiring is pretty critical. For two machine setups, you need to cross connect DSR and DTR, CTS and RTS, TxD and RxD between the serial ports, and also connect up the Ground pins on both machines, a total of 7 connections. The diagram in Fig. 1 shows how to connect up serial ports which use the 9 and 25 pin D type connectors, wired up as PC style serial ports. Buying a 'Null Modem' cable should work for systems with PC style serial ports. Some serial Interlink or Laplink cables may also be suitably wired for use with SERNET.

SERNET requires SMSQ/E to work, since standard QDOS cannot handle the independent input and output serial channels (SRX1 or SRX2 for receive on serial ports 1 or 2 and STX1 or STX2 for transmit on serial ports 1 or 2).

DB9	DB25	SIGNALS	DB25	DB9
2	3	RxD-TxD	2	3
3	2	TxD-RxD	3	2
4	20	DTR-DSR	6	6
5	7	GND-GND	7	5
6	6	DSR-DTR	20	4
7	4	RTS-CTS	5	8
8	5	CTS-RTS	4	7

Figure 1 - Cable diagram for standard 2 computer setup

It is absolutely vital that cabling is correct. It is unfortunately possible to get it to seem to work when you get the handshaking wrong, but a typical symptom would be that communication would work with very low baud rates or short transmissions, but errors would creep in during longer transfers.

To make a cable to connect three or more machines, you have to wire up a special cable where output signals from one machine are wired to the input of the next and so on, to form a complete circle or ring from first to last computer.

Now that you have made up a cable, you need to configure the SERNET code file supplied with SMSQ/E. Look for a file called SERNET_REXT (or sometimes SERNET_RXT) and then use MenuConfig level 2 to set it up to your requirements.

Make sure that you avoid v2.24 of SERNET, this had some problems which meant I wasted a lot of time trying to get it to work without knowing it wouldn't work at all. When I was sent v2.25, the problems went away and I was able to happily SERNET using that version.

MenuConfig offers two configuration items - select the SERNET v2.xx option first. This will ask you to make a few entries:

Name for Serial Receive - enter SRX1hd or SRX2hd

Name for Serial Transmit - enter STX1hd or STX2hd

Server timeout - enter 0 here unless you know better! (1 or 2 can be used with slow computers, although these settings affect performance).

If your hardware does not support hardware handshaking, you can replace the 'hd' parameters with 'xd' for xon/off software handshaking instead. Hardware handshaking is preferable if supported.

The other configuration item allows you to specify SERNET/Modem. This asks if you want sernet to work via modem. Select the NO option for our cable connection. Now

select 'OK' to save the reconfigured file.

Before you load SERNET, set the baud rates for both computers. 9,600 baud is a good value to start with, although this will prove to be slow. When you get it working, experiment with higher baud rates, whatever your hardware is capable of.

Install SERNET_REXT with the LRESPR command -

```
LRESPR FLP1_SERNET_REXT  
or whatever.
```

Activate the sernet job with the command
SERNET

The sernet station numbers (similar to QL network station number) is set with the command SNET. To set your computer to be station 1, use

```
SNET 1
```

To set it to be station 2, use

```
SNET 2
```

and so on. They seem to default to station number 1 - both machines on a 2 machine system can both be left as station number 1.

You are now ready to test the connection. From BASIC, try a command such as

```
DIR s1_FLP1
```

If you get the list of files, all is well. Try copying or spooling a file over the network to the screen of the other computer:

```
COPY FLP1_FILENAME TO s1_SCR
```

The sernet device name defaults to S, so station 1 is device S1_ and station 2 is S2_

This can be changed - to change the sernet device name to R for example:

```
SNET_USE R
```

If another program closes the serial ports used by sernet for some reason, you can reopen them with the

```
SNET_ROPEN  
command.
```

A week-end with SERNET

Davide Santachiara

Premise

For many year I've been using the QXL card on my PC because it allowed me to exchange data easily with my Aurora machine via the network port. Another good point of the QXL was that, unlike QPC1, it is able to run under Windows (without passing through DOS) and this was quite useful for me even if the i/o is much worse than QPC. Now that QPC2 exists and it is well supported (unlike QXL's SMSQ/E) there is almost no need to use the QXL unless you have a slow PC. The only problem remained was how to exchange data between my two QL systems because QPC, being a software emulator, has no network port like the QXL. This is why last week I decided to put my hands on SERnet in an attempt to link my QPC2 and Aurora systems.

SERnet is a useful addition to SMSQ/E which allows to link two machines in a network via the serial port. Please remember that SERnet will only work if you have SMSQ/E on your systems. In fact it uses a feature of SMSQ/E which, as far as I know, is not present in any other QDOS compatible o/s: independent rx/tx channels for the serial port.

My Systems hw/sw

QL1: Aurora - Super Gold Card - SuperHermes - SMSQ/E 2.95

QL2: QPC2 mounted on a PC with AMD K2-400 - SMSQ/E 2.95

I found the sernet_rext file on my QPC1 disk. The version number was v2.22 (Dilwyn Jones kindly sent my 2.25 but I've had not time to test it). SERnet is a resident file that has to be LRESPRed on your QL system. Before doing this it has to be configured with JMS Menu-config in order to set the correct serial port channels. I have also seen that SERnet allows to set up a network by connecting two machines via a Modem which could be very useful if you need to connect two systems which are not in the same place. However I've not tested this (undocumented?) feature.

Choosing a serial port and a cable

On the Aurora side I wanted to use SuperHermes ser3 but it seems that sH does not support STX2hd or SRX2hd channels (independent serial tx/rx channels required by SERnet). SuperHermes Serial 3 is much faster than the standard QL serial ports in fact I bought it as soon as it came out to connect the QL system to a modem for (Ergon) BBS use. Furthermore, from previous tests I made with the Aurora serial ports (ser1/2), I've found that hardware handshake does not work properly. But due to the inability to use superHermes serial 3 the only possibility was to try Aurora ser2 at 19200 connected with the PC serial port (COM1). So I went in that direction.

For the cable I've bought a null serial cable (9/25 female connectors at both end) in a commercial centre for around 20 euro. I believe this is a standard PC null serial cable which can be found everywhere.

First tests

My first test was done in order to check that the cable was OK and that communication was established between the two serial ports. So I connected the null serial cable and I made the first attempts by using QTPI both on Aurora and QPC systems, setting the baud rate to 19200. To my surprise, typing some characters on one QTPI side, I immediately saw the same characters on the other side and vice-versa. But as soon as I tried a ZMODEM transfer things did not work at all. This was not a surprise as I was never able to make a reliable connection between Aurora ser1/2 and a modem. Just for security I connected superHermes serial3 to the same PC port to check that the cable was OK, and in fact the link worked perfectly even with a ZMODEM transfer.

SERnet test

Even if the first test was not 100% OK, I decided to see what happened with SERnet. I configured via menuconfig the serial ports in the SERnet_rext file:

SRX1hd STX1hd on QPC2 for rx/tx (this depends to the serial port you want to use)
SRX2hd STX2hd on Aurora for rx/tx (same as before) server timeout **0** sernet via modem **NO**

I added the following SuperBasic lines to my BOOT file and I rebooted my system:

```
LRESPR sernet_rext SNET 1 :REMark SNET 2  
on the other machine (this command assign  
the SERnet number to the machine - accord-  
ing to Dilwyn you can also set SNET 1 on  
both machines)
```

```
SERNET
```

```
BAUD 19200 : REMark BAUD 1,19200 if you use  
ser1 BAUD 2,19200 for ser2.
```

**[I suggest you set the BAUD rate before you
LREPSR sernet! - Jochen]**

After some fine-tuning I was able to make the two Aurora machines talking in some way: i.e. DIR s1_win1_ on the SNET 2 machine basically worked reporting the directory listing of the other computer. Unfortunately after some tests I found that the communication was not 100% reliable. For instance sometimes the communication stopped and the machine remained locked for some times till a sernet communication error was returned in #0. The annoying thing is that while SERnet is waiting sending or receiving data the QL system (both QPC or Aurora) remains completely locked. Strangely I noted that while I was able to download files from the Aurora machine to the PC machine the vice-versa was not possible at all. Mmmmh!

So I started playing with the various serial settings (XON/XOFF instead of hardware handshaking etcetera) but I will not bore you with this matters and I'll go straight away to the solution (very simple, as usual): I lowered the baud rate at 9600 on both sides (BAUD 9600 instead of BAUD 19200) and things magically worked as expected. I suspect this again is related to the handshake of the Aurora serial ports: probably the speed of 9600 baud is so slow that even if handshake does not work properly the data transmission is done without problem (no characters are lost on the way).

Transferring data between the two QL systems at 960 cps (character per second) is not very comfortable even with QL short files. Just think that the QL/QXL network port throughput is roughly 4000 cps. Anyway for my need it's better than nothing - at least I can

backup the files of the BBS during the night in an automatic and easy way.

Conclusions

SERnet works. The critical aspects to make it running properly are mainly related to the serial ports and the cable - in this sense I was very lucky because I bought a commercial cable and in a week-end I was able to connect in a reliable way Aurora and QPC2. 9600 baud is a deadly slow speed, but better than nothing. Probably connecting two QPC2 systems or QPC2 with Q40 gives much better results (I think that 5760 cps throughput should be easily achievable).

Regarding the SERnet use it works like the QL/QXL network making transferring files between two computers a joke. The only minor thing missing is a command like Toolkit 2 NFS_USE which is quite useful as it allows to define a new device like lan1_ to access the remote machine (i.e. dir lan1_ is equivalent to dir n1_win1_ depending on how you configure NFS_USE).

Dilwyn Jones made a lot of tests with SERnet and was very helpful with me sending all the information he was able to find out in his tests. In particular he writes: "Sernet is an extremely annoying piece of software. When it works, it works VERY well. It took me a long time to get the cable right - it would sometimes work reliably for short transfers, but fail with long transfers because the handshaking was wrong".

I was a bit luckier than him but I can basically confirm his impressions: just remember that the software itself is simple to be configured and installed. The difficult part is having the right serial part and cable.

Davide Santachiara

GSM: +39 333 3440088

Email: ergon@geocities.com

Web page: <http://get.to/ergon>

Ergon BBS: 0522-300509 21:00 to 3:30

Fidonet 2:332/534

We always said: "it does work!"

Jochen Merz

First of all, I would like to thank Bernd Reinhard very much for maintaining and improving SERNET.

I am using SERNET since it exists to backup my BBS TT and my "main" working TT at 57600. Quite nice to have the same harddisk content on both machines.

I also used SERNET instead of MIDINET to backup my ATARI Stacy because SERNET was more reliable (handshake, which MIDI does not have).

When QPC appeared, I also used it to backup my PC Laptop from my TT at 57600 baud. No problem, it worked fine in every direction.

I never understood why some people said SERNET would not work, but from reading the experiences of SERNET users, it seems that it all comes down to the ports (it is public knowledge that the QL SER1/2 do not work reliably at 19200 baud) and the cables (not much I can do about cables: when I say "go and buy a null modem cable" and people go and buy a "laplink cable", then it is not really SERNET's fault.

I hope the previous article helped to understand what you need to do to use SERNET and to avoid potential problems.

The SERNET modem is not 100% perfected yet, but it should be working soon. A new version of SERNET which will work if installed more than once on the same system (to allow differently named networks on different serial ports is in the test phase too). You will get differently named nets and differently named BASIC control commands. At the moment, most commands will only modify the most recently LRESPred version of SERNET.

Suqcess

A Review by Tim Swenson

Suqcess is a Pointer Environment driven database program written By Wolfgang Uhlig. The documentation specifies that SMSQ/E is needed. I tried to run Suqcess on my JSU QL, but it locked up after some use. My version of the Pointer Environment may be a little old for Suqcess, or it really does not work without SMSQ/E.

What is a Database?

Before I go into the review, I want to first talk a bit about databases. A database is a collection of tables, each composed of fields and containing records. A field is a unit of data. It can be a first name, last name, address, number of products shipped, etc. Fields define what type of data is stored in the table. Records are the actual data itself.

A relational database is a collection of tables with like fields. This means that searches can be made across tables and include fields from both tables. Typical relational databases are dBase II, Oracle, Informix, and Sybase.

A true database (or database engine) is a program that only manipulates tables, meaning creating, searching, and deleting tables. A database engine typically has a text-only interface with a command line.

A database front-end is a program that takes a very plain and boring database engine and adds a graphical front-end to it, making it very easy to use.

Most databases sold on the market come with both a database engine and a database front-end. Some were just a database engine (dBase II) and

some were mostly database front-end with the database engine hidden (e.g. MS Access).

What is Suqcess?

Suqcess is really a database front-end to the DBAS database engine. D. Howell wrote DBAS as a database Toolkit that allows programmers to write database applications using S*Basic. It is a very useful utility, but it is just a database engine. Making a working database out of DBAS takes some programming skill and effort. It was not designed for the average user to use it to replace Archive.

Suqcess is the first application, that I am aware of, to make use of DBAS. Having looked at DBAS for my own programming needs, I'm glad to see that someone has made good use of DBAS.

The function of Suqcess is to provide a Pointer driven database for the average user to create and maintain a simple single-table database. Suqcess is not designed to handle more than one table at a time, so it can not handle a relational database.

Suqcess runs from a single window, with command icons on the top of the window and a big area to display records below. Suqcess displays a table similarly to a spreadsheet. Fields are columns across the window, and records are rows down the window. There are little scroll bars on each side of the table that allows for scrolling the table up, down, left, and right.

Documentation

Suqcess comes with a 31 page manual in roughly A5 format. It does a good job of covering all details of the program, albeit briefly. A tutorial for Suqcess it

is not. Having used a database before, the manual covered all of the bits that I needed to know about using the program. If you are new to databases, it may take a couple of readings to get the overall concept. A general book on databases from the local library might help in understanding databases.

The book has color examples of all of the graphics used in the program, including the different icons. I don't know if color printing was really needed, but its a nice touch.

Installation

Suqcess does require the Pointer Environment, QMENU (menu_rext) and the DBAS toolkit (DBAS_sys), and comes with all of these files. On my Q40 I already have the Pointer Environment and QMENU loaded, so I just had to add an LRESPR to my BOOT to load DBAS.

Suqcess is configured to run from the directory win1_suq... So, I did a "MAKE_DIR SUQ" and then copied all of the files off of floppy to this directory. If you want to run Suqcess from RomDisq or floppy, 'config' or 'menuconfig' can be used to change the default directory. Suqcess does have 2 files that it needs to read when it executes and these must reside in the default directory.

Running Suqcess

Once it was installed and DBAS was LRESPRed, I just executed Suqcess. Since I was in 1024x512 mode, Suqcess popped up a little larger than the normal QL screen. It does check existing screen size, and will change its display size to fit the screen.

Suqcess comes with an example database, which is perfect for testing out Suqcess and seeing how it looks.

Success displays a database very similar to how Qspread shows a spreadsheet. Each record is a row and fields are columns. Each side of the display area has the typical PE arrow bars that scrolls the data up, down, left, and right. Above the data area is what could be called the Toolbar; a row of small command icons. Some of the command icons are configured to be accessed as function keys. The key names are listed next to the icons. Success also displays which database is currently loaded, and how many of the records are being viewed.

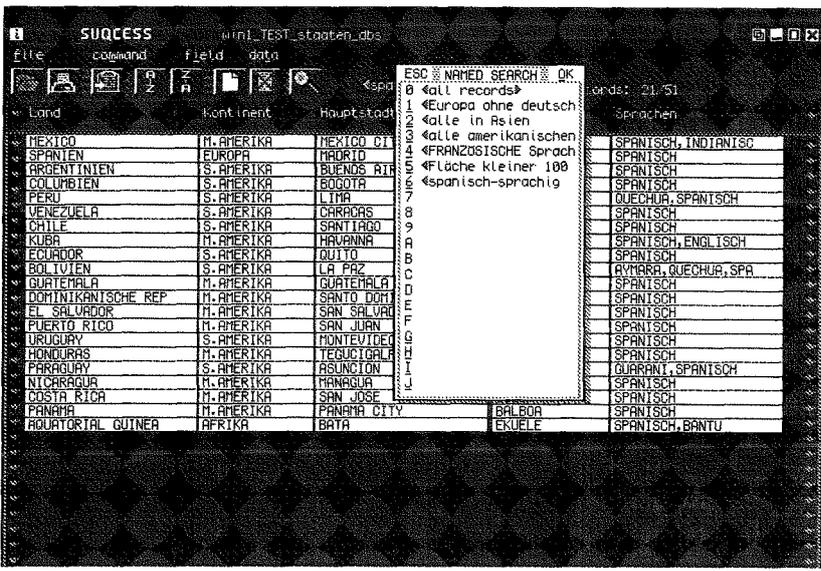
When doing searches, Success displays only those records that meet the search criteria. This is considered a "view" of the database. On the display Success shows something like (13/50), meaning that the current view is showing 13 of 50 records (50 being the total number of records in the database).

Searching a Database

While talking about searches, searches are done by point and click. Once the search menu is selected, a search is composed by clicking on the match criteria (equal, less than, greater than, etc), selecting the search field(s), entering the value in the criteria, and then execute. The result is displayed quickly. Searches are remembered and can easily be executed from a list of previous searches. Once a View is created it can be printed, copied to Scrap, or saved to a file.

Creating a Table

Like the rest of Success, creating a table is just a matter of pointing and clicking. A table is created by selecting the field type, entering its name, and



entering how long it is. The field types are String, Integer, Long (date), and Floating. Once a table is created, it is given a name and saved. Very easy.

Once a table is created, modifying it in the future is simple. Fields can be easily added or deleted from tables.

Some databases require Key fields, meaning that the combination of these fields must be unique in all the records. This means that duplicate records are not possible. Success does not have this requirement and actually has a command that duplicates a record, making it easier to create a new record with minimal typing if it is similar to another record.

Importing and Exporting Databases

Success allows for both importing and exporting data to the database. Given that the main database used by QLers is Archive, Success is designed to import data in the Archive Export (_exp) format. This format is a variation of the Comma

Separated Variables format (CSV) and is a common format. If exporting from another database, only minor modifications are needed to convert a _csv format to an _exp format file. A

full explanation of the _exp format is found in the original QL documentation (see under

Exchanging Data). If you have an existing database in either Archive or Abacus, converting it to Success is very trivial.

Configuration

Success does have a menu for configuration (it includes the one item that can be changed via Config or MenuConfig). The configuration options are:

- Startup Directory
- Startup Window Size
- Default Printer
- Maximum Field Width
- Maximum Visible Records
- Date Function

Once done making configuration changes, when the menu is exited, the configuration changes take immediate effect. No need to restart the program.

Database Limitations

There are some limitations to Success that are inherited from using DBAS. This are:

- No more than 255 fields per table.
- Each record can be up to 32767 bytes.
- Each table may have up to 32767 records.

So beware of these limitations if you plan to create some large tables.

Success also edits databases directly on disk. This means that when a change is made to a database, the file on disk is updated to reflect the change. This has some advantages and disadvantages. The main advantage is that the database is less prone to loss due to system failure. Changes are flushed to disk and not cached in memory. The main disadvantage is that if doing a lot of changes, this makes a lot of writes to the disk. If using a floppy only system, this can bog down the database and affect performance. If you need faster updates, the database can always be copied to RAM disk, edited, and then copied back to disk. Of course, any system failure before copying

the file back to disk will cause the loss of any changes.

Overview

Because Success is very pointer driven, it is fairly easy to use. A power user might find it cumbersome using the mouse so much, but for the average user, this makes the program so good.

If you are using Ababus or Qspread to create some single table databases, then Success would fit your need. If you are using Archive for simple databases, then Success would fit also. If you need a programming interface for making an application or generating automated reports, then Success will not fit that need.

Success can be used as only a simple front-end, for quickly viewing and editing a database. If you need to generate reports or mailing labels, then a little programming with DBAS can be done to accomplish this. When I was using Oracle, I used a tool similar to Success to easily edit the database. Using pure SQL to create and edit records is too cumbersome and painful.

For the average QL user, Success is the database PE front-end. It fits well amongst other PE programs in how it's used and how it works. If you are transferring over to using all PE programs, Success can fill your need to handle single table databases.

CAUTION: HOTKEYS - Don't Burn Your Fingers, Part 4

David Denham

The Stuffer Buffer

Although it sounds like the name of somewhere to place your computer when it misbehaves, the real purpose of this is somewhat less painful than the name implies.

The Stuffer Buffer is designed as a temporary storage area to allow information to be passed between programs - a sort of mini-clipboard which can be accessed by just about any program, even older ones like Quill.

The basic principle is that the text to be transferred is STUFFED into a little buffer area of memory with a HOT_STUFF command and a special keypress ALT-SPACE is used by the receiving program to read in the text.

The keystroke ALT SPACE (hold down ALT, tap the SPACE bar, release ALT) "feeds" the text from the stuffer buffer into your program as though the text had been typed at the keyboard. In addition, if there was a previous Stuffer Buffer entry, using SHIFT ALT SPACE (hold down SHIFT and ALT then tap the SPACE bar) will recover that previous Stuffer Buffer entry.

To insert a text string into the Stuffer Buffer in the first place you need to use the HOT_STUFF command - note that unlike most hotkey system keywords it's a command so does not need ERT.

To pass the current time/date to a program, go to SuperBASIC and type in:

```
HOT_STUFF date$
```

then CTRL C to your word processor or editor and use the ALT SPACE keypress to transfer the date and time to that program, which will look like this:

```
2000 Jan 30 21:20:03
```

If you need to transfer data from a BASIC program, you could use a simple routine like this:

```
100 INPUT a$  
110 HOT_STUFF a$  
120 GO TO 100
```

Although I have never used the program, I believe Roy Wood uses the command to great effect in his HTML Machine program for those who wish to make Web pages by composing in the HTML language in an editor on the QL. HTML requires the use of special embedded commands enclosed in < and > symbols to mark up a page and insert formatting information such as Bold, Italics etc. All the commands are available from menus in his program, and when required they are added to the stuffer buffer and can be quickly transferred into the editor to avoid having to memorise the sometimes tortuous syntax of HTML, although it

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

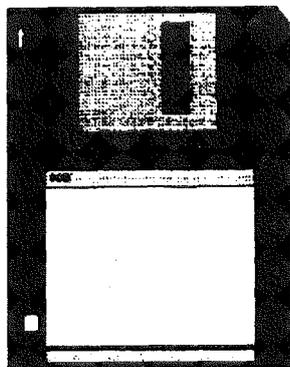
Subscription just £14 for UK members

Overseas subscription £17

Barclaycard: Visa: Access: Mastercard

*** Now in our SEVENTEENTH successful year ***

Further details from the Membership Secretary



Bill Newell
213 Manor Road
Benfleet
Essex
SS7 4JD
Tel. (01268) 754407



does mean a bit of toing and froing as you CTRL C between programs.

HOT_STUFF can be used in interpreted SuperBASIC, SBASIC and QLiberated BASIC programs.

Last Line Recall

When typing in repetitive commands from BASIC, or simply correcting a long entry in which you made a mistake, it is useful to be able to bring back a copy of the last line of text you typed in before you pressed ENTER for correction. Just hold down ALT and tap the ENTER key to achieve this. It is not 100 per cent reliable, but works in most cases.

Problems you may encounter with HOTKEYs

Q. ALTKEYs and HOTKEYs don't respond but have been defined.

A. The Hotkey job is not running. Check this by going to SuperBASIC and entering the command JOBS to see if a job called HOTKEY is listed. If not, enter the command HOT_GO

Q. ALT ENTER is not working at all

A. Same reason as above, but bear in mind that ALT ENTER is not infallible - it cannot bring back every line entered reliably.

Q. RESPR or LRESPR does not work - it keeps saying "Not Complete" when I try to use these commands to install some BASIC extensions.

A. With the JOBS command, check to see what jobs are running. Quit from all programs except BASIC, RESPR should then work. If the hotkey job is listed, you can stop this with either the HOT_STOP command or with RJOB "hotkey". After you have finished using RESPR, you should restart the hotkey job with HOT_GO

Q. I try to define a hotkey but it reports an error "In Use"

A. That key has already been defined as an ALTKEY or HOTKEY. Get rid of the old definition with a HOT_REMV

Q. Can I mix ALTKEYs and HOTKEYs?

A. Yes, but you can't put both on the same key.

Q. Can function keys be defined as hotkeys?

A. Yes, although it is not always considered a wise thing to do. You need to know the key CODEs of the five function keys.

F1 is CHR\$(232)

F2 is CHR\$(236)

F3 is CHR\$(240)

F4 is CHR\$(244)

F5 is CHR\$(248)

So you can define ALT F1 etc if really required. Add on keyboards which have more than 5 function keys usually define F6 to F10 as equivalent to SHIFT F1 to SHIFT F5 (so F10 is worth remembering on such keyboards as a shortcut to redraw the display in Quill, for example, and F9 toggles between INSERT and OVERWRITE modes in Quill), so just look up the codes for these keypresses in the user guide.

Reference Chart

The following chart lists the hotkey keyword and whether they are procedures (P) or functions (F)

ERT	P	Error Report - handle value returned by hotkey functions
EXEP	P	Like EXEC but handles Things etc.
HOT_CHP	F	Install program in common heap
HOT_CHP1	F	Start program or if already running PICK+WAKE
HOT_CMD	F	Send command to SuperBASIC
HOT_DO	P	Perform action associated with specified key
HOT_GO	P	Start hotkey job (enable hotkeys)
HOT_KEY	F	Assign text to a key
HOT_LIST	P	List hotkey definitions
HOT_LOAD	F	Load program from disk
HOT_LOAD1	F	Load program, or if already running PICK+WAKE
HOT_NAMES\$	F	Return name of program associated with this key
HOT_OFF	F	Temporarily disable a hotkey
HOT_PICK	F	Pick a running job
HOT_REMV	F	Force remove a key definition
HOT_RES	F	Place program in RESPR area for fast starting
HOT_RES1	F	Make program resident, or if running PICK+WAKE
HOT_SET	F	Restore hotkey after a HOT_OFF
HOT_STOP	P	Stop the hotkey job
HOT_STUFF	P	Place a text string into Stuffer Buffer
HOT_THING	F	Start executable Thing
HOT_TYPE	F	Type number of a hotkey definition
HOT_WAKE	F	Pick a job and send a WAKE event signal

Options for starting awkward programs:

P protect memory, usually used for Psion programs

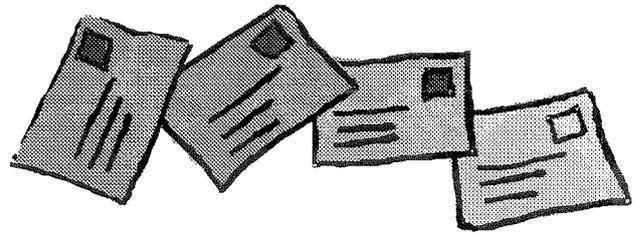
I impure program, self-modifying code

F freeze job when its windows are buried

G guardian window size

U unlock job windows so it can run even when its windows are buried

Letter-Box



Don Trevallion writes:

Unfortunately I am unable to use the cover discs as I do not have the Pointer Environment or Windows. The equipment that I have is:

- * JM ROM QL
- * Gold Card
- * Di-Ren keyboard and interface
- * Dual disc drive (Dennis Briggs, Adman Services)
- * Philips personal monitor
- * Brother M1009 printer

I would like to be able to make use of the cover discs but I do not have any idea of what I need to purchase to make this possible. There may be other people in the same situation as me. How about an article on how to do this?

If you can do this, please keep the information simple and to the point, and the costs as low as possible. Not too many abbreviations, as I am far from being a computer whiz kid. I still enjoy using the QL and at 68 years of age I do not wish to spend a lot of money on a new system. Print this in Letter Box if you wish.

Editor's reply:

After more than a decade and a half of using QLs, it is all too easy to lose track of the fact that we do not all have the latest computer hardware and software, and that a large number of people might be "in the same situation as" Mr Trevallion.

In brief:

To use 'pointer environment' or 'pointer driven' programs, you either need the SMSQ/E operating system or to add the 'pointer environment' files to QDOS. The simplest way is to purchase a program that uses pointer environment, most

include the necessary files, called PTR_GEN, WMAN and HOT_REXT. A few such programs also require the MENU_REXT system you get with some of the se programs. These contain the necessary software for most of these programs. Use the LRESPR command to install these files before you run the program concerned. See David Denham's Hotkeys article in Volume 4 Issue 3 page 28 for an introduction to how to set up a minimal pointer environment system. We may run an article soon (any volunteers to write it?) on how to go about moving your system to Pointer Environment.

Ultimately, the most comprehensive way of getting into pointer environment is to purchase QPAC2, although some people find its manual a bit daunting at first. An easier way of easing yourself into pointer environment is to get hold of a copy of Norman Dunbar's Pointer Environment Idiot's Guide, which can be obtained free or for very minimal costs from some PD libraries, bulletin boards and web sites. If you cannot get hold of a copy elsewhere, send me a floppy disk and return postage and I'll copy it for you.

I have been trying to persuade some of the traders to put together an introduction book or disk for pointer environment. If you think this is a good idea, let QBranch or Jochen Merz know so that they know there will be a demand for it.

QBranch & JMS have taken notice of this! You should soon be able to purchase the "essential bits", i.e. Pointer Environment, HOTKEY System and MENU Extension on disk for a small fee!

Cover Disk & Postcard

Many thanks to those who returned the postcard. Unfortunately, we had not put a deadline for the cards in the last QL Today, and cards arrive every day. Therefore it would be a bit unfair to declare a winner now. Every card which arrives before June, 30th, 2000, will count - the ones which arrive later will not. And then, as promised, the winner will be drawn! However, the postcard has helped in deciding what to do with the cover disk this time. Slightly over 91% of our readers said that they are able to handle HD disks.

Therefore, we went ahead and decided to do this issues cover disk. If you cannot handle HD's, then ProWesS wouldn't be of any use to you anyway, so it is not a question of DD or HD, no it is a question of at least HD or not at all. The next cover disk will be DD again and suitable for everybody.

Many thanks to Joachim van der Auwera to allow us to add ProWesS to QL Today.

There is no printed documentation for ProWesS but you can print it yourself with the ProWesS Reader (also on the disk). Just stick the disk into flp1_ and BOOT and follow what is going on on the screen ... and enjoy ProWesS.

Programming in Assembler - Part 7

Norman Dunbar

In the last issue you were left typing away to get a large amount of code entered for the QLDis project. In this issue we slow down a bit and return to assembly language again. This time we are looking at extending SuperBasic by adding extra procedures and functions which can be loaded after boot up and then used by any SuperBasic program that we load or type in afterwards. Along the way, we will have to take a look at the manner in which we do the following:

- link assembler extensions (procedures and functions) into SuperBasic
- fetching parameters
- testing separators (eg the '#' before a channel number etc)
- the maths stack
- and all its problems
- returning values for functions
- accessing the SuperBasic channel table

Linking to SuperBASIC

When you have written some code that defines a new SuperBasic procedure or function, you must tell SuperBasic what it is called and where it lives. There is a vectored routine to do this and it is called BP_INIT (in QDOS) or SB_INIPR (in SMSQ). As an old hand at QDOS, I still use the QDOS definitions and names. As we are all using George Gwilt's GWASL assembler, and it uses the QDOS names, we shall continue to do so in this series.

Start up the QED editor (or your favourite) and type the following in:

```
start  lea    define,a1      ; Pointer to the definition table
        move.w BP_INIT,a2    ; The vector we need to use (= $110)
        jsr    (a2)          ; Call the vectored routine
        rts                    ; And return any errors back to SuperBasic
```

You will note that we only execute a small stub of code. This is simply because we are linking the new routines into SuperBasic and the actual code for the routines will be executed when a SuperBasic program uses one of the new routines. All will become clear.

The definition table required by BP_INIT has to be in the following format and it must start at an even address. A1.L points at the table when BP_INIT is called:

word	How many new procedures (A1 points here)
word	Repeat for each procedure :
word	Offset to code start for this procedure
byte	How many bytes in the procedure name
bytes	The procedure name
word	Zero = end of procedures
word	How many new functions
word	Repeat for each function :
word	Offset to code start for this function
byte	How many bytes in the function name
bytes	The function name
word	Zero = end of functions & table

As an example, our code file will introduce 1 new procedure and the definition table will be set up like the following which you should now type into the editor following on from the code that is already there:

RWAP SOFTWARE

QL Cash Trader v3.5 £5

A well established accounts package for the small to medium sized business, including automatic generation of profit & loss account, balance sheet, VAT return, reports and analysis for audit trails and management decisions. Previously sold for over £100.

QL Payroll v3.5 £5

Manage a payroll for a small to medium sized business. Handles up to 99 employees easily, producing P45s and P60s as well as the payslips on a monthly or weekly basis. Calculates tax and national insurance and is easy to update to take account of current tax year rules.

Q-Help v1.03 £10

Provides on-screen help for SuperBASIC commands, including most toolkits (TK2, Turbo Toolkit, SMSQ/E and PD toolkits). Copy of PD toolkits available separately for £2. Can easily be used to add help pages to your own programs - simply produce ASCII text files containing each help page, an index to the help page and the program will automatically cross-reference and display the links on screen.

Sidewriter v1.08 £10

Produces landscape printouts of Easel/QSpread spreadsheets and output from QL Genealogist, as well as any other standard text file. You can specify the fonts to be used on the page. Works with all EPSON compatible printers, from 9 pin dot matrix up to inkjet printers. A most useful utility written by Dilwyn Jones - you know it must be easy to use.

ProForma ESC/P2 Drivers v1.01 £8

New improved printer drivers, providing up to 720 dpi printout in full colour from all programs written for use with ProWesS, such as LineDesign and Paragraph. Work on all Epson inkjet printers which support binary mode compression (740,850 and 900 models at least). 1440 dpi to follow.

QL Genealogist v3.24 £20

Genealogy For Windows £50

These programs enable you to keep a track of your family tree - add individuals, with details of their parents and children, and watch all of those links build up into a formal family tree layout. Notes and pictures connected with an individual may also be stored, making this the perfect way to preserve the history of your family for future generations. The QL version now allows you to keep details of both the male and female side of the family in one large family tree. Easy to use with a step by step tutorial for starters. This is a special offer price - valid until 31/3/2000.

D-Day MKII v3.04 £10

Grey Wolf v1.8 £8

For the wargaming enthusiast - D-Day is a classic table top wargame, where you control either the Allies or the Axis forces and play against either the computer or another human player. With the ability to define your own army set ups and a choice of four different scenarios, this should keep you entertained for a while. Grey Wolf places you in charge of a submarine - can you sink the enemy shipping whilst avoiding their planes and destroyers??

SBASIC SuperBASIC Reference Manual £40

Updates £6 each, £10 for 2 (Current Version - ReL3)

Have you ever tried to write a program, but been lost as to the means of performing a certain action? This Reference Manual provides you with a full description and examples of how to use all of the keywords found on a standard QL, plus the keywords under SMSQ/E, Toolkit II and many different public domain toolkits. Details of any possible problems are provided, together with descriptions of how to use the device drivers and how to ensure that your programs are compatible across the range of QL platforms.

This book is ideal for all QL users and is kept up to date by regular updates.

Orders are currently being taken for the next print run of this popular tome.

(Note Price for the book does not include postage and packing).

Return To Eden v3.08	£10
Nemesis MKII v2.02	£8
The Prawn v2.01	£8
HorrorDay v3.1	£8
West v2.00	£5
The Lost Kingdom of Zkul v2.01	£5

Classic QL adventures, now re-released without any need for microdrives. These include mainly text adventures, catering for all tastes, from the spoof Prawn, through to a Hammer Horror, fighting the bad-guys in the old West and battling with robotic hoards and goblins. Return to Eden is a massive three disks of adventure, with pictures for each location and a captured prince to rescue. With three characters to control, each with their own abilities and skills, this one should keep you amused for many an evening.

All six adventures are available together for only £25.

FlightDeck v1.04 £10

Can you learn to fly a twin-engined passenger jet? This simulator includes full shaded 3d views of the world around which you are flying, together with the ability to add navigation beacons, airports and even landscape features to make FlightDeck the ultimate QL Flight simulator. A database of the main UK airports is included to allow you to fly around the UK for training missions.

Q-Route v1.08C £25

This is the latest version of this popular route finding program. Find the quickest route or the shortest route between any two places, using roads. A wide range of maps is available for this program. (see elsewhere in this advert). The program is easy and quick to use. You can even add your own places and roads to the maps to include local detail.

A range of games to keep you amused on the QL. Some are old favourites, like Golf and a quiz program (with over 500 questions). Whilst others are fast, colourful arcade games. Plenty of variation and skill required - what more can you ask for? All 5 programs £20 only.

Open Golf v5.20	£8
QuizMaster II v2.07	£5
Stone Raider II v2.00	£5
Hoverzone v1.2	£5
Deathstrike v1.5	£5

These are the latest maps for Q-Route (now at v1.08C). Find your way around the various countries covered. South and West Yorkshire Map is a much more detailed area of that beautiful part of the British Isles.

Britain.map v1.09	£2
BIG Britain Map (needs 2MB) v2.01	£5
South & West Yorkshire Map v1.03	£1
Ireland Map v1.00	£2
Belgium Map v1.01	£2
Catalonia Map v1.01	£2



RWAP Software, 4 Anvil Crescent,
Coseley, West Midlands
WV14 8GA

TEL: 01902 836888

* Also known as Trading Accounts

Cheques in £sterling
payable to 'R.Mellor'

```

define dc.w 1 ; 1 new procedure
      dc.w psi_cls-*
      dc.b 7, 'PSI_CLS'
      dc.w 0 ; End of procedures

      dc.w 0 ; Number of functions
      dc.w 0 ; End of functions

```

Notice that the format of the procedure name is slightly different from normal QDOS string in that the size of the name is stored in a BYTE and not in a WORD.

Now then, there is a caveat - isn't there always ? If the average length of the names of all the procedures, or functions, is greater than 7 then the simple word for the number of procedures or functions is changed to the value given by this calculation:

(total number of characters in proc names + number of proceduress + 7) / 8

Checking our table above we have a total of 7 characters in the procedure name and there is 1 new procedure. This gives an average of 7 characters per name (round up always !) so we are ok.

And that is it. On QL's of JM vintage and below, the machine must be NEW'd before you can use them. On JS and above, this need not be done.

Once a set of procedures and/or functions has been linked into SuperBasic, the definition block is no longer required. If your code requires the use of some workspace, then you can use the definition table. Just make sure that you don't use more bytes that there are available!

So, let's write our first procedure.

Procedures

Procedures in assembly are very much like PROCedures in SuperBasic. For example, consider the following:

```

DEFine PROCedure PSI_CLS(chan%, P%, S%, I%)
  PAPER #chan%, P%
  STRIP #chan%, S%
  INK #chan%, I%
  CLS #chan%
END DEFine PSI_CLS

```

I have left the line numbers off in the above example. This simple routine is probably at the heart of many SuperBasic programs and is called like this:

```
PSI_CLS 1, 2, 4, 0
```

To give channel #1 red paper, green strip and black ink. Assembler procedures are very similar and in fact we shall now dive straight in and convert the above into assembler.

Back into the QED editor with the code from the start of this article typed in. We have so far typed the code to link the new procedure and the definition block for the new procedure, now we need to write the code for the procedure itself. Your file should look like this so far:

```

start lea define,a1 ; Pointer to the definition table
      move.w BP_INIT,a2 ; The vector we need to use (= $110)
      jsr (a2) ; Call the vectored routine
      rts ; And return any errors back to SuperBasic

define dc.w 1 ; 1 new procedure
      dc.w psi_cls-*
      dc.b 7, 'PSI_CLS'
      dc.w 0 ; End of procedures

```

```
dc.w 0 ; Number of functions
dc.w 0 ; End of functions
```

In the definition table there is an offset word to the start address of the new procedure. Ours is defined like this:

```
dc.w psi_cls-*
```

Which is a useful way to get the assembler to calculate the offset for us. The '*' is assembler short-hand for 'where I am now' or 'the current address'. Our example uses the label psi_cls so our code has to start there.

On with the procedure. In assembler you must take great care to ensure that you have enough parameters etc (see below) and that they are all the correct type. In this example, we will get using integer parameters but the first one must have a hash (#) in front of it. Of course, when using INK, PAPER etc in SuperBasic, you can default the channel number and #1 will be used instead. This means that the following statements are equivalent:

```
PAPER #1,2
PAPER 2
```

It would be nice if our PSI_CLS routine did a similar thing so that the following was equivalent:

```
PSI_CLS #1, 2, 2, 0
PSI_CLS 2, 2, 0
```

This turns out to be quite easy to do.

Here then, is a list of what our procedure must do :

1. Count how many parameters were supplied. There must be 3 or 4.
2. If 4 parameters supplied, check that the first parameter has a hash in front of it.
3. Fetch all parameters onto the maths stack.
4. Convert parameters from maths stack to registers & validate them.
5. Set the paper, strip and ink colours for the correct channel, defaulting to #1 as appropriate if only 3 parameters were supplied.
6. Clear the channel using CLS.
7. Return to SuperBasic.
8. Abort nicely whenever it detects an error.

Type the following after the definition block:

```
err_bp    equ    -15                ; Bad parameter error
err_no    equ    -6                ; Channel not open
bv_chbas  equ    $30              ; Offset to channel table
bv_chp    equ    $34              ; Offset to channel table end
bv_rip    equ    $58              ; Maths stack pointer

psi_cls   move.l  a5,d7            ; End of parameters
          sub.l   a3,d7            ; Minus start of parameters
          divu   #8,d7            ; How many parameters ?
          cmpi.w  #3,d7            ; Are we defaulting channel id ?
          beq.s  hash_ok          ; yes, skip hash check
          -----
          * We do not have 3 parameters so test for 4 and if not found, error exit. If we
          * do have 4 then the first must have a hash in front.
          -----
hash_check  cmpi.w  #4,d7          ; We better only have 4 parameters
           bne.s  error_bp        ; Oops !
           btst  #7,1(a6,a3.l)    ; Is there a hash before parameter 1 ?
           beq.s  error_bp        ; No, we reject it then

hash_ok    move.w  ca_gtint,a2     ; We want word integers
           jsr   (a2)             ; Fetch them all
           tst.l  d0              ; Did it work ?
```

```

        beq.s    got_ok          ; Yes it did
        rts                    ; Bale out with error code otherwise

*-----
* We expected to get 3 or 4 parameters and should have, but now that we have
* got them, check to make sure we have received that which we expected to.
*-----
got_ok    cmpi.w    #4,d3        ; Were there 4 of them ?
        beq.s    got_4          ; Yes

        cmpi.w    #3,d3        ; Maybe default channel in use
        beq.s    got_3          ; So that is ok too

error_bp  moveq    #err_bp,d0    ; Bad Parameter error code
error_exit rts                    ; Bale out with error

*-----
* We have 4 parameters, so fetch the channel id into D0 - this is the first
* of the parameters. We need to tidy the maths stack as well so that get_rest
* works correctly regardless of whether we have 3 or 4 parameters.
*-----
got_4     move.w    0(a6,a1.l),d0 ; Get channel id
        bmi.s    error_bp      ; We don't like negative channels
        adda.l   #2,a1         ; Tidy stack pointer
        bra.s    get_rest      ; Skip the default channel id bit

*-----
* At this point we default the channel being used to #1. By moving one to D0
* and processing as normal, we can do this without much effort.
*-----
got_3     moveq    #1,d0        ; Default channel is #1

*-----
* Here convert the SuperBasic channel number in D0 into an internal id in A0
* and bale out if it fails, or if the channel is not open or has been closed
* - there is a difference. A closed channel has a negative id while a channel
* not yet opened is not in the table.
*-----
get_rest  bsr      channel_id    ; Convert D0 to QDOS id in A0.L
        bne.s    error_exit    ; Bale out if errors

```

Don't type the following!

At this point we have (A6,A1) pointing to the paper parameter on the stack and A0.L holding the channel id for the requested channel (or the default of #1). Now we can set the paper colour (which does not set the strip like SuperBasic does!)

Looking at the QDOS documentation for SD_SETPA and the others, we see that A1 is 'undefined' on return from the routine. This is bad so we need to preserve it across calls or we can fetch all the parameters first. Registers D4 to D7 are not mentioned in the documentation so they are preserved/not used by the routines so we shall fetch the parameters into these registers first of all and this way we can also validate them for errors.

Continue typing:

```

*-----
* Because we tidied the stack pointer in A1 when we fetched the channel id, the
* following code expects to see the paper colour at 0(A6,A1) and this is the
* same as if we never were supplied with a channel id in the first place -
* cunning stuff eh ?
*
* Fetch the remaining 3 parameters into registers that will not be trashed by
* the QDOS routines that set the paper, stip and ink. We reject any parameter
* which is negative as we don't deal with negative colours and just in case, we
* also mask out the high work of the parameter to ensure it is in range 0 to 255
*
* NOTE : we could do away with the negative check and just mask. This would in
* effect convert from a negative to a positive number - but this is the real

```

TF Services

superHermes

A major hardware upgrade for the QL
All Hermes features (working ser1/2, independent baud rates/de-bounced keyboard/keyclick) PLUS full 19200 throughput on ser1/ser2 not affected by sound IBM AT keyboard interface (plus foreign drivers) // HIGH SPEED RS232 industry standard two-way serial port. 4800cps throughput (supergoldcard - qtpi - zmodem) at 57600bps // THREE low speed RS232 inputs (1200 to 30bps)- incl SERIAL MOUSE driver. Other uses include RTTY/graphics tablet etc // 3 spare I/O lines (logic) with GND/+5V // Capslock/ scrollock LED/Turbo/keylock connectors // 1.5k user data permanently in EEPROM

All this on a professional board about twice the size of the 8049 co-processor it replaces

Cost (including manual/software). £90 (£92/£87/£90)
IBM AT UK layout Keyboard..... £22 (£24/£23/£27)
Serial mouse £11 (£13/£12/£14)
Capslock/scrollock LED £1 (£1.50/£1/£1.50)
Keyboard or mouse lead £3 (£3.50/£3/£3.50)
High speed serial (ser3) lead £4 (£4.50/£4/£4.50)

superHermes LITE

All Hermes features (see above) + an IBM AT keyboard interface only. Entry level superHermes.
Cost (incl keyboard lead)...£53 (£55.50/£51/£53.50)

Minerva

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

The ORIGINAL system operating system upgrade

OTHER FEATURES COMMON TO ALL VERSIONS

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

First upgrade free. Otherwise send £3 (+£5 for manual if reqd).
Send disk plus SAE or two IRCs

MKI...£40 (£41/£40/£43) MKII...£65 (£66/£63/£67)

QL REPAIRS (UK only)

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

£27 incl 6 month guarantee

QL RomDisq

Up to 8 mbyte of flash memory for the QL
A small plug in circuit for the QL's ROM port (or Aurora) giving 2, 4 or 8 mbytes of permanent storage - it can be thought of as a portable hard disk on a card, and reads at some 2 mbytes per second. Think of it - you could fully boot an expanded QL, including all drivers/SMSQ etc off RomDisq at hard disk speed with only a memory expansion needed.

2 mbytes RomDisq.....£39 (£41/£37/£40)
4mbytes RomDisq.....£65 (£66/£63/£67)
8 mbytes RomDisq.....£98 (£100/£95/£99)
Aurora adaptor.....£3 (£3.50/£3/£4)

MPLANE

A low profile powered backplane with ROM port

A three expansion backplane with ROM port included for RomDisq etc. Aurora can be fitted in notebook case and powered off single 5V rail - contact QBranch for details. Two boards (eg Aurora and Gold Card/Super Gold Card/Goldfire) fixed to base. Suitable for Aurora (ROM accessible from outside) & QL motherboard in tower case. Specify ROM facing IN towards boards, or OUT towards back of case.

Cost.....£34 (£36/£33/£35)

I2C INTERFACES

Connects to Minerva MKII and any Philips I²C bus

Power Driver Interface 16 I/O lines with 12 of these used to control 8 current carrying outputs (source and sink capable)

2 amp (for 8 relays, small motors)..... £40 (£43/£38/£44)

4 amp total (for motors etc) £45 (£48/£43/£50)

Relays (8 3a 12v 2-way mains relays (needs 2a power driver)..... £25 (£28/£23/£27)

Parallel Interface Gives 16 input/output lines. Can be used wherever logic signals are required. £25 (£28/£23/£27)

Analogue Interface Gives eight 8 bit analogue to digital inputs (ADC) and two 8 bit digital to analogue outputs (DAC). Used for temp measurements, sound sampling (to 5 KHz), x/y plotting £30 (£31.50/£29/£30)

Temp probe (-40°C to +125°C)..... £10 (£10.50/£10/£11)

Connector for four temp probes £10 (£10.50/£10/£11)

Data sheets £2 (£2.50/£2/£3)

QL SPARES

Keyboard membrane £24 (£25/£24/£27)

1377 PAL £3 (£3.50/£3/£4)

Circuit diagrams £3 (£3.50/£3/£4)

68008 cpu or 8049 IPC £8 (£8.50/£7.50/£9)

8301/8302 or JM ROM or serial lead.. £10 (£11.50/£10/£11)

Power supply (sea mail overseas)..... £12 (£17/£16/£21)

Prices include postage and packing (Airmail where applicable). Prices are: UK (EC/Europe outside EC/Rest of world). Payment by cheque drawn on bank with UK address, debit card/Mastercard/Access/Eurocard/postal order or CASH! (No Eurocheques). SAE or IRC for full list and details

26 Sep 99



29 Longfield Road, TRING, Herts, HP23 4DG
Tel: 01442-828254 Fax/BBS: 01442-828255
tony@firshman.demon.co.uk http://www.firshman.demon.co.uk



* world (?) and we have to perform parameter validation.

```
*-----
    move.w 0(a6,a1.1),d4      ; Paper in D4
    bmi.s  error_bp          ; Negative is bad news
    andi.w #$00ff,d4         ; Force range 0 - 255

    move.w 2(a6,a1.1),d5      ; Strip in D5
    bmi.s  error_bp          ; Negative is bad news
    andi.w #$00ff,d5         ; Force range 0 - 255

    move.w 4(a6,a1.1),d6      ; Ink in D6
    bmi.s  error_bp          ; Negative is bad news
    andi.w #$00ff,d6         ; Force range 0 - 255

    adda.l #6,a1              ; Tidy the stack

    moveq  #sd_setpa,d0        ; Paper trap code
    move.w d4,d1              ; Paper colour
    moveq  #-1,d3              ; Infinite timeout
*                               ; Channel id is still in A0
    trap   #3                  ; Set the paper
    tst.l  d0                  ; OK ?
    bne.s  error_exit         ; No bale out
```

*-----
* Now the paper has been set, and the documentation says that A0 is preserved
* along with D3, we can set the strip colour now.
*-----

```
    moveq  #sd_setst,d0       ; Strip trap code
    move.w d5,d1              ; Strip colour
    trap   #3                  ; Set the strip
    tst.l  d0                  ; OK ?
    bne.s  error_exit         ; No bale out
```

*-----
* Now the strip has been set, and the documentation says that A0 is preserved
* along with D3, we can set the ink colour now.
*-----

```
    moveq  #sd_setin,d0       ; Ink trap code
    move.w d6,d1              ; Ink colour
    trap   #3                  ; Set the Ink
    tst.l  d0                  ; Ok ?
    bne.s  error_exit         ; No bale out
```

*-----
* And finally, we can CLS the screen. You have seen this before in QLTDIs.
*-----

```
    moveq  #sd_clear,d0       ; CLS whole screen
    trap   #3                  ; Do it
    bra.s  error_exit         ; All done
```

*-----
* This routine takes a SuperBasic channel number in D0 and converts it
* into a QDOS internal channel id in A0. If the channel is closed or not
* yet opened, the routine returns D0 = ERR_NO and A0 is invalid.
* D0 will be zero if all is ok.
*-----

```
channel_id  mulu   #$28,d0      ; Offset into channel table
            add.l  bv_chbas(a6),d0 ; Add table start address
            cmp.l  bv_chp(a6),d0 ; Valid ?
            bge.s  ch_bad       ; No, channel # off end of table
            move.l 0(a6,d0.1),d0 ; Channel id
            bmi.s  ch_bad       ; Channel closed
            move.l  d0,a0        ; We need id in A0
            moveq  #0,d0        ; No errors
            rts                ; Finished

ch_bad      moveq  #err_no,d0   ; Channel not open (-6)
            rts                ; Bale out
```

QBOX-USA BBS



Operating since 1993 on a Sinclair QL from Utica, Michigan, USA
Supporting ALL Sinclair and Timex users
Message and File Areas for QL, Z88, Spectrum, TS2068, ZX81, TS1000
Modem speeds 300 bps to 33.6k bps supported
24 hour operation - call anytime
810-254-9878

Save the file and assemble it using GWASL. Once all errors have been sorted out, either LRESPR it or ALCHP/LBYTES/CALL in the normal manner. If you have a JM and below, type NEW then try this:

```
PSI_CLS #1, 2, 4, 0 (or PSI_CLS 2, 4, 0)
```

And see what happens when you

```
PRINT 'Hello world' (or PRINT #1, 'Hello world')
```

If you have a JS or above, then just try it without the NEW.

You should see the words 'Hello world' written in black, on a green strip on red paper - assuming your display can handle the colour mixture!

In the code, you will notice that whenever I detect an error, I simply return to SuperBasic with the error code in D0. This doesn't look very friendly does it? Actually, QDOS is very friendly when it comes to procedures because in the event of an error, QDOS will do all the tidying up that we need to do so we don't have to worry about it. This is discussed below in 'FUNCTIONS' and in 'MATHS STACK'.

Functions

Wouldn't it be nice to do this instead of the above:

```
PSI_CLS #1, RED, GREEN, BLACK
```

In SuperBasic this would be done either by:

```
DEFine FuNction RED  
  return 2  
END DEFine RED
```

```
DEFine FuNction GREEN  
  return 4  
END DEFine GREEN
```

```

DEFine FuNction BLACK
    return 0
END DEFine BLACK

```

OK, I know it could be done like this:

```

RED = 2
GREEN = 4
BLACK = 0

```

but we are dealing with machine code functions and this is more illustrative of what we are about to do. (So there!)

We shall now extend our original example so that we can specify colour values by name - this is much more friendly in my opinion.

The following two lines in the definition block need to be removed:

```

    dc.w    0                ; Number of functions
    dc.w    0                ; End of functions

```

And replaced by the following:

```

    dc.w    8                ; There are 8 functions

    dc.w    black-*         ; First function
    dc.b    5, 'BLACK'

    dc.w    blue-*          ; Second function
    dc.b    4, 'BLUE'

    dc.w    red-*           ; Third function
    dc.b    3, 'RED'

    dc.w    cyan-*          ; Fourth function
    dc.b    4, 'CYAN'

    dc.w    green-*         ; Fifth function
    dc.b    5, 'GREEN'

    dc.w    magenta-*       ; Sixth function
    dc.b    7, 'MAGENTA'

    dc.w    yellow-*        ; Seventh function
    dc.b    6, 'YELLOW'

    dc.w    white-*         ; Eighth function
    dc.b    5, 'WHITE'

    dc.w    0                ; End of functions

```

The following is the code for the new functions, type it into the file after the end of the 'channel_id' subroutine:

```

black    moveq    #0,d7
         bra.s    return_d7

blue     moveq    #1,d7
         bra.s    return_d7

red      moveq    #2,d7
         bra.s    return_d7

magenta  moveq    #3,d7
         bra.s    return_d7

green    moveq    #4,d7
         bra.s    return_d7

cyan     moveq    #5,d7

```

```

        bra.s   return_d7

yellow   moveq   #6,d7
        bra.s   return_d7

white    moveq   #7,d7

```

```

*-----
* This routine returns the word value in d7 to SuperBasic as the result of the
* function we are running. It requires two bytes on the top of the maths stack
* and because there were no parameters supplied to any of the functions, I can
* safely ask QDOS for these two bytes.
*-----
return_d7  move.l  bv_rip(a6),a1   ; Because we had no parameters passed
          moveq   #2,d1           ; Size of stack space required
          move.w  bv_chrix,a2      ; Routine to allocate maths stack space
          jsr    (a2)             ; Go get some space NO ERRORS OCCUR !

*-----
* The maths stack has been extended by two bytes BUT it may have moved around in
* memory so we need to get the stack pointer into A1 again.
*-----
          move.l  bv_rip(a6),a1   ; New top of stack
          subq.l  #2,a1           ; Make space for our integer result
          move.w  d7,0(a6,a1.1)   ; Stack the result
          move.w  #3,d4           ; Signal word result on stack
          move.l  a1,bv_rip(a6)   ; Store new top of stack for SuperBasic
          clr.l   d0              ; No errors
          rts                    ; Return result to SuperBasic

```

That is the end of the code. Assemble it, debug it and test it using the following:

```

PAPER GREEN
STRIP RED
INK BLACK
CLS
PRINT "Hello world"

```

or, if you like:

```

PSI_CLS GREEN, RED, BLACK
PRINT "Hello world"

```

In the procedure, PSI_CLS, we obtained some parameters for the various colours and channels. I shall now discuss how this is done in much more detail.

In the next part of this series, you will learn how to get various types of parameters.

Last Minute News from JMS

By the time I write this, Marcel informs me that he has the colour drivers working. I should be able at least to demonstrate them in the USA. Marcel will improve other things as well, e.g. serial ports, parallel port, and will speed up the emulation in general.

First speed tests were encouraging. Although 8 times as much information needs to be transferred (16 bits per pixel instead of 2) most screen operations do not slow down significantly. We will keep you informed about the situation.

QD Version A.07 interacts even better with FiFi now. You can right-click on a word in QD, mark it and start FiFi searching for this word in QD's context - with a single mouse-click! When FiFi has found the word, you click on the found files and another QD will pop up - displaying the selected file starting from the position of the search string. Interaction is very good - you will like it if you program a lot in assembler and want to search for external references. Update is free if you already own QD98.

regarding the colours: we will start updating our software to handle more colours as soon as Tony Tebby has released the specification for the Window Manager with more colours.

One of the hardest things that people can do is to change their whole way of thinking. This is probably the reason that so few people ever manage to do it. It is even worse when it is applied to whole communities because the other member of the community tend to reinforce the norm. The reason for this, slightly philosophical, start to this issues' column is that I have noticed a paradigm shift in the QL community of late.

When Stuart Honeyball brought out the QXL card several years ago he experienced a reaction against it from die hard QLers. Many felt that it was tantamount to treason to consider buying a PC even if it was for the express purpose of running a faster and more powerful QL. Although the QXL upheld the standards that Miracle had always been known for, he sold considerably less of these than he had originally thought he would. These days that attitude has shifted appreciably and, although the QXL cannot be made any more (suffering the same fate as the Super Gold Card by using the same obsolete chip), its spirit is very much alive in the emulators that abound today. We would not have had SMSQ/E had it not been for the QXL.

More and more users are running their programs on various emulators such as QPC and UQLX and getting a better performance than any of the old QL hardware could give them. This is also being backed up by Geoff Wicks excellent 'QL 2 PC Transfer Program' which allows users to move their text files between formats. There

are many things on the PC platform which we have not got on the QDOS/SMSQ one, so combining the two in this way is a very good compromise.

Horses for Courses

There is always a trade off between fun and functionality. The QL for me has always been a source of enjoyment and, although it can be annoying when it fails to be able to do what I want it to do, even my limited programming expertise can bend it to get closer than other, more hide-bound systems.

I have recently been using 'SAGE' the massive accounting program used by many businesses to keep track of their stock, produce invoices etc. This is a very professional product and one which can do vast complex calculations but, in the end the program 'wot I writ' to use the DATAdesign database and produce the Q Branch invoices does all I need it to do and is open to being adapted whenever I feel the need to include other stuff. I could never write anything as complex and far reaching as 'SAGE' but there is a sense of satisfaction when I use something which I have produced myself.

This could be the reason that several of the people who abandoned the QL a while ago to move over to a PC have re-kindled their interest in the system. On the whole the PC is pretty much 'set in stone' unless you invest in the expensive Visual BASIC or 'C' packages. LINUX is attracting those who like the more complex

areas of programming but there is little to compare with the sheer simplicity and flexibility of QDOS/SMSQ Super-BASIC, especially now that faster systems, more memory and a swathe of colours are within reach.

A temptation Avoided

All of this leads me to another little tale. As you may know Jochen produces this magazine using Calamus because it is a very good program and makes it easy to do the job. (and, as you can see, the results are excellent). When we do the flyers for the shows Tony produces them using a PC package of some sort so when I send artwork(?) for the magazine or the flyers I have to send printer dumps from my LINEdesign package. I recently had to do the flyer for the Manchester meeting and I was ever so tempted to insist that they all provided me with LINEdesign files - I did, however, let them get away with and I used a PC program myself.

Any Colour You Like - So Long As It Is Black

It has also been interesting to see that one or two old QL users have been in touch with me recently asking to buy original QLs. I had thought that there would be no market for these and I had about six sitting in a cupboard in the shop. I decided to put them on my website when I did a second user page and I got some immediate requests. Maybe this is because people who tore their systems apart to build them into tower cases or used some of the parts to build Aurora systems have been experiencing a wave of nostalgia and want to make sure that they still have an original.

I have always kept a standard QL with a Gold Card, Colour Monitor and Miracle systems DD disk drive so that I could test software in the basic environment and so I could format DD disks to work reliably. I was recently presented with a JS ROM QL which had obviously never been used. You can always tell what use systems have had because some of the keys have lost their matt finish and developed a polished sheen as a result of numerous key presses. Usually it is the SPACE, ESC and ENTER keys which are affected.

The big problem with retaining a 'heritage' QL in working condition is the keyboard membrane. Tony Firshman tells me that there are very few of these membranes left now and he has been forced to regulate the supply. There is no real way around this problem since manufacturers require a guaranteed run in the hundreds before they undertake to do the job. At one point there were several replacement keyboard tops available for QL cases but these have long since disappeared so the situation is getting desperate for users who still want to use the original product. Maybe someone out there has contacts in a company who might be willing to take on a small run of membranes - do let us know.

Speed vs Colours

My stalwart source for comments in this column, the ql-users internet discussion group, recently had an exchange of views between Claus Graf and Marcel Kilgus about the relative speeds of their projects. I have yet to try QPC II on a processor faster than an AMD K6/II 500 but I still feel that the Q 40 has an edge over it (maybe because I sell Q

40s but I think I am being even handed here). I will be interested to see how much the new colour drivers for QPC will slow it down when they are released. As always there is a trade off between speed and function.

The colour drivers are, at last, beginning to creep into the other systems now and the last Eindhoven meeting saw the release of the new interface for the QXL. As I have mentioned above, the QXL was a very good device which suffered from a few teething problems, bit of system resistance and, what was increasingly regarded as, a slow system interface. Not only does the new version of SMSQ/E for the QXL give you more colours but it fixes some of the slowness inherent in the old disk and graphics interface.

My QXL was sold a few years ago although Q Branch has a few second-user ones that I may try out but this new version could give the old warhorse a new lease of life. A QXL in an old 486 system will give you colour drivers on a budget.

A Moving Story

Just in case you missed it in the Q Branch advert I am moving the Q Branch HQ to it's old loft location back at my house. The reason for this is that I have stopped running the shop at the Bank Volt. There are many reasons for taking this step but the main one is that I have been headhunted into a job with one of my principle hardware suppliers. Steve Hall will now run the shop but he has not been very involved with the day to day running of Q Branch for some time so I felt that it was better for me to re-locate the HQ. This move will also involve the QL

Today UK Office since that is a Q Branch function too.

Be assured that I have no plans to abandon Q Branch and I fully intend to be at as many meetings as ever. Check out the Q Branch advert for details of the addresses and phone numbers.

Turbo'd

Mark Knight and George Gwilt have told me that they now have a version of the Turbo Compiler which allows you to create Pointer Environment Programs. This is a major breakthrough in many ways because there are people who have always used Turbo to compile their programs and the old Turbo code was always very unfriendly towards the PE. The alternative compiler has always been QLiberator and this has been the one favoured by many of the QL programmers over that last few years. There are, however, problems within QLiberator and, in spite of many approaches to its authors, we have always failed to get them to produce new versions. It is evident, from recent experimentations with the Q 40, that the new colour drivers will put a strain on some of the older programs and may throw up many incompatibilities. If we cannot repair the basic QLiberator compiler and there is no longer any support from it's authors then a new one is urgently needed. George Gwilt, Mark Knight and David Gilham have done a lot of work on the old D.P. programs in the last few years and seem to be committed to continuing this work so, if this new Turbo version proves to be as good as is suggested, it will be a welcome addition to our programmer's armoury. The new version of the Turbo compiler is not available for use

yet because it is not fully tested and still needs some fine tuning but George is very optimistic about it and we should see it out there soon.

I want to be Free!

Once again there have been complaints in the ql-users group about the lack of openness in the later QL formats and the fact that some programs actually cost money! (Shock, Horror). The main stimulus for this came about because I suggested that Claus Graf used QMenu as a file selector in his qpv graphics program.

No sooner had I committed this email than I had a raft of complaints by people saying it was not free and therefore could not be used in a free program. True you have to pay for it and true you cannot freely distribute it but it is a very good collection of menu utilities and the one thing we have always lacked in the QL community is some standard way of accessing things.

In other systems, whenever you have to select a file, or save a file or, go to a subdirectory or enter a string, or something along these lines you get a standard interface which has other tools attached to it to make your job easier. Within the QDOS/SMSQ community this has increasingly become the task of the QMENU extensions. There is a lot to be said for a standard interface for these things and the FILE_SELECT Menu is an absolute masterpiece. Yes, you do need to know a bit about it in order to get the most out of it but, even at its most basic level, it is relatively simple to

use and here are a few tips to help you out.

In the enclosed screenshot you can see a typical FILE_SELECT menu.

ESC - Quit the menu,

CF2 - (lightning symbol) - refresh the menu,

OK - select file named in the next line.

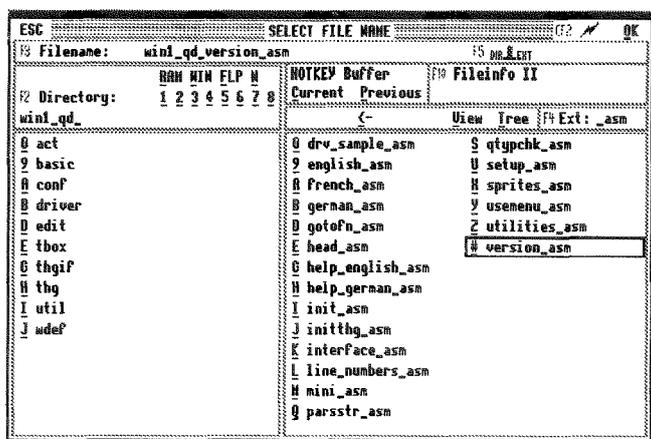
F3 Filename: This is the box which displays the name of the file you have selected. You can left click this and then type a filename if you want.

F5 DIR-EXT - This is a neat way of giving you the subdirectory and the file extension

change the drive number. If you are in WIN1_ and you want to see FLP2_ it is best to change the number first since a directory of win2_ (if it exists or not) is always faster than a directory of FLP1_ (especially if there is no disk in that drive). Below this you will find the name of the drive and letter you are currently displaying. Left click it and you get the cursor so you can type the subdirectory. Right click it and you can get a list of subdirectories to use as well as another opportunity to change device and drive number. These direc-

tories are configurable in the MENU_REXT config block.

Right Box. To the right of this box is one marked HOTKEY Buffer. Right Click this box and it will load the file that was last stuffed into the hotkey buffer. If you have saved a file from some PE. Programs that filename is stuffed into



selected in the box below it with the cursor placed at the junction of the two ready for you to type a name. This is of more use when you want to save a new file but still a clever idea.

Left Box. This is the nitty gritty. The big box will allow you to change the device by clicking on any of devices in the top line. A left click will directory the device selected and display the directory in the box below and to the right. It will assume that the drive number is the same as that already selected. This means if you have a directory of WIN1_ displayed and you click on FLP you will get a directory of FLP1_. Similarly clicking on the number below the devices will

the buffer as it is when you 'HIT' a file from QPAC 2. Below that are two options **Current** and **Previous**. Current will give you the current HOTKEY buffer and Previous the last one you selected. Fairly obvious that one.

Next we have the **F10** option which will call Fileinfo II if installed. If you have a filename selected it will call Fileinfo II and perform whatever action you have selected for that file extension.

Below this is a line with four options: The symbol <- will move you up one subdirectory (ie if you are in WIN1_DATA_ and hit it you will then be in WIN1_). **View:** if you highlight this button by left clicking on it and then select a file from the

menu below you can get a quick preview to ensure you are selecting the correct file. **Tree:** this will expand the tree - beware! if it is a big drive and you are at the root you have time to go and make the tea!

The last box is the extension selection box. This allows you to select only those files with the extensions you are looking for. Left Click this and you can edit the file name manually. Right click this and you get a list of extensions you have preset in the config block for MENU_REXT. This box has an option marked 'No'. Click on this and you get no extension displayed in the box and every file in the subdirectory displayed in the big window below. You can select an extension by clicking on it in the list and the file list is updated to display only those files whose extensions match the selection. If you select the 'Edit' function at the bottom and then click on one of the extensions in the list you can edit that extension. You are also offered the option to have that change saved back to the main program so it will always replace the previous extension in the list.

This may be a long explanation but it just goes to show you what you get with a FILE_SELECT window and this is just one of the menus available in this package. To continue to deny that this is the best option for a file selection window is just 'head in the sand' stubbornness. Maybe not everyone can use it to its full extent but those who have even a slight knowledge of it get frustrated when presented with anything inferior and those who understand it (all my readers now) are even more hampered.

You can add a call to use the FILE_SELECT extension with a single line:

```
File$=FILE_SELECT$('Files'  
,,WIN1_DOC_,_T91)
```

(calls up a FILE_SELECT window with WIN1_DOC_ as the subdirectory and _T91 as the selected extension)

If you write your own files menu it will take several lines and your program will be much bigger. If you are worried that users will not have MENU_REXT loaded you can always use a config block so the end user can choose to use it or not. The program to insert config blocks into a compiled BASIC program is free and available from most BBS and PD Libraries. You could also use the Environment variable system if you prefer that method.

You can use the same variable for your own File Selection menu as in the line which calls FILE_SELECT so you will have added very little to your own program.

Lets face it you have no real reason not to use this great extension and many reasons to use it!

Obituary

I was very sad to hear of the death of Cyril Phillips recently. Cyril was often present at QL shows and was a lively and dedicated user of QL software. Many of us will remember him from the workshops and his enthusiasm for fractals and graphics was unbounded. He will be missed by the London sub-group, of which he was a regular, and by us at Q Branch. Our condolences go out to his wife.

Users Demand....

Ok so here are a few more of those acronyms.

UDMA - Users Demand More

Acronyms

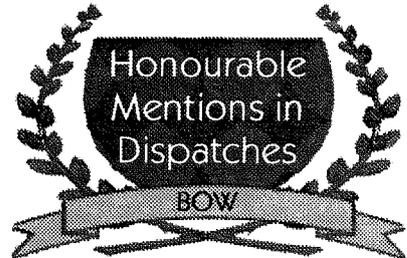
OCR - Often Creates Rubbish
TCP/IP - The Cost of Phoning Increases Phenomenally

EIDE - Eventually Installed Drives Expire

TWAIN - Type Without An Interesting Name

GIF - Greed Is Foremost (see compuserve's licence policy)

ISA - It Slides Away



Once again the honours in this month's issue go to Dave Westbury who has produced yet another little graphics utility for QL systems. I have tried this one out on my Q 40 and the Aurora and it is 'neat' to use Stuart Honeyball's vocabulary. PHOTON is a small program which can be EXEC'd on the above systems to display JPEG images. Dave says that he intends to develop the program to display other formats such as PIC and GIF (including animated GIF Files). He says that it can be used on any QL compatible system and does not require the Pointer Environment. It runs in the standard QL modes 4 and 8 as well as the Q 40 mode 33 (65536 colour mode). It will also make use of the Aurora 256 and 16 Colour modes but only in a compromise which means that the high resolution mode of the Aurora requires a reset. Nevertheless this is a good and efficient piece of programming which will included in all of the Q 40 release disks from now on.

This program is completely free and contains full instructions for use. Dave is beginning to become one of our most prolific programmers.



The QL Show Agenda



Sat., 26th of August - EINDHOVEN!!!

The Italian QL Show will be either 1st or 8th of October, depending on the Austrian QL show which will most likely be held on the 8th or 1st of October. We hope to have fixed dates next issue!

QL 2000

Roy Brereton informs us, that QL 2000 will be held at the Horizon Centre, Sundridge Close, Cosham. Saturday 14th October will be the opening day with the usual QL activities, i.e. Traders, Bring 'n' Buy, Quanta table etc. We will open at 10:00hrs and close at 16:00hrs.

Sunday 15th October will commence at 10:00 hrs with an Open Forum entitled "The Way Forward". It is hoped that everyone will attend as this is vital to the continued existence of the QL community. Looking on the bright side, everything points to a good influx of visitors from overseas including our 'friends' from the USA.

The nearest accommodation is the Travel Inn at North Harbour with a price of £40.95 per room smoking or non-smoking. Breakfast is usually charged at £6 for a full English breakfast or £4 for a Continental breakfast. For those who plan to come to this extravaganza, it is worth booking before the end of June this year to secure your room. Tel: No: 023 9232 1122.

The Travel Inn is located next door to a Beefeater and I have been advised that if evening meals are required then it is advisable to book a few weeks early to ensure seating. If I can have contact from members who intend to come to the event then I can book tables in advance for Friday 13th(!) and/or Saturday 14th October. Please email or phone me asap.

For those sub-groups who wish to take advantage of the Quanta offer of payment for the hire of a mini-coach for that weekend, can they please contact John Taylor in the first instance

(janvtaylor@btinternet.com).

Hopefully, one or two of these mini-coaches can help to ferry our overseas friends to and from the venue.

That's all for now, more info when I get it.

Roy (roy.brereton@tesco.net)