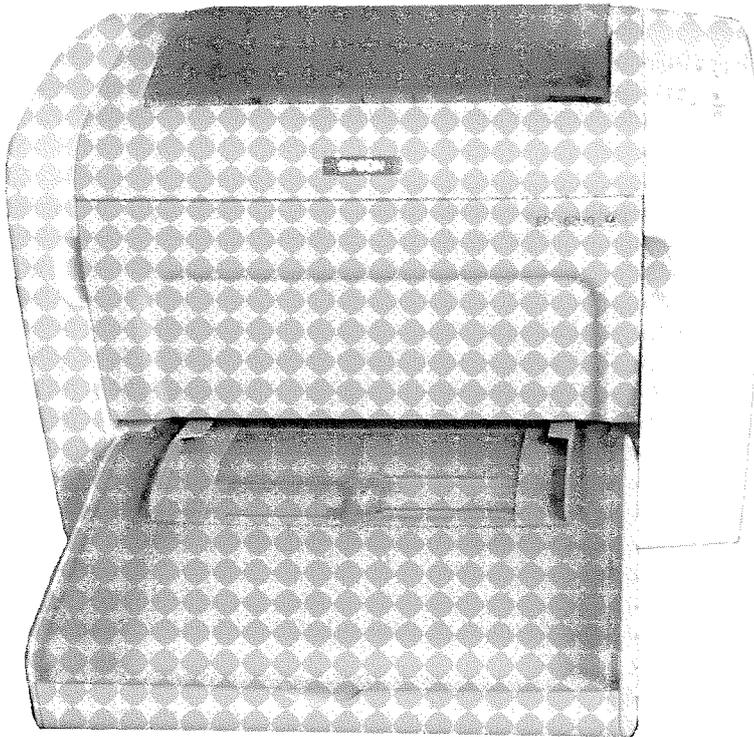


QL Today

Volume 9
Issue 2
July/August
2004

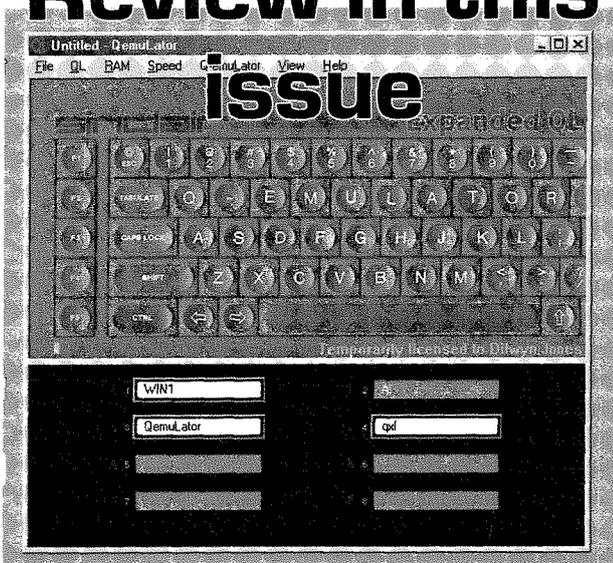
ISSN 1432-5454

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



**The current
EPSON Laser
Printer - is it
any good for
QL Output?
Roy Wood
takes a closer
look for you...**

**QemuLator -
Dilwyn Jones'
Review in this**



**QL 2004
more details...
prepare your-
self for the
biggest QL
event of the
year!**

Contents

3	Editorial	
4	News	
8	Gee Graphics! (on the QL ?) - part 39	H. L. Schaaf
10	A Bit at a Time	Geoff Wicks
16	Q-Emulator Review	Dilwyn Jones
24	JPEG Viewers for SMSQ/E	Timothy Swenson
28	Programming in Assembler - Part 10 (2) Doubly Linked Lists	Norman Dunbar
32	DBAS2HTML	Wolfgang Uhlig
35	Easy Success	Bob Spelten jr
36	Programming QPTR in S*Basic (Part 10)	Wolfgang Lernerz
38	The EPSON EPL 6200 - any good for QL printing?	Roy Wood
40	The MOVEM Instruction and its Register List	Hugh Rooms
48	Julian & Gregorian Calendar	Stephen Poole
50	QPC2 TIP - Starting Windows Files	Dilwyn Jones
50	QPC2 PRANKS	Dilwyn Jones
52	To Eindhoven for only 8 pounds	Al Boehm
53	Byts of Wood	Roy Wood

Advertisers

in alphabetical order

Dilwyn Jones	7
Jochen Merz Software	33
Phoebus Dokos	51
QBranch	26, 27
Quanta	21
RWAP	47
TF Services	43
Geoff Wicks	11

QL Today

ISSN 1432-5454

German office & Publisher:

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

English office:

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

Editor:

Dilwyn Jones email: dilwyn.jones@tesco.net
41 Bro Emrys email: QLToday@J-M-S.com
Tal-Y-Bont, Bangor
Gwynedd
United Kingdom LL57 3YT

Co-Editor:

Bruce Nicholls Tel +44 1708 510764
38 Derham Gardens Fax +44 870 0568755
Upminster email: qltoday@q-v-d.demon.co.uk
Essex RM14 3HA email: QLToday@j-m-s.com
United Kingdom

QL Today is published bi-monthly, our volume begins at the 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 2004 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.

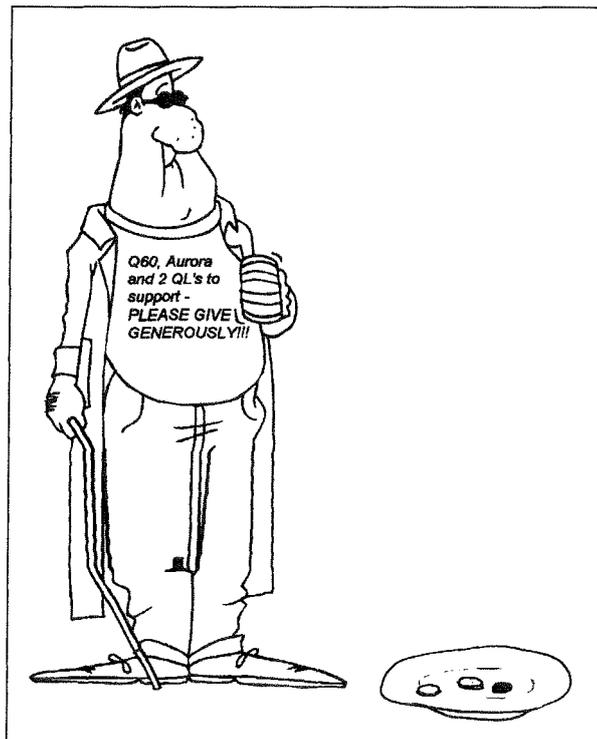
It's been rather a hectic time for me lately. What with changing jobs, having to quickly master all sorts of non-QL computing and generally a lot happening at this time of year. Time at the QL has been at a premium for me lately. Which just went to prove to me how much I still enjoy QLing – every time I did get a chance to use QDOS or SMSQ/E it was like a first bite of chocolate after two weeks without any!

Recently, I needed a little utility to help with the renaming of QXL.WIN path names in QPC2 and decided to write one myself to make sure it was just what I wanted. The program was written in a day or two – see QXL.WIN Manager in the news pages – and given how busy I was at the time the experience of knocking up a complete little program like this in such a short time with Easyptr once again reminded me just why the QL remains such a part of my life. As I write this, I've been setting up a complex spreadsheet at work, a job which took up the best part of a week interspersed with other day-to-day work, to do one very specific task. Then I come home and remind myself that on the QL I can put together a program in no time at all – it took longer for me to do something in an already written program at work and it's still got bugs.

Long live QDOS and SMSQ/E!

During June, I had a month of using the full version of QemuLator for Windows, courtesy of a temporary registration code granted by its author, Daniele Terdina. For whatever reason, the free emulators QLay & uQLx and the commercial one QPC2 get much more publicity than QemuLator. Well, I finally realised what I was missing out on. It can run QDOS or SMSQ/E, very fast, and... well, see my review in this issue for more information. There is much more to this emulator than I'd previously thought and it deserves to get all the users it can get.

Once again, a big thank you to our loyal readership for staying with us and an equally big thank you to all our regular contributors!



Cartoon

NEWS

QL2004

Several activities are planned for the QL2004 show in Eindhoven, hosted by Sin-QL-Air, the Dutch user group. We hope to revive the spirit of the old style Eindhoven international shows for which Sin-QL-Air was world famous.

INTERNET: Most QL-ers know it is now possible to email from a QL, but do not know how near we are to a user friendly email program. Jon Dent will demonstrate the latest versions of his software.

Apart from Jon's work we intend to have an internet connection available throughout the day.

COLOURS: The GD2 colour drivers have now been available for about 5 years, but we urgently need more programs in the new colours. Wolfgang Uhlig will lead a colour workshop, and many people who have been active in writing programs and utilities using the new colours will be present at QL2004. This is a unique opportunity to learn how to use the colours in your own programs.

ALTERNATIVE POINTER: Everyone thought pointer programming was restricted to QPTR, EasyPtr and QLiberator until George Gwilt came along. He will demonstrate the software he has written to allow pointer programming in GD2 colours using the Turbo compiler.

QL PLATFORMS

NATIVE HARDWARE: Nasta hopes to be present to describe the latest progress in developing expansion cards for native hardware. (If Nasta is unable to make it, then Roy Wood will take over.) A good chance to discuss native hardware problems.

Q60: Unfortunately we are not expecting any official Q60 representation at QL2004, but Wolfgang Lenerz will bring his Q60 along. George Gwilt, who develops his programs on a Q60, will use this machine to give his demonstration. An opportunity for Q60/40 users to meet together to discuss mutual problems.

QPC: Wolfgang Lenerz says if we have something for the other two platforms, there we should have something for QPC users. Nothing planned as yet, but, if anyone has ideas, we will be pleased to do it.

QL FORUM: We hope to end the show with a QL Forum. We have a few names for the panel, but

are keeping nominations open to ensure we have a balanced view of QL opinion. Nominations for the panel and possible topics for discussion are welcomed.

I have updated the QL2004 website. It now gives more details of the planned activities plus some extra hotel and tourist office information.

The Hotel Parkzicht has been removed from the hotels list as we understand it has now closed. Please remember to ask about weekend prices at hotels.

<http://members.lycos.co.uk/geoffwicks/ql2004.htm>

JUST WORDS! MAKE-OVER

Just Words! will be unveiling its GD2 house style at the QL2004 Eindhoven show, and hopes to release several upgraded programs from its range on the day. Priority is being given to the older, freeware programs to enable the maximum number of QL-users to benefit from the new colours. These programs will also be available for downloading from the Just Words! website.



QPC2 v3.20

Undergoing beta testing at the time of writing this, version 3.20 of QPC2 is a major rework of QPC2 by **Marcel Kilgus**. He switched programming tools and undertook a significant rewrite of much of the code, hopefully producing a much more robust emulator as a result (which does not mean in any way that QPC2 is unstable!). The first beta release of this version included several improvements to aspects of QPC2 concerning Windows Direct-X which has allowed the following improvements:

- run several QPCs in parallel
- change to fullscreen mode and back, even when several QPCs are running.
- change Windows' colour depth/resolution while QPC runs in window mode.
- put laptop in standby/suspend mode.

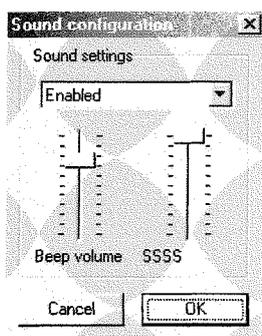
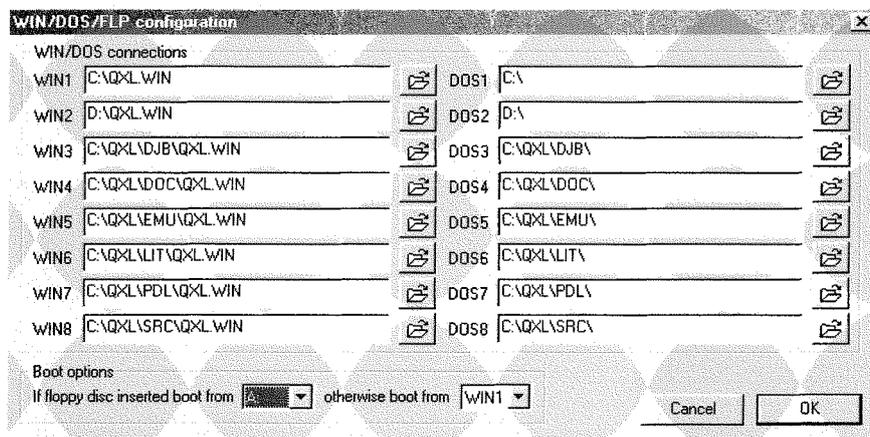
The third Beta release included some improvements to the sound system, most significantly an improvement to the handling of short BEEPs to

work around limitations with DirectSound and some sound drivers. And as he happened to be working on it anyway, QPC now supports SSSS, the SMSQ/E sampled sound system! Marcel said that this was quite a tough job, but it seems to work fine now. Only basic Q40 compatibility so far, i.e. 8-bit 20kHz stereo.

All available tools for the Q40 seem to work fine (especially Simon Goodwin's SOUND device). Except Claus Graf's QSPLAYER, because it refuses to run on anything except a Q40.

There is now a unified WIN and DOS configuration dialog and a new sound configuration dialog. Switching Windows jobs using ALT+TAB doesn't send a TAB to SMSQ/E any more.

Fixed handling of SYS_CAPS capslock system variable, where one of the two bytes had been holding an incorrect value from early days of QPC, which may have upset one or two of the software Caps Lock indicators around.



Above:
The new combined WIN/DOS configuration dialog

To the left:
The new SSSS configuration dialog

Q-WORD

Phoebus Dokos writes:

"Thanks to a much anticipated new QPC version by Marcel Kilgus (QLSSS is supported), we can now use the same Music Server software on both Qx0 and QPC... that means that QPC will not suffer from sound playback latency... I am working as fast as I can to fix the code for QPC... That also creates changes on the way Q-Word is installed which will be able now to be completely independent!

I already have done necessary changes to create three language editions of Q-Word (English, German, French) and these will be all bundled for release with the new changes in code for QPC.

The required version of QPC2 for the sound to work will be 3.20

PASCAL

Jean-Yves Rouffiac has secured permission from the former MD of Computer One (a company which released programming software such as assemblers and monitors in the early days of the QL) to make their Pascal compiler for the QL available on a non-commercial basis. Copyright on the software is retained. The software itself will be available from my software download site www.dokos-gr.net/~dj/ as well as available through the usual sources of freeware QL software.

At present, all we have as a manual for this software is a scanned set of the main pages, which are saved as graphics, so quite large. These manual files are too large to place on the website as they are, so will initially only be available from my PD library service until the full manual can be scanned and OCR'd as text rather than graphics

KNOWARE Website

Per Witte reports on software added to his website:

QUIT works with Minerva Multibasics
Added Button - a MC toolkit to use the Button Frame

Added DelDir. A mini directory tree traversal tutorial.

Added HDRen. A routine to rename QXLWIN (QLWA) hard disks.

Added Findname. A MC function to check for S*Basic function/procedure names.

All this and more can be found at

<http://knoware.mysite.freemove.com/index.html>.

SQLUG Newsletter

John Sadler reminds us that subscriptions are still available for the SQLUG Magazine (Scottish QL User Group).

The magazine consists of 4 A5 pages a month containing news of and articles by George Gwilt, or by John himself

At the time of writing (1st July 2004), subscription was £4.50 for the remaining 10 months for addresses in the UK

Please forward subscriptions to

John Sadler, 8, Broom Place, Kirkcaldy, Fife, Scotland KY2 6UE, United Kingdom

GWASL 1.19

Norman Dunbar reports that he has received a version 1.19 of the Gwasl (Gwass Assembler Lite) assembler for 68008 processor systems. The full Gwass assembler only runs on systems with 68020 or later processors. This version of Gwasl fixes a problem Norman Dunbar ran into when preparing his assembler articles for this magazine – the new version allows more space for labels to allow for users like Norman!

CPTR Update

New versions of Cptr are on the SQLUG website. The main changes are in the examples. There is a new version of UCALib. Note that if you use the cptr2_o you may find you get RELOC_START link errors when you compile a program. Just use the old version.

www.jms1.supanet.com

Clocks and Calendars

This is a small suite of programs from **David Denham** and **Dilwyn Jones**, which were written following the Clocking In series in QL Today.

AnaClock is an analogue clock display which can be resized from about 1 text character in size right up to the full screen size.

Dates Calculator is a general purpose date handling utility, which does things like working out the number of days between two dates. Enter your birth date and today's date and it will tell you how old you are, for example. It will also work out Easter dates, and which day of the week that a given date falls on (e.g. your birthday next year will be on a Tuesday!)

MiniCal is a small display calendar, which shows the current month and lets you step backward or forward a month at a time.

Button Clock is a little digital clock which runs in the QPAC2 button frame, using Jochen Merz's Button Frame extensions.

These programs will all be available as freeware from the usual sources.

SUCCESS

Wolfgang Uhlig writes:

I want to announce the colour version of SuQcess, the database frontend for the QL. After having overcome (nearly) all the EasyPtr-difficulties with the help of Marcel Kilgus and Per Witte, Bob Spelten and I gave the program a totally new outfit.

With all the improvements Suqcess has had during the last two years, it is now a program really worth giving a try.

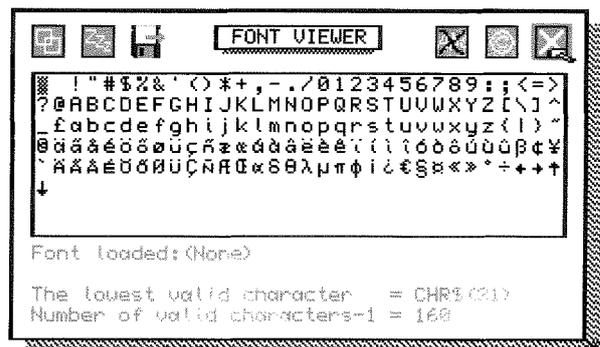
You can download a trial version on my website:

www.uhlich.nl/ql

Every feature can be tested, the only thing that does not function is saving changes to the very database.

FONT VIEWER

This is a small pointer driven utility for viewing QDOS or SMSQ text font files. The program simply displays all characters in a given font file. It can be used with File Info 2 to view fonts directly from QPAC2's Files menu via the Exec command, or from Q-Trans's Exec command, if File Info 2 is set up for the correct action with the font file's filename extension.



J-M-S is moving

Please note that the address will change on the 1st of October. "Im stillen Winkel 12" will become "Kaiser-Wilhelm-Str. 302" ... the rest, including the postcode, phone numbers etc., will stay the same. This issue is produced under high pressure (we're renovating as well) and it will go on for the next six weeks or so, at least.

So, in case you find it difficult to reach J-M-S on the phone, then you know why.

Please use E-Mail 'til October, then I can reply whenever time permits:

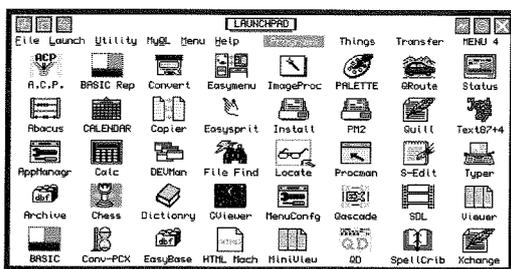
SMSQ@J-M-S.com

If you have a look at the J-M-S homepage you may find that, after quite some time, something has actually happened:

SMSQ.J-M-S.com

LAUNCHPAD

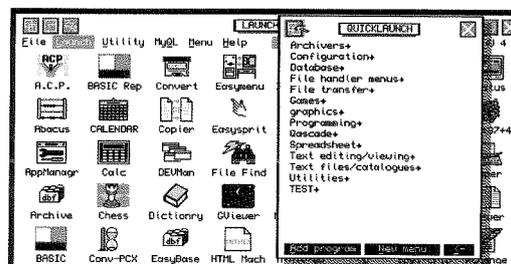
A New Graphical Interface For The QL



An all-new graphical user interface for QDOS or SMSQ/E from Dilwyn Jones. Fed up of typing in EXEC 'filename'? With this pointer driven application, simply set up desktop icons to launch programs, or create menus or "lists" of programs to launch, all in a simple to use no-nonsense graphical "point and click" system – use with mouse or keyboard. Runs on anything from expanded memory QL to QPC2 or Q60, as long as you have pointer environment and Toolkit 2

- Set up icons for launching your programs with a single mouse click from one of four desktop surfaces
- Set up your own program launching menu
- Up to 16 users, all with optional passwords
- MyQL menu to customise and remember your QL settings for each user
- Runs as a simple job, does not prevent you using BASIC or anything else
- Several accessory programs (calculator, calendar, screen saver, file handler, games and so on)
- Revolutionise your QL system – you'll wonder how you managed without it!

£20.00



THE QL CD-ROMs ARE STILL AVAILABLE...

QL Emulators CD £5.00 - QL Documentation CD £5.00 - QL PD-CDR £5.00
QL PD Library CD £10.00 - QL Religion CD £10.00 - QL Literature CD £10.00
Line Design Clipart CD £10.00 - Famous Faces Clipart CD for Line Design £10.00
PD Software Library catalogue available on my website <http://homepages.tesco.net/dilwyn.jones/index.html>

Want To Know More?

Visit the Launchpad web page at:
<http://homepages.tesco.net/dilwyn.jones/launchpad/launchpad.html>
from where you can download a free trial version, limited only by the number of programs you can set up to run on it – more than enough to try out Launchpad.

Launchpad is available (price £20.00) from either the author:

Dilwyn Jones, 41 Bro Emrys, Tal-y-bont, Bangor, Gwynedd, LL57 3YT, U.K.

email: dilwyn.jones@tesco.net

Payment in Pounds Sterling only

or from:

Q-CELT COMPUTING,

The Falconry, Glenmacnass, Glendalough, Co. Wicklow, Ireland

email: darrenb@esatlink.com (Payment in Pounds Sterling or Euros)

Gee Graphics! (on the QL ?) – part 39

H. L. Schaaf

```
100 REMark GaussLegendreQuad_bas
110 REMark for GG#39
120 REMark HL Schaaf June 24, 2004
130 :
140 WTV : PAPER 0 : INK 7 : CSIZE 1,0 :CLS
150 menu : max_min : graph_it
160 places = 0 : old_places = 0 : old_area = 0
170 n = 2
180 start = DATE
190 REPEAT gr
200 PRINT #0,n,
210 GRULE(n)
220 area = get_area
230 dif = ABS(area - old_area)
240 old_area = area
250 IF dif: places = INT(ABS(LOG10(ABS(area/dif))))
260 IF NOT(dif) : places = 8
270 IF places>old_places THEN
280 PRINT n;' points'!!!area,places;
290 PRINT ' places'!!!!DATE-start;' elapsed seconds'
300 old_places = places
310 END IF
320 n = n + n
330 IF NOT(dif) : EXIT gr
340 IF (ABS((area/dif)) > 1E7) OR(places >= 7) : EXIT gr
350 END REPEAT gr
360 :
370 PRINT#0\\, , 'Touch [space bar] for another, Q to quit'
380 IF INKEY$(-1)=='q' : STOP
390 RUN
400 :
410 DEFINE PROCEDURE GRULE(n)
420 REMark Ref: P.J. Davis & P. Rabinowitz
430 REMark "Methods of numerical integration"
440 REMark Academic Press, New York, 1975
450 LOCAL I, M, E1, T, U, V, H
460 REMark and X0, PKM1, PK, T1, PKP1, DEN, D1
470 REMark and DPN, D2PN, D3PN, D4PN, DP, FX
480 M=(n+1)/2
490 DIM AB(M) : DIM W(M)
500 E1=n*(n+1)
510 FOR I=1 TO M
520 T=(4*I-1)*PI/(4*n+2)
530 X0=(1-(1-(1/n)))/(8*n*n))*COS(T)
540 PKM1=1
550 PK=X0
560 FOR K=2 TO n
570 T1=X0*PK
580 PKP1=T1-PKM1-(T1-PKM1)/K+T1
590 PKM1=PK
600 PK=PKP1
610 END FOR K
620 DEN=1-X0*X0
630 D1=n*(PKM1-X0*PK)
640 DPN=D1/DEN
650 D2PN=(2*X0*DPN-E1*PK)/DEN
660 D3PN=(4*X0*D2PN+(2-E1)*DPN)/DEN
670 D4PN=(6*X0*D3PN+(6-E1)*D2PN)/DEN
680 U=PK/DPN
690 V=D2PN/DPN
700 H=-U*(1+.5*U*(V+U*(V*V-U*D3PN/(3*DPN))))
710 P=PK+H*(DPN+.5*H*(D2PN+H/3*(D3PN+.25*H*D4PN)))
720 DP=DPN+H*(D2PN+.5*H*(D3PN+H*D4PN/3))
730 H=H-P/DP
740 AB(I)=X0+H
750 FX=D1-H*E1*(PK+.5*H*(DPN+H/3*(D2PN+.25*H*(D3PN+.2*H*D4PN))))
```

GRULE on the QL

Gaussian quadrature with Legendre polynomial coefficients.

What's quadrature? In some ways it is like trying to square the circle. In integral calculus we were shown that by dividing the area under a curve into little rectangular strips and then summing their individual areas we could approximate the area under the curve. As we made finer and finer divisions with more and more strips we approached a limit, known as the integral. I think quadrature is an approximation, whereas the integral is presumed to be exact.

We can approximate the integral by dividing the curve into strips of equal width, do the summation, and check the answer. Then we can keep doubling the number of strips, doing the sums, and comparing the answers. Eventually the answers should settle down closer and closer to the "true" value. Several clever refinements that use equally spaced sample points have been developed including Simpson's rule and Romberg methods.

Gauss took samples at UNEQUAL spacings based on Legendre's differential polynomial equations. The samples were taken at the roots (zero crossing) of the polynomial and the strip widths were from the adjacent peak to trough on the polynomial curve. This gave much better answers with fewer samples. The only hassle was trying to solve the Legendre polynomials and finding the roots, peaks, and troughs. A Fortran program named

GRULE written in 1954 at the 'dawn of the computer age' made it possible to solve Legendre functions for high order polynomials.

A nice long story about 'the dawn' and GRULE can be found at:

www.siam.org/siamnews/general/dawn/htm

I simply converted GRULE to QL's S*Basic. The real roots of the Legendre polynomial all lie between -1 and +1, and to use the Legendre coefficients they have to be mapped to the curve in question by shifting and scaling. The y-values are calculated at the corresponding abscissa and multiplied by the distance (weight) between peak and trough for each sample point to get the area of each quadrilateral strip. The quads are summed up and that total rescaled to find the approximate area. Again as more and more samples are taken the answer gets closer and closer to the 'true' area under the curve.

A nice explanation of all this can be found on the web at:

<http://mathews.ecs.fullerton.edu/n2003/GaussianQuadMod.html>

A lovely graphic example is given at:

math.fullerton.edu/mathews/a2001/Animations/Quadrature/GaussLegendre/GaussLegendreaa.html

The listing

GaussLegendreQuad_bas will give you a feel for how it works with known answers for a few textbook examples. Choice 7 allows DIY Lame curves, if the exponents are equal you can crosscheck the

```

760      W(I)=2*(1-AB(I)*AB(I))/(FX*FX)
770      END FOR I
780      IF ((2*M)>n): AB(M)=0
790 END DEFine GRULE
800 :
810 DEFine FuNction f(X)
820 SELEct ON example
830 = 1: RETurn SIN(X) : REMark Shoup
840 = 2: RETurn (X*X) * COS(X) : REMark Kuo (3/2)
850 = 3: RETurn (1/EXP((X)))*(SIN(8*(ABS(X)^(2/3))))+1 :
      REMark Mathews
860 = 4: RETurn SQRt(ABS(1-(X*X))) : REMark semicircle
870 = 5: RETurn (2/SQRt(PI))/EXP((X*X)) : REMark Scheid 15.31
880 = 6: RETurn SQRt(ABS(1-X*X)):REMark quadrant
890 = 7: RETurn Lameval(X) : REMark Lame curve
900 END SELEct
910 END DEFine :REMark f(X)
920 :
930 DEFine FuNction a_b(L_Ab)
940 REMark maps Legendre Abcissa (from -1 to +1) to curve (from
      a to b )
950 RETurn (a +((b-a)/2)*(L_Ab+1))
960 END DEFine :REMark a_b(L_Ab)
970 :
980 DEFine FuNction get_area
990 LOCAl I,E
1000 area = 0
1010 FOR I = 1 TO n/2
1020   tp = a_b(AB(I))
1030   tm = a_b(-AB(I))
1040   quadshigh = f(tp) + f(tm)
1050   quadarea = quadshigh * W(I)
1060   area = area + quadarea
1070 END FOR I
1080 areahigh = f(a_b(0))
1090 IF ((n/2)-INT(n/2)) AND (n>1) THEN
1100   quadarea = areahigh * W(I+1)
1110   area = area + quadarea
1120 END IF
1130 IF n = 1: quadarea = areahigh * 2: area = area + quadarea
1140 area = area * (b-a)/2
1150 RETurn area
1160 END DEFine :REMark get_area
1170 :
1180 DEFine PROCedure menu
1190 REPEat choice
1200 CLS : CLS #0
1210 PRINT 'choose from following examples:'
1220 PRINT ' 1 - Shoup'
1230 PRINT ' 2 - Kuo'
1240 PRINT ' 3 - Mathews'
1250 PRINT ' 4 - semicircle'
1260 PRINT ' 5 - Scheid'
1270 PRINT ' 6 - quadrant'
1280 PRINT ' 7 - Lame curve'
1290 examp$ = INKEY$(-1)
1300 IF examp$ INSTR('1234567') :EXIT choice
1310 END REPEat choice
1320 example = examp$
1330 SELEct ON example
1340 =1: a = 0 : b = PI : ans = 2 : name$ = 'Shoup'
1350 =2: a = 0 : b = 3 : ans = -4.952115 : name$ = 'Kuo'
1360 =3: a = 0 : b = 2 : ans = 2.01628 : name$ = 'Mathews'
1370 =4: a =-1 : b = 1 : ans = PI/2 : name$ = 'semicircle'
1380 =5: a = 0 : b = 4 : ans = 1 : name$ = 'Scheid'
1390 =6: a = 0 : b = 1 : ans = PI/4 : name$ = 'quadrant'
1400 =7: get_input :a=0 : b = Lame_a : name$ = Lamename$
1410 END SELEct
1420 END DEFine menu
1430 :
1440 DEFine PROCedure get_input
1450 PRINT 'examples only for 1st quadrant ! '

```

```

1460 PRINT 'please enter values for Lame curve'
1470 CSIZE#0,0,0 : INK#0,7
1480 prn$="Please ENTER a positive real number for " : CLS #0
1490 REMark values for width, height, and exponents 1 and 2
1500 INPUT#0; prn$ & "half width ? ";Lame_a
1510 INPUT#0; prn$ & "half height ? ";Lame_b
1520 INPUT#0; prn$ & "width exponent ? ";Lame_e1
1530 INPUT#0; prn$ & "height exponent ? ";Lame_e2
1540 Lamename$="Lame curve (x/"&Lame_a&" )^"&Lame_e1
1550 Lamename$=Lamename$ & '+ (y/'&Lame_b&') ^"&Lame_e2&' = 1'
1560 CLS #0
1570 END DEFine get_input
1580 :
1590 DEFine PROCEDURE max_min
1600 LOCAL I, y
1610 miny = 1E6 : maxy = -1E6
1620 FOR I = a TO b STEP (b-a)/256
1630 y = f(I)
1640 IF y<miny : miny = y
1650 IF y>maxy : maxy = y
1660 END FOR I
1670 END DEFine max_min
1680 :
1690 DEFine PROCEDURE graph_it
1700 SCALE 2*(maxy-miny),a-.1,miny-.1:CLS
1710 PRINT name$,,
1720 IF example<>7 : PRINT ans
1730 PRINT
1740 LINE 0,0 TO b,0
1750 LINE 0,0 TO 0,maxy
1760 POINT a,f(a)
1770 FOR I = a TO b STEP (b-a)/128
1780 LINE TO I,f(I)
1790 END FOR I
1800 END DEFine
1810 :
1820 DEFine FuNction Lameval(X)
1830 RETURN Lame_b*((1- ((X/Lame_a)^Lame_e1))^(1/Lame_e2))
1840 END DEFine
1850 :
1860 REMark end of listing GaussLegendreQuad_bas

```

answers by using Allard from GG#38. You can change the starting n on line 170 and the rate at which n grows on line 320; try n + 1, or n + n + 1, etc. Most of the total elapsed time is used in filling in the AB and W GRULE values. Getting the area only takes a second or so. After seeing how this worked on several curves, I call the GRULE only once in a program with n = 256 and feel that the answers are probably good to at least 6 or 7 significant digits. Enjoy.

Next time, maybe we will have a go at the length of compound Lame curves?

As requested in the past by several readers, we have printed the listing on the outer areas of the page to make it easier for our readers who like to type the listing in from the opened magazine - Editor.

A Bit at a Time

Geoff Wicks

A couple of Eindhoven shows ago, Wolfgang Uhlig approached me like a fierce schoolmaster carrying his cane. "You boy!", he snapped, "You have not done your homework!" As a punishment he made me stand in the corner for the rest of the show with a dunce's cap on my head.

I made that first paragraph up. Readers who know Wolfgang will know that he is far too polite and kind to behave like that, but he did give me a friendly rebuke for not reacting to his work on using the new QL colours. I had to plead guilty. A few technical problems and too much involvement in QL politics had prevented me from looking at the GD2 colours. I had tried a few simple experiments, but had been unable to get over the first hurdle, but more of that later.

One advantage of being a late-starter is that you can learn from the work of other people. In parti-

cular, Wolfgang has written several programs to display GD2 colours and has described how you to use these in EasyPtr. And in Scotland George Gwilt has produced software to make it easier for EasyPtr users to transfer their programs to TPTR.

George's program EPTR2WDA interests me because it converts menus produced by EasyMenu into a form suitable for use in TPTR.

In the very early days of his development of TPTR I did some beta testing for George in which I failed in attempts to reproduce my EasyPtr screens in TPTR format, but George has now successfully automated the process. I have tried his program on several Just Words! menus and it has worked perfectly in each case.

TPTR holds its screen information in a series of data statements and George has written another program EDIT_WDA which converts these data statements into a list of Information Windows, Loose Items and so on. This list includes their size, colours and other information, and is far

more detailed than comparable lists produced by EasyMenu. EasyPtr users may find this facility worthwhile even if they have no intention of using TPTR.

I find George's work fascinating because it is merging two, possibly rival, technologies for writing pointer programs. EasyPtr has helped many of us overcome the fear of writing for the pointer environment, but it does have some weaknesses. For example it is not possible to resize screens on the fly in EasyPtr programs. Is George challenging us to widen our pointer programming horizons?

This is a lengthy introduction to the purpose of this article. Having been suitably chastised by Wolfgang Uhlig, I have been busy designing Just Words! GD2 house style. There are many programs, including some Just Words! classics, that have been around for years that can now be put in a nice new jacket. Here is a little Just Words! experience to help you along.

HELP! I'm drowning!

Starting to program in the new colours is a daunting undertaking. There is a mass of technical information few of us understand. Should you use COLOUR_QL, COLOUR_NATIVE, COLOUR_PAL or COLOUR_24? If you have difficulty in choosing between a peach-coloured or a plum-coloured

shirt think of the problems of choosing between 65535 QL colours. To make matters worse in COLOUR_24 mode you are not allowed to use normal numbers but have to enter everything in Hexadecimal. Then there is a whole new sprite technology. They now come in 8 bits or 16 bits and can be multiple, dynamic, extended and chained.

When you are drowning in all this information, there are few lifeguards around to help you, and even they talk in a language you do not understand.

Keep calm! There are a few life jackets and other aids out there. The secret of learning to swim in this environment is to take it slowly and learn a bit at a time. Forget about sprites for the moment and just concentrate on the colours. Even the colours can be broken down into small manageable chunks. Once you have taken the first plunge you will find that swimming in the GD2 sea is quite fun. You quickly learn new techniques and enjoy the challenge of exploring new territory.

Your Lifejackets

You need a number of tools to help you survive in the GD2 sea.

Firstly, make sure you have your manuals by you. In my case I needed the EasyPtr manual to re-



JUST WORDS!

The big one...

QL2004
EINDHOVEN
16th October

For more information visit our website:

<http://members.lycos.co.uk/geoffwicks/ql2004.htm>

or use the Just Words! contact information.

Geoff Wicks, 56 Peveril Crescent, West Hallam, Derbyshire DE7 6ND, U.K.

Tel: +44 (0)115 - 930 3713

email: gwicks@beeb.net

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

mind myself of the precise syntax of some of the keywords and to learn one or two new ones.

Secondly, make use of a good text editor. Even, if like me, you do not normally use an editor to write SuperBasic programs, you will still find an editor worthwhile. It has search and replace functions that will save you a lot of time when you are modifying SuperBasic.

Thirdly, have Wolfgang Uhlig's article "Colours with EasyMenu" (QL Today volume 8, issue 5 page 8) by your side. Even if you do not use EasyPtr, this article contains some valuable information on the GD2 colours.

Finally, have some program available to help you choose colours. If you want to be ambitious and try the full monty straight away look at Wolfgang's programs. They can be downloaded from www.uhlich.nl/ql. However, I think it is better to work initially in the 256 colours of COLOUR_PAL mode. Type in and compile the short program under the section Palette maps on page 5 of the SMSQ/E manual display section. To compile the program in QLiberator you will have to give the repeat routine in lines 150 to 190 a name and change the hexadecimal numbers in lines 170 and 180 to decimal. (\$D0=208 and \$D8=216).

Decision Time

Before you start work you have to take a few decisions. In my case it was choosing between the George Gwilt way and the Wolfgang Uhlig way of using GD2 colours. I gave serious consideration to trying TPTR, but it would have involved a steep learning curve and experiments with colours that would have delayed the introduction of a Just Words! GD2 house style.

Another decision you have to make is the colours you want to use. Here again there are two possibilities. Some people want their programs to be totally integrated into the pointer environment. In other words they use the colours and style the user has programmed for use in his own version of the Window Manager. Other people prefer to use their own colours and want their programs to look the same on all QL platforms.

There are no prizes for guessing what the Just Words! choice was. I have put a lot of effort into developing a distinct house style, and my priority was the same colours on all platforms. For this reason I opted to use COLOUR_PAL, although there is a problem with this I shall explain later. Once you have developed your distinctive style using these 256 colours, you can easily "upgrade" to the higher colour definitions with the help of Wolfgang Uhlig's programs.

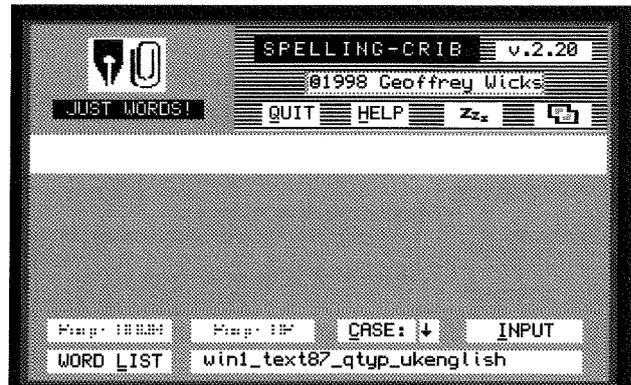
A Technical Bit

This next bit is for EasyPtr users. It is a practical application of the techniques Wolfgang described in his article. If you never use EasyPtr, you may still wish to read this section, but if not it will not be a major disaster if you skip to the next section.

First, load the menu you wish to modify into EasyMenu. If the OUTLINE has a border remove this, because you have to completely overwrite the outline. Next click on Info window and then Set up and define a new window of exactly the same size as the Outline. For working purposes this will become a new "pseudo-outline". The "pseudo-outline" must be the first window in the information window list. You do this by right clicking on the window number and changing this to 1.

We need a pseudo-outline because you cannot modify the colours in the true outline. This was the first hurdle I was unable to overcome when I first experimented with the GD2 colours. Wolfgang Uhlig was more clever and realised you could just overwrite it with an Information Window.

Now save the new menu, load it into appman_obj and modify the _cde file to use the redefined menu.



Before we go any further have a look at Fig. 1, and in particular the top left hand corner. This contains 3 information windows, each of which can be modified using the command

```
MWINDOW[chan%]num!
```

In the old version of Spelling-Crib:

Information Window num=1 was the background window with a border.

Information Window num=2 was the Just Words! logo

Information Window num=3 was the name "Just Words!"

However our pseudo-outline is now information window number 1, and the three information win-

dows in the left hand corner have now become numbers 2, 3 and 4 respectively. You should now load your basic program into the text editor, search for all occurrences of MWINDOW and add 1 to each of the window numbers.

We are now ready to start recolouring, and we begin with the pseudo-outline. This will be grey paper with a purple border. After drawing the menu, in say channel 5, using the MDRAW command, we insert a single line of basic:

```
MWINDOW #5,1! : PAPER #5,light_grey : BORDER  
#5,2,purple : CLS #5
```

Next we want to recolour the top left hand corner window. This is a small window in a different shade of grey, but with the same purple border. Here the basic is:

```
MWINDOW #5,2! : PAPER #5,grey : BORDER  
#5,1,purple : CLS #5
```

Information window number 3 is the Just Words! logo, which is a sprite. This has been overwritten by the previous command, and we rewrite it with the command:

```
MIWDRAW #5,3!
```

Information window number 4 has also been overwritten. This is a window with the text "JUST WORDS!" as its information object. In this case it is best to remove this information object in Easy-Menu and add the words in the basic program.

Thus a line:

```
MWINDOW #5,4! : PAPER #5,dark_grey : INK  
#5,light_grey : CLS #5 : PRINT #5, " JUST  
WORDS! "
```

In this way you can recolour and redraw each of your information windows. After you have done this, you may find you have to return to Easy-Menu to make a few pixel adjustments to the size or position of some windows.

The redrawing will also overwrite your loose items, and you can restore these by the simple command MLIDRAW #5.

It is possible to recolour practically everything in a screen designed by EasyPtr. There is one big exception and that is the individual Application Window Menu Items. Two Just Words! programs use these and I had to compromise by using the QL colours that fitted best with the GD colours (in this case a cyan-white stipple).

There has been a lot of discussion about an upgrade to EasyPtr to handle the new colours more easily. There is in limited circulation an upgraded version which does this, including application window menu items. Unfortunately this

version cannot be given a wider distribution because of copyright technical problems. It is hoped that these problems can eventually be resolved.

Some Colour Hints

You will probably have noticed that the PAPER and INK commands I used in the last section had colours and not numbers. It is a good policy to define your colours early in the program. Thus various shades of grey from light to dark could be defined as:

```
light_grey=13 : grey=12 : dark_grey=10
```

If you then want to make subtle changes to the shades of grey all you have to do is change the colour number. I tried about 10 different shades for the border before I had the one I wanted.

Another advantage of this is that if I had changed my mind and wanted a green border rather than a purple one, it would have been easier to have loaded the SuperBasic into a text editor and search for all occurrences of "purple", than to search for a number.

In practice I made several radical changes to the colours used in the Just Words! GD2 house style during its development. To use the 256 palette colours you have to have the command COLOUR_PAL at the start of your program. I removed this command, because instead I decided to use what are known as the "Window Manager" or "System Colours". To do this you have to use some new keywords. Instead of INK, PAPER and BORDER, you use the keywords WM_LINK, WM_PAPER and WM_BORDER.

There are 65535 system colours and the first 256 of these are the standard QL colours. The second 256 are the palette colours, so I had to add 256 to my previous colour values:

```
light_grey=269 : grey=268 : dark_grey=266
```

The use of system colours gives a lot of advantages. If necessary, and sometimes it is necessary with EasyPtr, it is easy to mix QL and palette colours. It is also possible to experiment with 3D effects and for advanced use there are 32,768 RGB colours.

You soon discover how to make creative use of the GD2 colours. I like working with greys because they blend well with most other colours. One criticism people sometimes make of Just Words! programs is that it can be difficult to distinguish between loose items and information windows. By colouring the latter a light grey, I can give more emphasis to the former and make them easier to find.

Now You are on Your Own

There is a big snag with using the 256 palette colours. The colours in the palette can be redefined by the user and you can be certain some clever clogs will do just that and spoil your carefully designed colour scheme.

Tony Tebby saw this problem arising and suggested that if you modify the colour palette you should modify the high values so that programmers could use the low values and know they will appear the same on all platforms.

However, there is a better way round this problem. Basically the 256 colours are a subset of 24 bit colours or in the case of the system colours a subset of the 32,768 15 bit RGB colours. If this is now getting too technical, don't worry. What it means is that the 256 palette colours were selected from 32,768 RGB colours originally. If you want to stop clever clogs spoiling your colour scheme you use the original colour instead of the palette colour.

Thus my line of basic in which I define the colours now becomes:

```
light_grey=61307 : grey=56022 : dark_grey=46509
```

Unfortunately, this is where it becomes really nasty, because at the moment there is no easy way to determine which palette colour corresponds with which RGB colour. I am preparing a table to do this and I should be finished by the time you are reading this, but I suspect it will be too long to print in QL Today. If you need the information, then contact me directly.

If you want to look at the RGB system colours, then you should run Wolfgang Uhlig's program QCP_obj. Alternatively you can type in the simple basic program I used when experimenting with the colours. This is not as sophisticated as Wolfgang's program, but it does help you understand how the colours are made up from the Red, Green and Blue components.

Why am I an old Fusspot?

Why am I being so fussy about using the correct colours, and and say you should only use the RGB colour that corresponds to the palette colour?

The reason is that the QL has three main platforms, native hardware, QPC and the Q40/60. Each of these platforms handles colour in a slightly different way. In particular, native hardware suffers from severe memory restrictions, and cannot handle colours in the same detail that the other two platforms can. If you were to print all the red shades from dark to light you would find

it would be a continuous strip in QPC and the Q40/60, but have steps in the Aurora.

To illustrate this run QCP_obj in QPC high colour mode and then in QPC 8 bit mode, which mimics as far as possible the colour handling of the Aurora. In effect the Aurora misses out some colours. When I was trying to find a suitable purple colour, I found several that were perfect in QPC, but which appeared as too blue or too red in the Aurora. This was a big problem in some programs where a mauve window has a purple border. It was only when I knew RGB colour 56128 corresponded to colour 83 on the palette that I had a shade of purple that looked the same on both systems.

Icons

When I had finished recolouring the program, it was time to think about icons or sprites. I started by doing some research on QL websites, and looked at the work of Wolfgang Lenerz, Per Witte and Tim Swenson. Wolfgang has written a program to transfer PC graphics in BMP 24 bit format into QL graphics; Per has written programs to display icons; and Tim has put a collection of sprites on his web page.

Their web addresses (in order) are:

www.scp-paulet-lenerz.com/14mljkl24/wolf/download

<http://knoware.mysite.freemove.com>

www.geocities.com/svenqhj/myfree.html

When I went to Tim's site I struck oil with the minimum of prospecting. On his site are a series of mode 16 sprites designed by François van Emelen, and these fitted perfectly with the Just Words! colour scheme.

There were two snags. The sprites were larger than the ones I was already using, so some new adjustments had to be made to the menu in EasyMenu. In addition, although you can load 16 bit sprites into EasyMenu and save them in a _men file, these sprites are not displayed when you load that saved _men file a second time into EasyMenu. Each time you have to reload the sprites in EasyMenu. This is a little bit annoying, but still worth the effort.

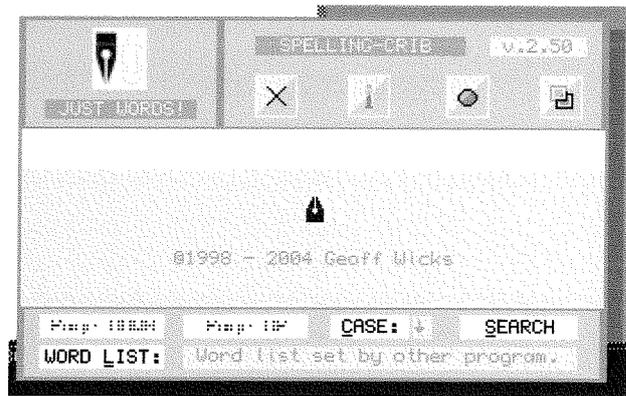
François' sprites gave me a new problem. Alongside these beauties my low resolution Just Words! logo looked cheap and dowdy. Fortunately I had the logo in PC _gif format. I loaded it into the Paint program that comes with Windows, smoothed out the jagged edges, did a little bit of recolouring and saved it as a _bmp 24 bit file. I then used Wolfgang Lenerz's program to transfer this to QL format.

The Challenge

It takes a lot of effort to take the first steps in using the GD2 colours, but once you start your enthusiasm and knowledge quickly grows. Inevitably you come across problems for which you have to find solutions. If you need help with these, then come to the QL2004 show. Wolfgang Uhlig will be leading a colour workshop and there will be many people present who have worked with the new colours.

You do not have to program in the pointer environment to use the new colours in your programs. Even a simple SuperBasic program can be recoloured without too much difficulty. Why not look though your own work and see which of your programs could be given a make-over.

The Result



```
100 WINDOW 250,150,0,0
110 BORDER 2,248
120 PAPER 0 : INK 7 : CLS
130 red=0 : green=0 : blue=0
140 print_colours
150 REPEAT change
160 e$=INKEY$(-1)
170 e=CODE(e$)
180 SElect ON e
190 = 114 : REMark r key - increase red
200 red=red+1
210 = 82 : REMark Shift + r key - decrease red
220 red=red-1
230 = 103 : REMark g key - increase green
240 green=green+1
250 = 71 : REMark Shift + g key - decrease green
260 green=green-1
270 = 98 : REMark b key - increase blue
280 blue=blue+1
290 = 66 : REMark shift + b key - decrease blue
300 blue=blue-1
310 = 208 : REMark Up cursor key - increase brightness
320 red=red+1 : green=green+1 : blue=blue+1
330 = 216 : REMark Down cursor key : decrease brightness
340 red=red-1 : green=green-1 : blue=blue-1
350 END SElect
360 IF red>30 : red=31
370 IF green>30 : green=31
380 IF blue>30 : blue=31
390 IF red<1 : red=0
400 IF green<1 : green=0
410 IF blue<1 : blue=0
420 print_colours
430 END REPEAT change
440 :
450 DEFine PROCedure print_colours
460 AT 3,23 : PRINT "Red: ";red;" "
470 AT 5,23 : PRINT "Green: ";green;" "
480 AT 7,23 : PRINT "Blue: ";blue;" "
490 Colour=32768 + 1024*red + 32*green + blue
500 AT 9,23 : PRINT "Colour: ";Colour
510 WM_BLOCK 100,100,15,15,Colour
520 END DEFine
```

The Simple BASIC Program

Q-Emulator Review

Dilwyn Jones

We are well accustomed to mention of the QPC2 emulator in the QL press. Rather less well known is Daniele Terdina's QemuLator. This was originally written in 1994/95 for the Apple Mac operating system, and was then ported to Windows three years later.

QemuLator is shareware. That means you can try it out before you actually buy it, then if you like it you can register the software with the author, paying one of two levels of registration, dependent on the level of performance you want from the emulator.

For US\$ 22.50 you get an emulator which very closely emulates an original QL - it runs at pretty much the same speed as an original QL and seeks to emulate as much of the hardware as it can - 68008 processor, memory, keyboard, video, sound and even a 256K memory expansion. You can also access QL floppy disks and the PC hard disk.

For \$40 you get what is called an "expanded QL" registered version, which includes all of the above and speed limited only by your PC's performance. It can also use up to 16MB of RAM, accelerated video driver and level 2 filing systems (use sub-directories). You also get access to QXLWIN filing systems (read and write, but not format or create them).

At the time of writing, 25 and 40 dollars translated roughly to these European currencies:

25\$	EUR 20	£13.70
40\$	EUR 32,80	£21.90

Assuming you download the emulator from the author's website, it consists of a download of about 450KB. This is the unregistered demonstration version, limited in speed and functionality. Registering gets you a code which you enter into QemuLator to give it whatever level of functionality you chose to pay for. The unregistered version is quite slow, about 30% of the speed of a standard QL, so not very good at all for serious use, but does allow you to try out the emulator to some degree. In addition to the speed restriction, you cannot access serial and parallel ports and there is no sound or mouse emulation and no ramdisk.

There is also an option to get a temporary registration code, valid for a given time. I was given this type of code to allow me up to a month to evaluate the emulator and write this review.

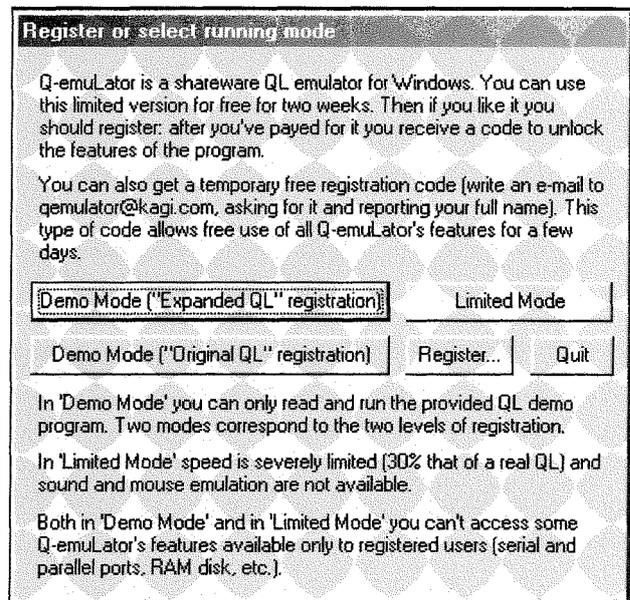


Figure 1. The opening screen of the unregistered version

Once you have received the registration code, you enter it into a registration screen, as shown below.



Figure 2. Entering registration details

Figure 3 shows the opening screen. This consists of the commands bar across the top, the QL emulator window in the middle and the drive slots at the bottom.



Figure 3. The opening screen

The command menus across the top let you configure, start and stop the emulator. There is also the facility on the File menu to screen dump the emulator's display to the PC's printer.

Before using the emulator, you need to install a QL ROM. The emulator can use any Sinclair ROM or Minerva. A recent version of Minerva is included, and instructions are given on how to save a ROM image from a standard Sinclair QL. ROM images are also available from the internet where copyright status permits. I've used the emulator with version JS and Minerva ROMs with no problems. The most recent version of QemuLator which I used for this review was blessed with full facilities (like the \$40 version) and that includes the facility to run the Gold Card version of SMSQ/E, although it must be a fairly recent version of SMSQ/E. To start SMSQ/E you simply boot up into QDOS in the normal way then start the SMSQ_GOLD file from wherever it's stored on your PC. This is pretty much like running SMSQ/E on a QL where the machine boots into QDOS then reboots into SMSQ/E. The author emphasises that SMSQ/E compatibility is to some extent experimental, but seemed to work pretty well in the time I was using it (I used SMSQ/E v3.06 for Gold Card).

In addition to installing a QL ROM image, you can specify the equivalent of a plug in EPROM, in a similar way to that in which a QL may have an EPROM plugged into the EPROM slot at the back. The likeliest option here is to use a Toolkit 2 ROM image, saved from a standard QL EPROM, or if you prefer you could probably use the disk-based reconfigurable Toolkit 2. If you are using SMSQ/E, that includes the equivalent of Toolkit 2 of course. There is in fact the facility to use two plug in ROM images if required - I only used it for the Toolkit 2 ROM image which worked without any problem.

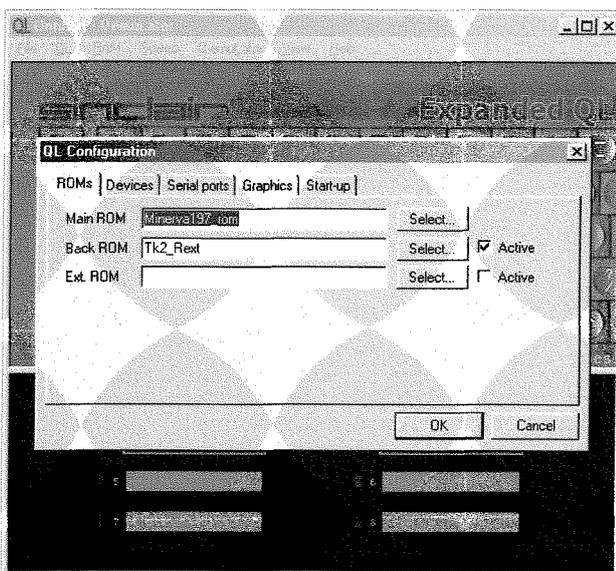


Figure 4. Specifying ROM to use.

You should also set up the devices such as serial ports, parallel port, screen driver details (whether using a QL window, full screen or fast screen, all have their advantages and disadvantages depending on how you prefer to use the PC), whether or not to use a level 2 filing system and whether or not the TCP/IP facilities (if used, a basic internet facility is available using the Lynx text browser, for example). When setting the screen, there is no facility for high resolution display or high colour modes, QemuLator is an emulator of a traditional Sinclair QL after all. If you just want to run your existing QL software, go for QemuLator, if you want to dabble with high resolution screens, GD2 colours etc, pay more and go for QPC2.

The QL menu lets you specify among other things whether or not QemuLator is to use Accelerated QDOS Graphics Routines. For maximum compatibility with QDOS, these can be turned off if required.

The RAM menu lets you specify how much RAM to use. Anything from a standard 128K QL up to a massive 16MB of memory may be specified here. This is a very useful facility, because you can set up your QL's memory to work like a Trump Card, Gold Card, or an older 256K or 512K memory expander, useful for running older programs or testing software.

The Speed menu lets you set the speed at which the emulator runs. While this may seem a bit of an odd facility at first, the speed at which the emulator runs on modern fast PCs means that some programs like older QL games run much too fast if the emulator is allowed to run at maximum speed (which on my 1.8GHz PC for example may be 100 times as fast as the original Sinclair QL). You can choose between roughly QL speed, Gold Card speed or maximum possible speed.

The QemuLator menu lets you register or un-register QemuLator. I'm not quite sure why you'd want to un-register it, unless you wanted to pass on an unregistered demo version perhaps.

The View menu lets you turn on or off the status bar at the bottom of the device slots section. This is normally used for messages, but removing it expands the viewing space available for the device slots, which is two slots with status bar on and four with status bar off. Dragging the outline further down expands it to 8 slots maximum.

Which brings us to the device slots, a feature of QemuLator which I found rather quirky. Forget floppy disks, microdrives and hard disks in the traditional sense, these are all purpose slots which can be any device you want them to be.

Think of them as MDV1_, MDV2_ etc up to MDV8_, but although they have the name MDV they can be defined as being floppy disks or hard disks or a mixture. To see how this works, look at Figure 5 below.

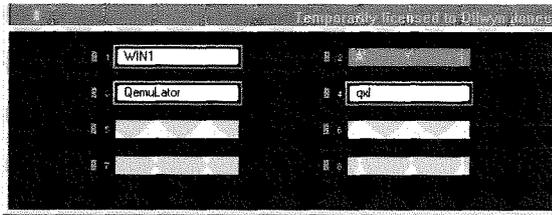


Figure 5. The 8 MDV slots

This gives you up to eight devices, MDV1_ to MDV8_. If you enable the FLP and WIN alias options in the configuration dialogues, these also respond to WIN1_ to WIN8_ or FLP1_ to FLP8_. What you need to do is tell the emulator exactly what device corresponds to drive 1. For example, if you want QemuLator to access your QXLWIN filing system as drive 1, you could specify this as C:\QXLWIN. If you prefer it to be floppy drive 1, you could make it equal to A: and if you want drive 2 to refer to the PC's main hard disk, you could make it equal to C:\ and to access the PC's CD_ROM drive as device 4 you could make slot 4 equivalent to drive D:\. Folders on the PC drives can be specified here too, for example, you could make slot 8 equivalent to C:\qemuLator\ to access files in QemuLator's own folder on the hard disk. Selecting a folder is done in pretty much the usual Windows way, as shown in Figure 6. Simply right click on the drive slot (may be done even when the 'QL' is running, real-time changes!). In Windows terminology, a 'Folder' is what we would know as a directory or sub-directory.

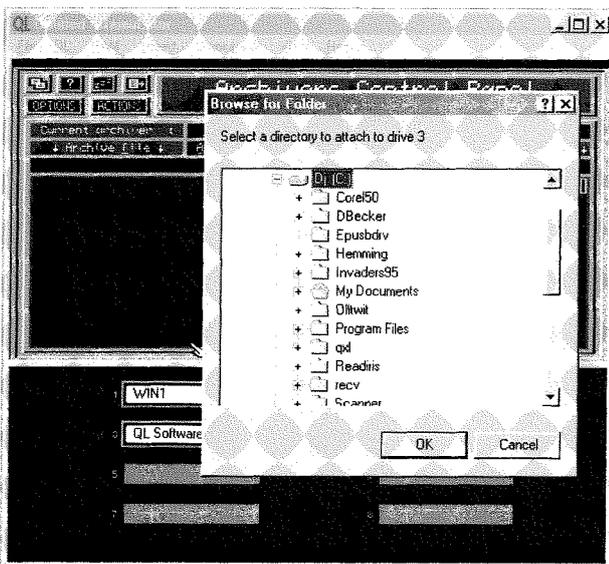


Figure 6. Browsing to a folder

In Figure 5 I have assigned device 1 to be my main QXLWIN (C:\QXLWIN). This is the one I normally use with QPC2 and is quite large - about half a gigabyte - and QemuLator copes with this just fine. It is only possible for either QemuLator or QPC2 to access this at any one time, whichever claimed it first gets access rights to it, and the other cannot access it if both are running at the same time, which is perfectly possible. Device 2 is the PC's floppy drive A:\ and the next two slots are set to directories called C:\qemuLator and c:\qxl in the Windows part of the hard disk (just any old folders, they could have been just about anything on the PC's hard disk).

If I then DIR MDV1_ or DIR FLP1_ or DIR WIN1_ I get the same results, a list of files in the QXLWIN, what other QL systems would treat as WIN1_. If I do DIR MDV2_ or DIR WIN2_ or DIR FLP2_ I get a list of files on floppy drive 1. This shows how powerful, yet potentially confusing, this can be. It is really useful to be able to make any device into any drive or path you want like this, and you can change them in real time too to run software configured for a particular drive. The downside is the scope for error or misunderstandings. My setup allows me to DIR FLP2_ yet get a list of files on FLP1_. By and large, I tend to just store my files on WIN1_ and run everything from there and use drive 2 to copy to and from floppy disk only, then use drive 3 to copy to and from the Windows side of the hard disk. It all sounds horribly complex, but you quickly get used to it and of course there is the visual element of always being able to see what the definition for each device slot is. Where this comes into its own is the ability to quickly and easily run software configured for a particular drive. For example, if the program is configured to run from MDV1_, simply copy it onto a floppy disk and set drive 1 to be floppy disk A:\. The program now runs from disk, but as far as the software is concerned it's running from a device called MDV.

The devices facility is rather difficult to explain clearly in print and sounds horrendously and needlessly complex. In practice, it's so simple anyone can get used to it pretty quickly I'd have thought.

Some file utilities are provided with QemuLator to help with the task of moving software to a special format to run from the Windows side of the PC's hard disk. Normally, if you copy an executable program to a PC's hard disk, the job data-space part of a file header is lost - in effect it is no longer an executable QL program. QemuLator

works around this by providing a program which adds 30 bytes to a file, storing the necessary extra information. When files are copied between QemuLator and Windows using QemuLator itself, this "translation" is done automatically. The file utility called QLtoWIN_exe (a QL program) can be used to manually do the translation on another QL system (e.g. a "real" QL or another emulator). There is also a complementary program to do the reverse process, of stripping the extra bytes of information from a QL program which is already in QemuLator format. These are fairly simple programs which just ask you to type in a couple of filenames. A third file utility called MDVtoWIN_exe translates QL files stored on QL copy protected microdrives such as old QL games into the QemuLator Windows file format.

It is also possible to indicate if sound emulation should be on or not (needs Windows DirectX sound facilities) and whether or not a joystick is available. This behaves like a QL joystick plugged into the CTL1 joystick port on a standard QL. Sound is via the standard QL BEEP command, played through the PC's speakers, as long as DirectX sound is available.

To start the QL emulation, you can either go into the 'QL' menu and click on start, or simply click on the QL picture in the main window. It boots up with a QL-style screen presenting you with the usual F1 or F2 (or F3 and F4 for Minerva) and Sinclair copyright notice etc. You can even configure it to 'auto-start' and bypass this, a facility which I think was originally implemented to allow a QL operating as a bulletin board system (BBS) to restart automatically after a crash or power cut.

By default, the emulator operates without mouse or pointer environment if you are running a QDOS ROM image (including Minerva). If you wish to run the pointer environment, you need the usual PTR_GEN, WMAN and HOT_REXT files to be installed. Standard QL ones are fine, there is no need to patch or modify anything. To use a mouse, the author supplies a file called Mac-Mouse11 which is RESPRed just after pointer environment is installed. This makes the PC's mouse operate as a QL style mouse to control pointer driven applications. While this is generally very good, on my system at least pressing the right mouse button did not equate to pressing ENTER on the keyboard which I have always found useful on other QL systems. If you are using SMSQ/E for Gold Card, the SMSQ/E

mouse driver is used for mouse control, so you do not need the MacMouse driver software. In fact, for SMSQ/E purposes, QemuLator now emulates some of the Gold Card hardware, just enough to support SMSQ/E. For me, this was a big plus point of using QemuLator, you can choose whether it operates as a Sinclair QDOS, Minerva, or SMSQ/E system, whichever you prefer or need at the time.

Once started, QemuLator performs pretty well like you'd expect any QL system to operate. Software generally seems unaware that it is running on an emulated system, although a very few older programs which use obscure hardware dependent facilities may have problems. In general, QemuLator makes a pretty good job of emulating a real QL and very few programs have problems running. Indeed, I'd say you are likely to run into more problems with games running too fast on a decent fast PC than to have problems not being able to run a program. You may well get ROM dependent problems as you would on a standard QL, programs which only run on a version JM or JS ROM, or programs which do not get on with Minerva. Do not blame QemuLator in this case, the problem lies with the software if it won't even run on a different ROM version on a standard QL! Software compatibility can indeed be described as very good, given my experience of emulators of other computers.

One thing I noticed while starting QemuLator with a Minerva 1.97 ROM is that BASIC extensions which announce their presence upon installation seem to send their output to the top left of a console channel rather than the usual channel 0 at the bottom of the screen. Turbo Toolkit is one such victim - normally when you LRESPR this toolkit it displays a copyright notice in channel #0. It does not seem to have any harmful effect, just an example of a slight difference in how things work.

A slight hiccup became obvious when trying to use Archive under pointer environment. Normally, the pointer environment saves and restores the screen display of programs as you CTRL C out and back in. I suffered a number of problems where the display was only partly restored. Usually, the top left corner of the display was missing - see the example screen dump in Figure 7. And when using Quill, the red ruler dots and numbers would sometimes just go missing. This only seemed to happen with the original QL programs - Xchange did not seem to be affected.

In all other respects, the programs seemed to work normally.

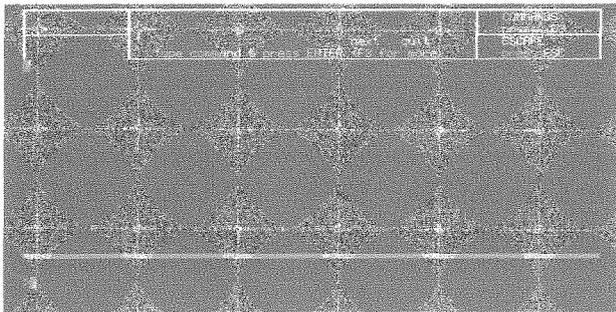


Figure 7. Screen save/restore problem.

The first real problem I ran into was with my printer. Since I have owned this PC, I've put the printer onto a USB port. QPC2 handles this by allowing the PAR port of the QL to be linked to the USB printer. QemuLator cannot do this in the current version at least, printers can only be on the parallel port. This is nothing to do with the printer itself, it has Epson control code sets and in fact has both USB and parallel ports, it is simply that QemuLator does not know about USB printer interfacing. This is one area where I'd urge the author to implement a new facility, to allow QL PAR to equate to PC USB ports (in the same way as QPC2 is able to do) given the increasing popularity of USB printers.

The keyboard includes some useful new key combinations, for example, the Backspace key on a Windows keyboard corresponds to the CTRL LEFT arrow key on a QL, and the PC's DELETE key corresponds to CTRL RIGHT on a QL keyboard. Page Up and Down correspond to SHIFT UP and DOWN on a QL keyboard and the SCROLL LOCK key corresponds to CTRL F5 on a QL.

There are some less obvious special keys too. F12 and SHIFT F12 toggle between Window display and Full screen or fast screen display. The INSERT key pastes up to 1KB of text from the Windows clipboard into the QL type-ahead keyboard buffer, which is now up to 1KB in size rather than the mere 7 bytes on an original QL if I remember correctly. This makes it quite easy to transfer short bits of text from a Windows program to a QL program. CUT or COPY the text in the Windows program, go into the QL emulator and press the INSERT key which will to all intents and purposes type the text into the waiting QL program via the keyboard buffer.

QemuLator includes a RAMdisk driver. RAMdisks are like floppy disks built into the computer's memory, but they lose their content when the computer is switched off or the emulator is stopped. They make useful temporary file storage space. This RAMdisk does not include level 2 directory handling facility but is adequate for most purposes. The RAMdisk may be dynamic or static, that is, you can force it to be a given size or have it vary its size automatically as required. If you prefer, you can switch off QemuLator's RAMdisk and load other RAMdisk software if you have a better type of RAMdisk which you prefer to use.

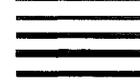
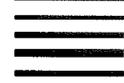
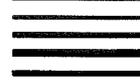
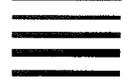
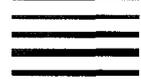
Fed up of having to set your QL clock's date and time every time you turn on the QL? Forget it, QemuLator takes its time settings from the PC which remembers the time even if switched off. You can even change the QL clock while the 'QL' is running and have it revert to the correct time next time it starts.

Up to 4 serial ports are available in the expanded registration version (2 in other versions). These serial ports can use higher baudrates than a standard QL serial port (up to 256,000 baud if supported by the PC's serial ports compared to up to 19,200 baud for a standard QL). The serial port drivers do not support the TRA character translate facilities built into some later QL ROM versions and into SMSQ/E, although the usual parity settings, flow control, CR and EOF settings are available. I mentioned the parallel port above, in fact up to 2 parallel ports can be used if available on the PC.

Daniele has implemented SCK_ and TCP_ device drivers to allow use of the Windows TCP/IP stack. Only a limited subset of the socket functions are available, but it is enough to run the QDOS version of the Lynx Internet browser and FTP TCP/IP is on by default, although it can be disabled in the emulator configuration if required. I'm afraid I did not get time to tinker with this facility before producing the review. The only other "QL" system I know of with any degree of inbuilt TCP/IP support is Richard Zidlicky's uQLx emulator.

Configuration is one of the emulator's strong points. You can configure a large number of settings and store these settings so that the emulator always starts with your preferred settings. You can save several configuration files and open one of your choice later if required - all this is well documented in the manual. The manual is a

Q U A N T A



Independent QL Users Group

World-wide Membership is by subscription only,
offering the following benefits:

Bimonthly 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 Twenty First Year

Further details from the Membership Secretary

**John Gilpin, 181, Urmston Lane
Stretford, Manchester, M32 9EH**

Tel. +44 (0) 161 865 2872

or

Visit the Quanta Web Site

<http://www.quanta.uni.cc>

E-mail: [quanta_membership @ uk2. net](mailto:quanta_membership@uk2.net)

HTML file, which is viewed in an internet browser like the Internet Explorer browser supplied with Windows. This is a brief manual describing mainly the QemuLator specific facilities and not a QL manual as such.

Some excellent QL graphics demos are supplied with QemuLator, including some impressive animations, it is well worth running these. Set device slot 1 to point to QemuLator's "QL Software\QL Demo" folder on the PC's hard disk and run the BOOT program then select one of the 4 supplied graphics demos and prepare to be amazed - you probably never knew the QL could do graphics like this!

Speed

Since an emulator has to translate the 68008 machine code instructions into equivalents for the processor used in PCs, it is bound to be slower than a true hardware based computer. But since QemuLator and QPC are designed to be run on PCs, and modern PCs have very fast processors, the emulators themselves are likely to be much faster than an original QL. Indeed, on a modern fast PC, both QemuLator and QPC2 may be expected to be even faster than a hardware QXL emulator for example. I ran some standard benchmark programs on my machine (a 1.8GHz Duron processor system) to see how well the emulator performed compared to a standard QL, a Gold Card QL and to the nearest competitor, QPC2.

QemuLator performed rather well in these tests, being much faster than even a Gold Card QL. On most tests, it was very similar to or very slightly faster than QPC2 in terms of speed, although the differences between QDOS SuperBASIC and SMSQ/E SBASIC means that BASIC programs inevitably show up as being much faster running in SBASIC on QPC2 than on a QDOS ROM in QemuLator. SBASIC performs more like compiled SuperBASIC in terms of speed. If you run SMSQ/E for Gold Card on a full speed QemuLator, SBASIC programs run at close to the same speed on both emulators. If you are not really interested in the cold hard figures below, do not worry, QemuLator handles and feels like an extremely efficient QL, more than fast enough for just about any purpose!

I used 3 'standard' tests to compare systems, and one very simple BASIC program of my own.

The 'QSI' is Eros Forenzi's QL Speed Index software. This runs as an application in Archive and is intended to work as a measure of speed

somewhat independent of the different speeds you might expect by simply running a program in BASIC on the various systems.

Bogomips and Dhrystone figures are applications used on other computer systems such as Windows and Unix as a relative speed measurement index system. I used Thierry Godefroy's QL versions of these programs to produce the figures.

Finally, for a very simple test of BASIC I applied the following short program:

```
LET b = 0
LET iterations = 1E6
FOR a = 1 TO iterations
    LET b = b+1
END FOR a
```

and timed how long this took to run. In all cases, I tried to run the tests several times and take average readings where relevant, to increase accuracy. Whatever your feelings about benchmarks and speed tests in general, rest assured, QemuLator and QPC2 are both fast enough not to worry too much about speed tests!

1. QSI Tests

This was run on QL Archive v2.3. A standard QL would take 800 seconds to run the test, while a 16MHz Gold Card should take 123 seconds and for reference a 20MHz QXL card on a 386 or 486 PC (SMSQ 2.16) takes 29 seconds.

This test seemed to give almost identical results on all the ROMs used in QemuLator, and when running SMSQ/E for the Gold Card. This was tested with the emulator set to run at full speed - obviously if the emulator was set to run at QL or Gold Card speed it would be restricted to the same results as those types of QL systems). Although I was going to include a second set of results based on different system setups such as with or without Toolkit 2 or pointer environment, it didn't seem to make much difference, the results were so close to each other that it might simply have been down to differences in the timing system, so I have simply printed an 'average' figure.

It took 7.2 seconds to run the test on my Duron 1800 PC.

QSI QL index = 111.11x times (speed relative to standard QL)

QSI GC Index = 17.08x times (speed relative to a 16MHz Gold Card QL)

Computer speed (Gold Card = 100) QSI Speed = 1708.33

This it can be seen for the purposes of running the test on my PC that QemuLator is shown to be about 111 times as fast as an original Sinclair QL, or about 17 times as fast as a Gold Card QL. I then proceeded to run the test on QPC2 on the same computer, with very similar results:

Took 7.5 seconds to run the test on my Duron 1800 PC.

QSI QL Index = 106.67x times (speed relative to standard QL)

QSI GC Index = 16.40x times (speed relative to a 16MHz Gold Card QL)

Computer speed (Gold Card = 100) QSI Speed = 1640

Thus you can see that with optimal full speed settings, QemuLator is very slightly faster than QPC2, although as always running speed tests of this kind may inadvertently favour one system slightly over the other.

2. BOGOMIPS

I then tried the Bogomips test program on both emulators and came up with the following results:

QPC2: 24.53 Bogomips

QemuLator full speed setting: 30.89 Bogomips

QemuLator Gold Card speed setting: 1.45 Bog.

QemuLator standard QL speed setting: 0.21 Bog.

3. DHRYSTONE

I then applied the Dhrystone test, the results reflecting the number of microseconds for one run through Dhrystone code. As the QL timers are only accurate to 1/50th second, the tests need to be run a large number of times and an average result taken for greater accuracy.

(uS = Microseconds for one run through Dhrystone)

	QPC2	Qem Full Speed	Gold Card	Standard QL
uS	: 33.3	32.8	491.5	280.6
Dhrystones per second	: 30,080	30,506	2034.4	491.5
VAX MIPS rating	: 17.092	17.363	1.158	0.160

So much for standard tests. I then tried some simple BASIC programs. The above tests do not reflect the increased efficiency of SBASIC compared to SuperBASIC. A simple FOR/NEXT loop as described above gave the following results:

QemuLator ROM	Full speed	Gold Card	Standard QL speed
JS	53 seconds	770 seconds	5300 seconds
Minerva 1.97	47 seconds	666 seconds	4600 seconds
SMSQ-Gold	8 seconds	112 seconds	817 seconds
QPC2	5 seconds	n/a	n/a

As if to prove that benchmarks are not always the be all and end all of speed comparisons, this

seems to show that SBASIC on QPC2 is nearly twice as fast as SBASIC on QemuLator. In fact, on the face of it, QPC2 would seem to be more than 10 times as fast as QemuLator running a JS ROM!

It would appear on the whole that in raw program execution terms, QemuLator with optimal speed settings is very slightly faster than QPC2, with results varying slightly depending on the test being applied. The difference between the two is negligible apart from the speed of running a BASIC program.

Pros and Cons of QemuLator

Pros:

- * Can run QDOS or SMSQ/E (QPC2 can only run SMSQ/E)
- * TCP/IP support built in
- * Choice of running speeds and memory configuration
- * Cheaper than QPC2
- * Support for most QL ROM versions
- * Supports QL second screen with Minerva ROM
- * Choice of QXLWIN filing system or storage of QL programs on PC hard disk
- * Extensively configurable
- * Sound and joystick emulation available
- * Accurately emulates a real QL
- * Being a true QL emulator, it may prove easier to set up and use than QPC2, although this may be a matter of opinion.

Cons:

- * No colour drivers
- * No high resolution display modes
 - * Unusual drive handling, although very flexible
 - * RAMdisks do not use level 2 filing system, although external RAMdisk software can be used
 - * Cannot create QXLWIN file system yet - you have to use one copied from an existing QXLWIN file or created by an external program
- * Cannot use USB printer ports.
- * Does not handle DD disks under Windows 98 very well (unless you fiddle in the BIOS settings), although this is a Windows problem, not a QemuLator problem. Handles QDOS format disks pretty effortlessly, though.
- * Disk formatting not possible unless the version of Windows is NT4 or later, XP or 2000.

This review was based on QemuLator v2.3.1 for Windows. It requires Windows 95 or later on a 486 or better processor. DirectX is required for sound and full screen graphics (built into some versions of Windows). The PC used for the test was a 1.8GHz Duron processor system, with 256MB RAM and a 40GB hard disk, running Windows 98 SE.

Should I buy QPC2 or QemuLator?

There are pros and cons for both emulators, so the answer is not a simple one or other. If you wish to dabble with SMSQ/E, high resolution graphics or high colour graphics, go for QPC2 of course since QemuLator is not designed for these facilities. QemuLator is much cheaper to buy than QPC2 and if you merely wish to run existing QL programs at moderate cost, QemuLator is cheaper to buy yet provides good compatibility with a standard QL and gives you the

option of using whichever QL ROM version you prefer. If you are a 'user' rather than a tinkerer and you prefer traditional QDOS, you will need to go for QemuLator since QPC2 cannot run QDOS, only QemuLator can do that. QPC2 provides QL devices such as FLP,WIN and RAM in the sense we are used to them (i.e. DIR FLP1_ !), rather than the slightly quirky system used in QemuLator.

In the end, I would say that if you want all the bells and whistles and all the latest features and are prepared to pay the higher price, go for QPC2. If you are after a cheaper option, or simply wish to run QDOS on a PC so that you can run older programs, or you are into "retro computing", you should go for QemuLator (but do go for the \$40 version, it's well worth the extra cash) as it's designed to be a QL we all know and love as opposed to QPC2 which goes more towards the "Super QL" end of the market.

JPEG Viewers for SMSQ/E

Timothy Swenson

Now that I have my Q40 back up and running, I want to be able to view JPEG images that I take with my digital camera. The Q40 is very capable of displaying good quality color images, esp. when in 1024x512 mode.

My criteria for using a image viewer is that it must handle JPEG images and GD2 colors. From looking around the Net, I found three viewers that fit my criteria, Photon, PhotoQL, and PQIV.

Photon

Photon is a simple image viewer written by Dave Westbury. Photon is executed from the command line like this:

```
exec photon;"win1_jpg_vulpeak_jpg"
```

It will then read the JPEG image, convert it, and display it on the QL. If you want to see another picture, you have to EXEC it again. Once you execute it, a small button pops up on the screen with the name of the file. As the loading and converting process proceeds, a bar moves across the button showing the progress. When the picture is displayed, you can hit the ESC, SPACE, or ENTER to go back to SBASIC. If you want the image to stay in the background, then hitting the W key will return to SBASIC but leave the image displayed. If the image is larger than the Q40 screen, then only the upper left hand part of the

image will be displayed. You can then scroll through the image. F4 will redraw the image.

There are command line options to affect how Photon displays the image. A \d option configures dithering, \m affects the mode (4 or 8), \r affects scaling, and \y produces a black and white image. I don't know exactly what scaling the \r option does. I tried it on a 1024x512 image and the image appears larger than the screen. Without the \r option, the image fills the screen. The \y option for black and white works fine. Photon can handle JPEG, PIC, and GIF images.

PhotoQL

PhotoQL is a viewer written by Roberto Porro. I tested version 0.19. Once executed, PhotoQL comes up with a blank window. You have the typical PE loose items; resize, move, sleep, and ESC. If you hit the ESC item, a menu pops up that lets you Quit, Cancel or Load. If you hit Load, a menu pops up that lets you select the image you want to load. The menu allows you to navigate through the different file systems (WIN1_, FLP1_, etc). Once you select the image, PhotoQL will load the image and display it. Like Photon, PhotoQL also allows the file name to be included when the program is executed.

I noticed that it took a fair bit of time to load the image and it was displayed taller than it should have been. An image that was full screen when displayed with Photon, was shown smaller and taller with PhotoQL.

I tried to load a fairly large image and PhotoQL gave a "not found" error after it tried to load the image. The documentation does not mention any limitation on the size of the image it can load. I have reported this problem to Roberto, so it might be sorted out soon.

PhotoQL can handle the following formats: TARGA, BMPPCX, GIF, JPEG, PNG, PIC, and TIFF.

PQIV

PQIV is an image viewer written by Claus Graf, one of the designers of the Q40. I tested version 0.17. Although the documentation is fairly short, the program can do a fair bit.

Once executed, PQIV comes up with a Window with a number of loose items. They are Sleep, Wake, Move, Resize, File, Image, and Help. The File menu includes Open, Close, Save, Print, Quit. The Image menu includes "Rotate 90 degrees Right", "4:3 Aspect Ratio", "Shrink to 2:1", "Extend to 2:1".

By default the display window is fairly small. Using the Move and Resize items, you can make the window a lot larger. To load an image, click on File, then Open. A menu pops up letting you move around the file system looking for the image you want to load. Once the image is loaded, if it does not fit inside the Window, there will be scroll bars on the bottom or right hand side of the Window. These are used for scrolling around the image. With this feature, I was able to load some fairly large (about 700K) JPEG images with no problem.

Once you have an image loaded, the Image item has a menu of image manipulation commands. The most useful is the "Rotate 90 degrees Right" option. If you take photographs in Portrait instead of the normal Landscape mode (taller than wider), the image will look sideways on the screen. Using the rotate command, you can turn the image upright. There are other aspect ratio commands, but I don't know exactly what they do, even if I've tried them. They might make more sense and be more useful to someone else.

Once you've manipulated an image, you can save the image. The other two programs were just viewers and had no "save" capabilities.

The Help item has a small menu for Version and Help. Neither of these menu selections did anything. I don't know if the program was expecting to find a certain help file or not, but nothing is mentioned in the documentation and no error is shown.

Analysis

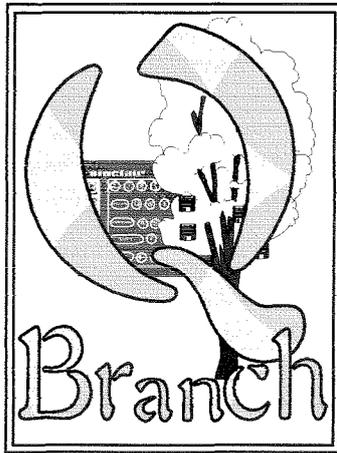
Of the three programs, I think PQIV is the most useful and the one that I will be using. Photon was useful, but I did not like it not using an PE window for display. You could not set the image aside into a button. Plus, I prefer to have a menu where I can walk my way around the file system to pick the image instead of having to type it out each time.

I found PhotoQL to be slow and not able to load the larger images I needed to view. These images were taken directly from my digital camera. The image it could load, I had to downsize to 1024x512 pixels.

Granted it took a bit to copy these large images via floppy to the Q40, but with access to a CD-ROM, I may be using the Q40 more for reviewing my photo library.

The following picture shows the output of a 640x480 picture made with a mobile phone camera on QPC (the "communications" setup at the US show).





Text 87

£ 79.00

Typset 94 £ 29.00

Fountext 94 £ 39.00

2488 drivers £ 29.00

Epson ESC/P2 drivers
£ 26.00

Text 87

High Colour

Patch

Available Now

£9.00

PROGRAMMING

QD 2003	£ 49.00
QD + QBasic	£ 63.00
QD + Qliberator + QBasic	£ 104.00
Qliberator	£ 50.00
Master Spy v 3.3	£ 30.00
QPTR	£ 32.00
Easyptr pt 1 & 2 (together)	£ 30.00
Easyptr pt 3 (C library)	£ 14.00
QMake	£ 18.00
QMon / JMon	£ 22.00
Basic Linker	£ 22.00
DISA 3	£ 34.00
QMenu	£ 16.00

Another High Colour Update

We are now able to release a new version of DATAdesign. This features a new, High Colour interface and icons and some other improvements. I am also hoping that a new version will be available soon which will be able to print using the QD/QSpread print filters since this will make it much easier to produce hard copy.

Now that the summer is here many of you will be out enjoying the sun but we hope to see you at the many events planned for this autumn.

Available Now!

High Colour Drivers for Gold /Super Gold Card SMSQ/E

(Aurora only - Send old master disk as proof of purchase)

£25.00

QL Keyboard Membranes

Last Few!

£ 17.50 ea + P&P

These programs have been updated to High Colour versions:-

QMake

WinEd

FiFi II

QPAC 1

QPAC 2

QSUP

<<<<< SuQcess >>>>>

Please send old master disk and £ 1.60 to cover postage and copying fee.

DATAdesign

High Colour Update

£8.00

UTILITIES

FiFi 2	£ 21.00
QSup	£ 30.00
QSpread 2003	£ 51.00
Cueshell 2	£ 15.00
Qload / Qref	£ 15.00
Disk Mate 5	£ 16.50
QPAC 1	£ 22.00
QPAC 2	£ 42.00
QTYP 2	£ 31.00
QLQ	£ 30.00

These Programs are paid Upgrades:

QSpread 2001 to 2003	£ 10.50
QD 98 to 2003	£ 10.50
QPC2	-
update from v3.xx	Free
update from v2	£ 13.90
update from v1	£ 34.00

Return Master Disk With Order

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) 1273-430501
Mobile +44 (0) 7836-745501
email qbranch@qbranch.demon.co.uk web : http://www.qbranch.demon.co.uk



20 LOCKS HILL, PORTSLADE,
E. SUSSEX. BN41 2LB. UK.

ProWesS

ProWesS (now free !)	£ 1.60
DATAdesign	£ 20.00
DATAdesign update	£ 8.00
Fontutils	£ 28.00
File Search	£ 11.00
PFlist	£ 11.00
Dilwyn's Fontpack	£ Call
LINEdesign v 2.16	£ 22.00
PWfile	£ 17.50

Paragraph

¶ The ProWesS word processor
Demo version £ 1.50 + postage
Full Registered version £ 18.00

Hardware

Second user items are sometimes available from us.
Call for details.

QXL II	£ 40.00*
Recycled superHermes	£ 65.00 *
R'cycld SuperGold Card	£ 110.00 *
Recycled Gold Card	£ 45.00 *
Recycled Aurora	£ 65.00*
Qubide	£ 45.00*
Qubide Rom Updates	£ 8.00
Qplane	£ 5.00
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. Call for details

Gold Card / Atari / QXL Version **SMSQ/E** Aurora High Colour
Version 3.06 out now! version available now
Now Only **£ 32.00** upgrade **£ 21.00**

QPC 2 v3.13 - full colour version!

Special offer!
Get Cueshell for only £15.00 with any copy of SMSQ/E
or QPC 2 upgrade!

Now Only **£ 65.00**

Q Branch Programs

The Knight Safe 3	£ 35.00	Q - Route v1.08C	£ 25.00
upgrades from previous versions	£ 5.00	Route finding programme	
Q - Count	£ 25.00	The Fractal Collection	£ 35.00
Pointer driven home accounting			

Post and Packing is included with Software Prices
Unless Otherwise Stated.



We can accept payment by Visa, Mastercard and Switch. We also accept
cheques drawn on a UK Bank in Sterling.
You can also pay directly into our bank in Euros.
Barclays Bank Acc# 79016888 sort code :20-71-02 Name :QBranch)



Programming in Assembler - Part 10 (2)

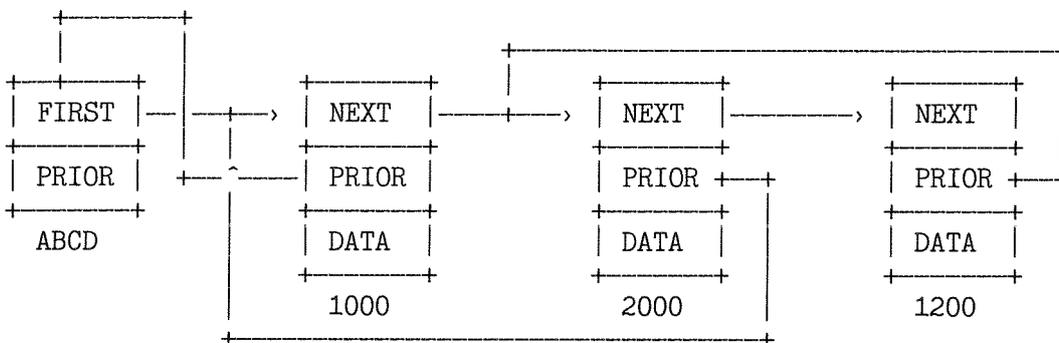
Doubly Linked Lists

Norman Dunbar

If we change the structure of our nodes and add a PRIOR pointer to each node and to the root node as well, we can store the address of both nodes neighbouring our current one, as shown below. Here is the node structure:

NEXT	4 bytes at the start of each node hold the address of the NEXT node in the list.
PRIOR	The next 4 bytes hold the address of the node PRIOR to this one in the list.
DATA	The data bytes can be any size and hold any data we like.

Our conceptual model of the doubly linked list is as follows:



Adding Nodes

Adding a new node is still simple. Having allocated a node on the heap, you set it's PRIOR pointer to zero and it's NEXT to the current address held in the FIRST pointer – almost identical to the single linked list code above.

```

Prior      equ 4                ; Offset to PRIOR pointer in a node

Prelude    lea FIRST,a0         ; Pointer to storage of first node address
           lea NewEntry,a1      ; Address of new node
  
```

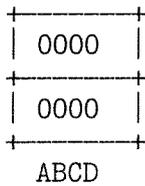
Then adding a new node to a doubly linked list is as simple as this:

```

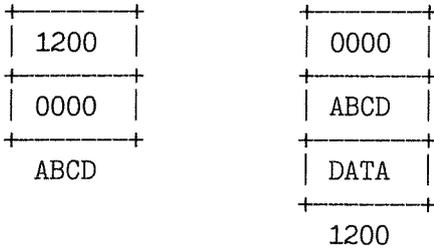
AddNode    cmpa.l a0,a1         ; Don't add the root node to its own list
           bra.s AddExit       ; Quietly exit if attempted
           move.l (a0),(a1)     ; Save current first node in new node's NEXT area
           move.l a0,(a1)       ; No PRIOR node for this node
           move.l a1,(a0)       ; Store address of new node in FIRST storage area
AddExit    rts
  
```

As with single linked lists there is nothing to it. The new node is always added at the start of the list, so the value in FIRST always points to the latest node added. The first non root node in a doubly linked list has no real PRIOR node, so that part of the newly added node is simply set to point back at the root node.

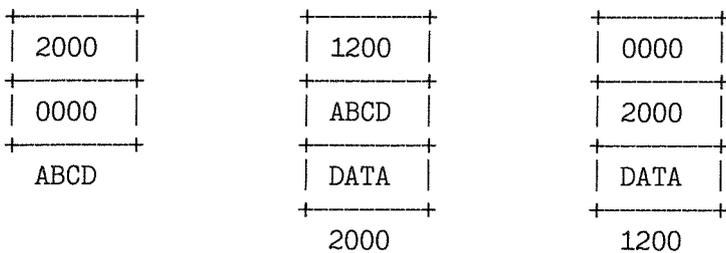
Building up the linked list above in stages, we would start like this:



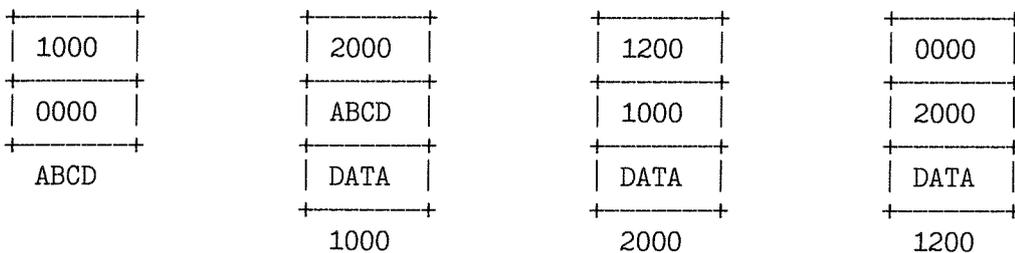
Then add the node at address 1200 as our first node, giving this:



When we add the node at 2000 as the second node, we get the following:



And finally, adding the node at address 1000 gives us this:



You can see how each node points onward to the NEXT one and also backwards to the PRIOR one. The last node has no NEXT nodes, so it has its NEXT pointers set to zero to indicate the end of the list.

Deleting Nodes

Deleting a node is much simpler. There is no need to scan the entire list from the start looking for the node prior to the one you want to delete because you already know it's address by following the PRIOR pointer backwards from the node to be deleted.

Here's the pseudo code to delete a node. We assume, as before, that A0.L is the root node pointer and A1.L is the node to be deleted.

If the two addresses are equal, we cannot allow the root node to be deleted, exit with an error.

If the address in the root node's NEXT pointer is zero then we still have an empty list so the value in A1 must be illegal. Exit with an error.

Fetch the deleted node's PRIOR pointer. Every real node in a list will have a valid PRIOR pointer, only the root node has no prior pointer and we don't allow that to be deleted.

Store the NEXT pointer from the deleted node into the NEXT pointer of the prior node.

Fetch the deleted node's NEXT pointer, which might be zero if we are deleting the final node in the list.

If it is not zero, store the deleted node's PRIOR pointer in the next node's PRIOR pointer.

Exit without error.

That's the pseudo code, here's the real code to do all of the above.

```
Prior      equ 4                ; Offset to PRIOR pointer in each node

Prelude    lea FIRST,a0         ; Pointer to storage of first node address
           lea OldNode,a1      ; Address of node to delete
           moveq #ERR_BP,d0    ; Bad Parameter = trying to delete the root node

Now, here's the actual code to find and remove the requested node.

DelNode    cmpa.l a0,a1        ; Don't allow the root node to be deleted
           beq.s DelExit      ; Bale out with error if attempted

           cmp.l #0,(a0)      ; Do we actually have a list ?
           beq.s DelExit      ; Yes, node not found, exit with error

           move.l Prior(a1),a0 ; Fetch the deleted node's PRIOR pointer
           move.l (a1),(a0)    ; Store the deleted node's NEXT pointer in the NEXT
                               ; pointer of the PRIOR node

           move.l (a1),a0      ; Fetch the deleted node's NEXT pointer
           move.l Prior(a1),Prior(a0) ; Store the deleted node's PRIOR pointer in
the
                               ; next node's PRIOR pointer.

DelFound   move.l (a1),(a0)    ; PRIOR node's NEXT = the deleted node's NEXT
address
           moveq #0,d0        ; Node deleted ok
           bra.s DelExit      ; Bale out with no errors

DelNext    move.l (a0),a0      ; A0 now holds the NEXT node in the list
           bra.s DelNode      ; Go around again

DelExit    tst.l d0           ; Set zero flag for success, unset for error
           rts
```

Finding Nodes

As with single linked lists, you may have a need to locate a specific node by its contents, so you need a generic FindNode routine again. The fact that the list has two pointers this time around is the only difference, so the code is basically the same as for the single linked list above.

The only difference is the offset to the data part of the node needs to be set to 8 bytes instead of 4, so while the code for the FindNode remains the same, the code for the Compare routine needs to be changed to the following to account for the extra pointer.

As before, the comparison routine must preserve A0, A1, A2 and D0 or it will all go horribly wrong.

```
NData    equ 8                ; Offset from start of node to the data part

Compare  cmp.l NData(a3),(a2) ; Is the data in the node = the value we want?
        rts                  ; Exit with Z set if so, unset otherwise.
```

Again, if an attempt is made to 'find' the root node, then it will fail.

A Better Mousetrap

Because the code for the linked list find routine is identical except for the offset in the compare routine you can use the same code. If you modify it so that it passes the offset over to the compare routine in a spare register, say D1.W for example, then you can even have the same compare routine for both single and doubly linked lists, as shown below.

```
Compare  cmp.l 0(a3,d1.w),(a2) ; Is the data in the node = the value we want?
        rts                  ; Exit with Z set if so, unset otherwise.
```

Another method, much loved in the internals of Microsoft Windows, is to store a word holding the offset to the data at the start of each node. This would remove the need for the D1.W register to be passed into the comparison routine as a parameter as it could easily extract the data from the node itself, as follows:

```
Compare  move.w (a3),d1        ; Fetch the offset to the data from the node
        cmp.l 0(a3,d1.w),(a2) ; Is the data in the node = the value we want?
        rts                  ; Exit with Z set if so, unset otherwise.
```

The drawback to this method is the redundancy of the data - each and every node has to have the first two bytes set to the offset to the data plus 2 for the size of the offset word itself. Two extra bytes per node may be the difference between getting all the data in memory or not. It is, of course, always up to you. If you decide to go down this route, don't forget to amend the code to add, find and delete nodes to take the extra two bytes into consideration when manipulating the pointers to NEXT and PRIOR. Your root node must also reflect these changes and have an offset word added to its own structure.

You might see a need to build a couple of comparison routines to compare two nodes rather than the example above where a node is being compared with a value. On the other hand, you could simply write one routine to compare two nodes and when looking for a value, create a dummy node and use that in the comparison routine. That way, you don't need separate routines to compare values and nodes.

Double Trouble

The problem with a doubly linked list is that while adding nodes is just as simple as before, but deleting them could be problematical. If you are passed the address of a node which is not in the list, how do you tell that it is or is not a valid node address? You can end up trashing bytes of memory almost at random as you start changing the NEXT and PRIOR pointers for two areas of memory which may not be in your list.

My solution is to use a flag word or long word after the two list pointers in each node and when passed in a node address to delete, compare this value in the flag to see if it is correct before attempting to delete the node. As ever, I leave this 'as an exercise for the reader' to modify the code above to carry out said checks.

Sorting Lists

The best way to sort a list is not to have to sort it at all. When you store a node in the list, store it in the correct place according to its value. A doubly linked list is used here again as you will need to go NEXT until you hit a value greater than the one you want to insert, then you might need to go PRIOR to insert it in the correct location. I'll leave you to figure out that little exercise.

There is another way, which involves TREES of nodes rather than lists. A tree is simply a linked list which has a LEFT, RIGHT and UP pointer in each node.

With a tree, the nodes are not in a long line, but they are off to the LEFT and RIGHT of the root node. Each node may itself have children to the LEFT and RIGHT as well as a parent found by following the UP pointer.

Unfortunately, trees are a bit beyond my skills at the moment. I remember doing them in college and learning all the different ways to navigate them, but I cannot remember much about them nowadays – it's been over 30 years since I last considered them.

Remember those arrays?

Way back at the start of this article, I mentioned arrays and their problems. Well, combining an

array with a linked list could be useful – but remember, the array is limited by the fact that you have to pre-define the number of entries.

Bearing this in mind, you could allocate an array of, say 1000 entries of 4 bytes each. Each entry in the array holds the address of an individual node, not the actual data stored there. Our address book system of 100 byte strings (not much of an address book I admit!) will now only need about 4Kb plus 102 bytes per used node – including the string length word for each entry. Using a plain array it would need almost 102Kb even for a blank address book.

Now you have compromised on memory needs as you don't allocate the space required to store your data until you need to, and you do allocate a much smaller amount to hold the 'contents table'. As you create new nodes, add their address to the array. You can still use the single or double linked lists if you wish, but there is no need. The array holds all the locations of each node in the order that they were created and you can navigate forwards, backwards and even access nodes at random using this method because the formula to find a given node is once more usable.

Have fun trying that out!

That's it for linked lists until next time when we'll get the real meat of the demo routines sorted out.

DBAS2HTML

Wolfgang Uhlig

I have written a small utility which makes a HTML-file out of a DBAS-database. It is a SBASIC-file so that you can edit the included CSS-style attributes and /or the layout of the

table in order to meet your personal wishes. This program is really useful if you want to print your database with formatting options you don't have on the QL. If you want to show the resulting HTML-file with the ProForma HTML-viewer, you must delete the CSS-style attributes which cannot be handled by ProForma. You can download it from my website: www.uhlich.nl/ql/util.html

```
100 REMark ***** small routine to reformat a DBAS-database into HTML
110 REMark ***** formatting with CSS2 styles, thus not compatible with
120 REMark ***** Proforma HTML viewer unless you delete the style block
130 REMark ***** 2002 - Wolfgang Uhlig - w.uhlig@tiscali.nl
140 :
150 open #3,con_
160 out1 #3
170 db$=file_select$("which database?","", "_dbs")
180 close #3
190 Open_data #3,db$
```

JOCHEN MERZ SOFTWARE

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

J-M-S is moving!

But the only visible change is a new street name! Please note, that the address change will become active on October, 1st., 2004. It will be

Kaiser-Wilhelm-Str. 302

and even the postcode remains the same! It is possible that during the move, phone and internet facilities will stop working for a day or two, but the phone numbers will move with me.

Other changes? Well, except from the fact that after this issue is out, all time will go into renovation etc. until the next issue will be due (fortunately delayed to be ready at QL 2004 - what a coincidence). But the new office will be larger, and hopefully less hot in the summer ... laser printers and human brains don't work well at temperatures over 36° - I can tell from experience!

I will be at the Berchtesgaden show beginning of October, of course!

I have taken the time to update the SMSQ Homepage and hope to do it more often - I know, I said this before, but I'll try even harder. Why not have a look:

<http://SMSQ.J-M-S.COM>

```

200 fieldcount=FLNUM(#3)
210 reccount=COUNT(#3)
220 :
230 REMark ***** read field names
240 DIM n$(fieldcount-1,20)
250 FOR i%=0 TO fieldcount-1
260 n$(i%)=FLNAME(#3,i%+1)
270 END FOR i%
280 :
290 REMark ***** write array to HTML-format
300 :
310 open #4,con_:out1 #4
320 name$=read_string$("Filename",,"Please give only a name for the\HTML-file, no
path, no suffix.",20)
330 close #4
340 Open_over #4,"ram1_"&name$&".html"
350 :
360 print #4,'<!doctype html public "-//W3C//DTD HTML 4.01 Transitional//EN">'
370 print #4,"<html>"
380 print #4,"<head>"
390 print #4,'<meta name="author" content="?">'
400 print #4,'<meta name="generator" content="Suqcess, the QL database manager">'
410 REMark *****
420 REMark ***** style elements and attributes, can be adapted to your wishes
430 :
440 print #4,'<STYLE type="text/css">'
450 print #4,"<!--"
460 print #4,"    body,div,table,thead,tbody,tfoot,tr,th,td,p {"
470 print #4,'    font-family      :"Verdana, Arial, Sans-Serif";'
480 print #4,"    font-size        :x-small }"
490 print #4,"    table {"
500 print #4,"    border-collapse:collapse;"
510 print #4,"    background-color:#F4F4F4;"
520 print #4,"    }"
530 print #4,"    td {"
540 print #4,"    border            : 1px solid Maroon;"
550 print #4,"    height           : 14px;"
560 print #4,"    padding          : 2px 2px 2px 2px;"
570 print #4,"    text-align       : left;"
580 print #4,"    }"
590 print #4,"    #big {"
600 print #4,"    font-size        : large;"
610 print #4,"    color            : Teal;"
620 print #4,"    }"
630 print #4,"-->"
640 print #4,"</STYLE>"
650 :
660 REMark ***** end of style definition
670 REMark *****
680 print #4,"</head>"
690 print #4,"<body>"
700 print #4,"<table>"
710 print #4,'<thead><tr id="big">'&name$&'</strong></tr></thead>'
720 print #4,"<tbody>"
730 REMark ***** first fieldnames
740 for i%=1 to fieldcount
750 print #4,"    <td><strong>"&n$(i%-1)&"</strong></td>"
760 end for i%
770 :
780 REMark ***** now the rest
790 for i%=1 to reccount

```

```

800 RPOSAB #3,i%
810 print #4,"          <tr>"
820 for o%=1 to fieldcount
830     print #4,"          <td>"&FETCH(#3,o%)"&"</td>"
840 end for o%
850 print #4,"          </tr>"
860 end for i%
870 print #4,"</tbody>"
880 print #4,"</table>"
890 print #4,"</body>"
900 print #4,"</html>"
910 :
920 CLOSE #3,#4
930 open #3,con_:out1 #3
940 id=item_select("Ready!", "ram1_"&name$&".html has been generated", "okay")
950 close #3
960 STOP

```

Easy Success

How Success2 came about and hints for EasyPTR programmers
Bob Spelten jr

When the new colours came out we (Wolfgang Uhlig and I) immediately wanted to modify Success to make it look better. But first we had to learn and understand how it all worked and how to use them. Much was explained in various QL Today articles and emailing the experts also helped us on the way. As the Success menus were made with EasyPTR this presented the first problem, because the new colours could not be set there directly. Wolfgang Uhlig has solved that partly while making QCoCo as explained in QL Today v8i5, but that still meant a lot of extra programming to recolour all the Info Windows and Loose Items from SBasic.

The first major help came from Marcel Kilgus when he - in one night's "brute force attack" - created a new "colour version" of EasyMenu_exe where the new colours can be set directly. Now there was no more need to change them from within the program. Unfortunately this is not an official release (yet).

The EasyPTR menu sizes cannot easily be changed from SBasic, so Success has several fixed menus for different screen sizes, each with their own set of sprites. As the new sprites are bigger than the old ones this meant the menu code block would become about 2.5 times as large. Therefore I wanted to design the menus without the sprites and change them when each menu is opened, with the EasyPTR commands MITEM and MINOB using the same set of sprites for all.

This presented the next problem because these commands do not like the new GD2 sprites. Fortunately Per Witte came to the rescue. He had solved that problem before by making use of the option to link sprites together (e.g. Msprv & Sprv, QL Today v8i6). The trick is to first check the mode of the sprite and when it's in a new mode, link that sprite to a dummy mode 4 sprite. MITEM will happily process the mode 4 and WMAN2 will then show the

next "new mode" sprite. The SPRS command could not be used here, although it has no problems with displaying the GD2 sprites, when used for Info- or Loose Objects they are overwritten each time the status changes or the windows are refreshed.

So, for Success2 I made all menus with only text objects, space saving 2/3 for each, plus a new size for Aurora (512x480). I appended them with the mode 32 sprites made by Wolfgang, each paired with a mode 4 version and still saved 10K on the code block. EasySprite is only usable in mode 4 but the old APPMAN program did the job well. Although it crashed regularly in the high mode environment, luckily it was mostly after I had saved the new code.

Where possible Success2 makes use of the extended system sprites. At the start of the program all sprite addresses are set, dummy's created if needed. MITEM and MINOB use these addresses to draw the right sprite. When run in mode 4 the mode 32 sprites are skipped and only the mode 4 ones used. In mode 16 or 32 they are linked to the mode 4 sprites "showing off" their true colours.

The menu colours are almost entirely defined as \$2xx system colours, then you can choose palette 0 to 3 and load your favourite theme there (using QCoCo or the Colour Utilities Disk). You can change colours while the program is running. Also in mode 4 you can now use the familiar palettes known from QMenu and Qpac2. There is a small SBasic program to set a system palette to the old Suqcess colours (and make all the other applications look like a Suqcess ;-)

One remaining snag is that the scroll/pan arrows colour can not be set. For that I made a small procedure to set these

colours just before the application window contents are drawn, using:

```
MAWSETUP #ch\subw,...
```

set colours in the Working

Definition:

```
MAWDRAW #ch,subw
```

Only the application window border colour remains stubborn and is set to a mode 4 "gray" stipple, which should be ok for most colour schemes.

The whole EasyPTR package needs a big overhaul to bring it to GD2 standard but with the help from some experts we managed to find a workaround for most problems, except for the "gray" stipple. Something to do for the next update.

Compared to the previous Suqcess version 1.19 there is not much functionality added to version 2. You can now open a database "Read Only", some bugs were fixed, improvements made in View, in Direct Sort and to a few prompts. Every Loose Item now has a key attached to it. Further changes were mostly in the "looks" department. Suqcess2 only runs under SMSQ/E 3 because of the new colour commands. A trial version can be downloaded from Wolfgang's site:

www.uhlich.nl/ql/

Full versions can be obtained from Jochen Merz Software. They come in English, German and Dutch flavours.

Programming QPTR in S*Basic (Part 10)

(It seems we somehow lost track of the part numbering in previous issues, but 10 should be OK now).

Wolfgang Lenerz

We follow on from the last instalment by examining, this time, a more convoluted way, even if it is a "direct pointer read"

II - READING THE POINTER DIRECTLY

With this command the pointer can be read at any time and the return from the command can be either immediately, or at the occurrence of a certain event, as specified by the programmer. Contrary to the RD_PTR command that we saw last time, there is only one command, RPTR (no "RPTRT"). However, RPTR also takes into account job events. This command (Read Pointer) takes the following parameters:

```
RPTR xabs%, yabs%, end%, winnum%, yrel%,
yrel%, return$
```

→ * *end%* is a variable that determines under what conditions this command returns to the programmer. The conditions are determined by setting individual bits in this variable to 1, according to what one wishes. The following table contains the return conditions, if the corresponding bit is set to 1:

Bit set return if:
to 1

- | | |
|---|---|
| 0 | a keyboard key, or a mouse button is pressed |
| 1 | keyb. key or mouse button continues to be pressed |
| 2 | the key or button is released |
| 3 | the pointer moves away from the given coordinates |
| 4 | the pointer is, or moves out of, the window |
| 5 | the pointer is in, or moves into, the window |
| 6 | NEVER set to 1!!!! |
| 7 | "special" mode |

Most return conditions may be mixed together at your heart's content: if you set both bits 4 and 5 to 1, then the command will return immediately because the pointer is always either in or outside of the window!

You may set any individual bit in this variable to 1 by first setting the entire variable to 0 and then adding 2^x to this variable, where x is the position of the bit in the variable. If I add 2^4 (=16) I set bit 4 to 1. So, by adding 48 (=16 + 32 = $2^4 + 2^5$) I set bits 4 and 5. Of course, you must add this only ONCE for each bit.

The "special mode" which is chosen when bit 7 is set, will lock all windows of all other jobs and show a special sprite, which can be:

- the change size sprite, if bit 1 is also set to 1
- the "move window" sprite if bits 1 and 0 are both 1
- the "empty window" sprite if both bits 1 and 0 are set to 0

When bit 7 is set to 1, all other bits (except 0 and 1) should be set to 1.

This parameter is also used to set the job events on which one wishes the program to return. We discussed the job events last time, please refer to the last instalment of this series.

The job events are included in the high byte of the `end%` word. To set any of these events, proceed as above (2^x where X is the event number, from 0 to 7) but then multiply that value by 256. (Note: from S*Bazic, you can only set the first 7 events (0-6) and not event nbr 7, as that would be exceed the value of an integer in S*Basic. You would need to use a negative number for that). So, to set job event n' 2, I'd add $(2^2)*256$ to `end%` variable.

- * **winum%** contains the number for the main window (=1) or the number of the application subwindow to which the pointer read should apply (especially to know whether the pointer is in the (sub-)window or not).
- * **xrel%** and **yrel%**, which are return parameters, contain, on return, the pointer coordinates in the window or in the application sub-window in which the pointer was when the command returned. They are both relative to the origin (upper

left hand corner) of this window or application sub-window.

- * **xabs%** and **yabs%** are used when bit 3 of `end%` is set to 1. They then contain the ABSOLUTE pointer position - when the pointer moves from this position, the command will then return.

These parameters also contain, on return, the absolute position of the pointer (in all circumstances). Again, this is relative to the screen origin (upper left hand corner).

- * **return\$**, another return parameter, contains the character code (`chr$`) of the key pressed, or one of the following values, with the following meaning:

Key	content of return\$	CHR\$	Meaning
none	0		no key pressed
SPACE/left button	1		Hit
ENTER/right button	2		Do
ESC	3		cancel
F1	4		Help
CTRL F4	5		Move window
CTRL F3	6		Change window size
CTRL F1	7		Sleep
CTRL F2	8		Wake

Thus, with this command, you can also read the pointer. Its disadvantage is that it doesn't take into account any loose items etc... It is thus more difficult to use than the `RD_PTR` command and doesn't use all of the facilities offered by the Pointer Environment.

That's it for today. If you've been following this series continuously, you should now have a firm grasp of the concepts used by the Pointer Environment, and also how to use them from S*Basic.

Next time, we'll look at some additional keywords, which will probably conclude this series.

The EPSON EPL 6200 - any good for QL printing?

Roy Wood

There used to be a large number of Epson mono laser printers and many of those would work with the QL. In recent months the number has been cut drastically. The good news is that the prices of the surviving printers have also come down to affordable levels. So much so that the EPSON EPL 6200, which is the subject of this review, is now about the same price as the 24 pin dot matrix printers were at the start of the QL's inception. By printer standards that is still quite high given that a cheap colour inkjet will set you back less than £40 but, as I have mentioned before in these pages, most of these are as much use to a QDOS/SMSQ machine as a chocolate teapot. The Epson EPL 6200 does work with a QDOS/SMSQ machine and is a very good printer as well so it is worth reviewing it here. Many of you will have more than one system and, I have no doubt, also possess a PC of some sort so I will also mention the PC side of installation and use as it goes along. One thing to mention here is that potential buyers should avoid the cheaper version of this printer, the EPL 6200L. This one will not work with the QL at all because it does not have the emulations built in.

The Printer Itself

The printer is styled along the 1950's bread bin aesthetic so

much in favour with many of the modern printers. For a laser printer it has a relatively small footprint of 407mm x 436mm and stands 261 mm high. There is, however a 250 sheet paper tray which projects from the front by 210 mm. The paper tray will hold most of the plain paper that you want to feed printer with but there is also a smaller paper feed guide at the top of the main tray. If you place a sheet of paper in this it will



take precedence over the tray beneath. Useful for printing on coloured sheets and for such things as invoices and headed notepaper which have a pre-printed design. The paper exits at the top of the printer into a flap which can be closed when the printer is not in use. It comes with the toner cartridge pre-installed and needs a minimal amount of assembly to get it up and running. There are two different methods of input available on the rear of the printer. A standard centronics parallel interface is provided along-

side the ubiquitous USB one. The only other socket on the rear is for the AC input. the printer also has a small fan at the side to vent the hot air from the cabinet. This is probably very necessary given its compact design.

Installation

Installation for QL computers is, of course, not an issue but Windows will need a driver. This is an automatic process and EPSON seem to have managed to sort out the teething problems they had with the automatic installation on my C80 printer. (I only mention this for emulator users who will need to get it up and running on the PC first.) All you have to do is to plug the printer in and leave it switched off. You run the CD setup program and switch the printer on when prompted to do so. This will then detect the printer and set it up on the PC. Once this is done you can go to your emulator and point it at the right device to print from. I only have experience of QPC2 but it was relatively trivial to get QPC2 to see this and set it up to be PAR 1. At this point I was full of apprehension because my currently installed printer is an EPSON C80. This was installed in a similar way and, in spite of its boast that it had an ESCP/2 emulation it has, to this day, steadfastly refused to do anything at all when a QL program tried to print from it. Marcel and I spent a fruitless evening trying to find out what the problem was but could not pin it down. At first I was inclined to believe that EPSON had either

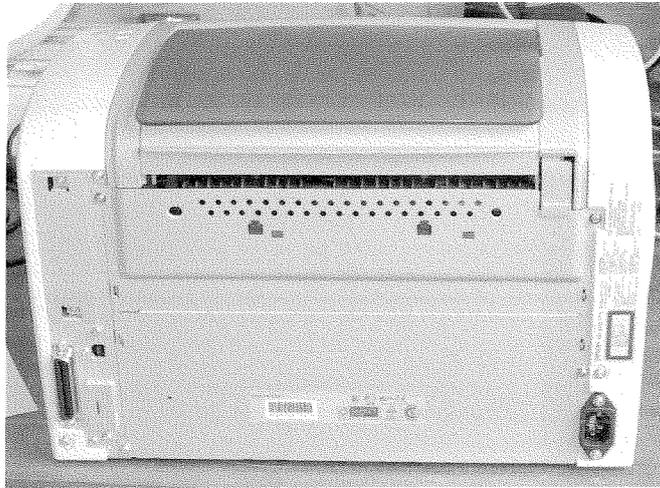
lied about the emulation or changed the way it was accessed. This was confounded when it printed straight from the Q40. Admittedly it only printed in B&W but that was because EPSON had cut down the ESC/P2 emulation on it to be monochrome only.

Let's print!

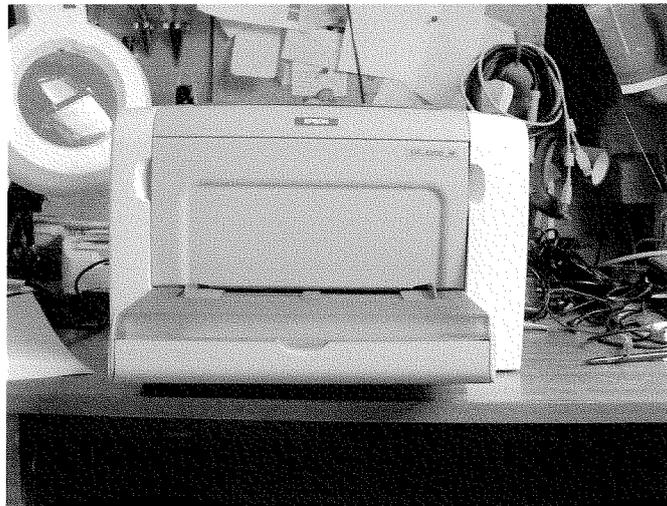
First test was from the PC. I just printed a short page of text. It did this very rapidly and the printed output was crisp and precise. It has a nominal resolution of 1200 x 1200 dpi so this should be good. the QL will, of course, not use this resolution but Rich Mellor's extended ESC/P2 drivers for ProWesS may push it a bit more. Next test was to run it up from QPC2. A page from LINEDesign using the standard ESC/P2 drivers worked well and the output was very good. Text 87 also produced good output using the ESC/P2 drivers from Jochen Merz. The Xchange suite was more of a problem. In spite of the driver being set to the LQ800 mode which had always worked so well with other printers nothing seemed to happen. I tried it again and it printed the page with all the translations perfectly rendered (i.e. £ signs etc.). Both QD and QSpread produced good output when used with the EPSON printer filter as supplied with the programs. It was also able to print directly from MicroEmacs and this indicates that it handles straight ASCII dumps.

Advanced Options

Jochen emailed me a couple of test files to see if it could handle other modes and whether the emulations were switchable as were other EPSON laser



drivers. It seemed to fall down here because it did not recognise the control codes sent to



it to tell it to switch and the output was not useable in this way. *[After several inquiries, EPSON told me it cannot handle EJM - the EPSON Job Language to switch emulation, but it was worth a try as we thought there must be a way of switching it ... and most of the hotliners have no concept about these details anyway - Jochen]* Interestingly one of these was an LQ dump which it had seemed to handle in the

Xchange suite so there is obviously something different in the implementation of this printer's control codes. Physical switching on modern printers has long been abandoned. On the CD which comes with the printer there is a DOS-program which will allow you to change emulation modes but this is limited to running with the printer on the parallel port (since DOS does not know anything about USB) and I am not sure how it would benefit us unless someone could run the program and capture the output so a similar QDOS utility could be run.

Conclusions

On the whole this is a good, functional, low cost laser printer. there are cheaper ones out on the market but their ability to handle QL output is questionable and the ESC/P2 drivers available for ProWesS and Text 87 will give you a good quality print. Lasers are generally cheaper than inkjet in overall running costs and have a longer working life.

I was convinced enough to decide to buy one for my own use. If you are looking for a printer which gives good monochrome output from QDOS/SMSQ then this should be a serious consideration. The EPSON 24 pin Dot Matrix printers (LQ-300+) are only around £130 but the difference in both the output and the speed is considerable. The only other printer in the EPSON range that boasts an ESC/P2 interface is the Stylus

Colour 1520, which will handle paper sizes up to A2 and has a higher resolution at 1440 dpi. This is so far untested by me and, since it has a hefty £320 price tag on it will remain so unless someone can persuade EPSON to give one to me for review.

Given the scarcity of good, modern, printers that are QL compatible this would seem to be a good buy for the user who prints a lot of text or B&W graphics.

Editor's Note:

A good opportunity to list the known, working models of EPSON printers. The following printers are known to work perfectly well (full ESC/P2 colour), but are not made anymore. You will find plenty on Ebay:

EPSON Stylus Color (very old)
 EPSON Stylus Color II (old)
 EPSON Stylus Color 800, 850
 EPSON Stylus Color 740
 EPSON Stylus Color 900, 980
 EPSON Stylus Color 880 (the last known model to work)

SPECIFICATONS

Interface	: 1 x USB 1 x Parallel (Centronics)
AC Input	: 220/240v AC with standard IEC connector
Page Size	: A4, B5, A5, LT, HLT, GLT, EXE, LGL, GLG, F4, Monarch, DL, C5, C6, C10, IB5 (Custom size from 76 x 148mm to 216 x 356 mm)
Installed Memory	: 8Mb, upgradable to 136Mb
Installed Emulations	: LJ4, GL2, ESCP2, FX, I239X, Adobe Postscript 3
Hieght	: 261mm
Width	: 407mm
Depth	: 436mm
Wieght	: 7kg (inc. consumables)
Resolution	: 1200dpi
Performance	: 20Page per minute (max using PC software)
First Page Out	: 10.23secs
Sheet Feed Capacity	: 250 sheets
Max Monthly Capacity	: 30,000 pages
Consumables Life:	
Toner Cartridges	: Standard Yield Toner Cartridge 3,000 pages C13S050167 : High Yield Toner Cartridge 6,000 pages C13S050166
Photoconductor	: 20,000 pages C13S051099

The above figures are approximate and are based on the number of sheets printed under conditions of continuous printing. Intermittent use may reduce page yield.

Approx Cost in the UK : £ 240 (approx. EUR 360)

The MOVEM Instruction and its Register List

Hugh Rooms

I sent Norman Dunbar an algorithm that would disassemble the MOVEM register list without the need for a table of output strings. I was surprised to see that he had a problem with it, QLToday Vol 9 Issue 1 page 20, since I was sure that it worked for all cases. However it turned out that the 'one small problem' he reported was due to a misunderstanding of the way it works.

So here is an explanation of the MOVEM register list, my algorithm, a demonstration that it works (including the case quoted as a 'problem') and a final version in assembly code.

Background – MOVEM Register List

The MOVEM instruction allows a collection of registers to be stored in memory or retrieved in one go, most commonly at the entry or exit of a subroutine.

In assembly code the MOVEM list is given as one of the fields after the instruction field. In Motorola's documentation a comma is used only to separate the two fields, source and destination, and the register list components can take these forms (the quotes are not part of the list): 'Am/An' or 'Am-An' or 'Dm/Dn' or 'Dm-Dn' or any combination that can be translated by the assembler. In the register list the '-' indicates a range of consecutive registers, and a '/' separates individual registers or ranges, eg: MOVEM.L D0-D4/A6/A2-A4/D6, -(A7)

would store registers D0,D1,D2,D3,D4,D6,A2,A3,A4,A6 on the stack and

```
MOVEM.L (A7)+,A2-A4/D6/D0-D3/D4/A6
or
MOVEM.L (A7)+,D0/D1/D2/D3/D4/D6/A2/A3/A4/A6
```

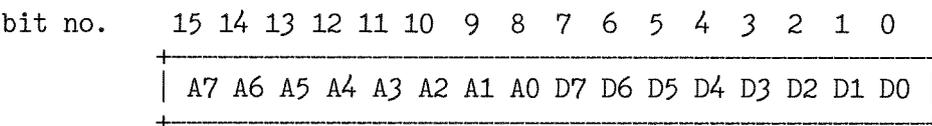
are two of the possible forms to restore them. As this shows there is a huge set of equivalent instructions that would do the same thing in most cases. For a disassembler we need to choose a standard form, more about that later on.

First a word about the binary register list. Norman pointed out (Vol 8 Issue 5 p22) that one case needs the bits in reverse order to the others, but this is a consequence of the way the processor deals with the list.

For subroutines, saving and restoring registers uses Pre-decrement $-(A7)$ and Post-increment $(A7)+$ addressing modes with the normal stack pointer, as in the examples above. The registers can also be saved on a user-defined stack with a different address register, and they can be stored in memory other than on a stack, using a number of addressing modes. In all of these modes, including stacks, the registers are stored in memory so that the result ALWAYS has D0 at the lowest address through to D7, then A0 to A7 at the highest address. If, as is common, only a selection of registers is given, then the sequence is preserved, but only the specified registers are actually copied to memory – there is no 'empty space' left for those that are not copied.

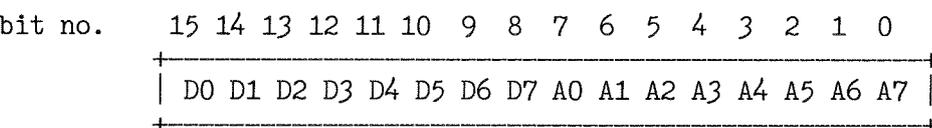
In machine code the 16 bit register list has one bit assigned to each register, and the list is always traversed FROM RIGHT TO LEFT – bit 0 to bit 15. If a bit is set then the register is copied, if the bit is zero then it isn't copied.

So for all the modes which store at, or retrieve from, progressively higher addresses the list has to take the form:



– with D0 stored first at the lowest address.

The Pre-decrement mode is unique in that the start address given is 'one above' the storage area, as always for a stack, and for each register in succession the address is reduced by the data length before the register content is stored, so that the stack pointer at the end has the address of the lowest byte used. To give the memory sequence described earlier, the list word has to be:



So that A7, if specified, is stored first, at the highest address. This is the odd one out, with the bits 'back to front'.

Standard Form

My algorithm disassembles the list to a standard form, in the sequence D0 to D7 followed by A0 to A7, and using '/' and '-' to give as compact a list as possible, so that ranges are used if three or more adjacent bits are set for either Data or Address registers. So the first example given above, for D0/D1/D2/D3/D4/D6/A2/A3/A4/A6, would be disassembled as:

```
MOVEM.L D0-D4/D6/A2-A4/A6,-(A7)
```

In the case of just two adjacent registers, e.g. D2 and D3, then to me D2/D3 looks more apt than D2-D3.

Disassembling the Register List

Two things need attention before starting to analyse the register list.

First, it seems on the face of it that a blank register list is permissible, although it would not be of much use, and I don't see how it can be produced by a MOVEM line in assembler code. There's no help on this in any of the documentation I have, and I don't have access to 68000 hardware to try it out, but if it could occur in a legitimate machine code program then I would test for this as a special case and disassemble it with an explanation, thus

```
<address>      $4890  DC.W   $4890   ; This is the code for MOVE.W to (A0) ..  
<address>      $0000  DC.W   $0000   ; .. with an empty register list.
```

Second, it would be most convenient to reverse the list for $-(An)$ so that the same algorithm could be used for all cases, and I'll assume this is done.

The register number for Data registers is the plain bit number position in the List, but in order to use the same algorithm for Address registers we note that the Address register number is the bit number with 8 subtracted. As bit 3 is already clear for Data registers then clearing it anyway is o.k. for both types. Add \$30 to convert to ASCII.

Algorithm

A problem like this one can often be solved by an approach based on a Finite State Automaton – a rather grand term for a fairly simple idea. My adaptation of it here is a loop that is passed through once for each bit, in a 'State' that depends on the previous history, to decide if each set bit found is a single register or part of a range.

Note that in the output list I put a '/' separator on the end ready for the next item, so there's an extra '/' at the end to be removed as the last action.

We start in State0 and stay in it, with no output of course, until a set bit is met, then that register goes on the output, say D2/, and we change to State1 for the next bit. If this bit is not set then we revert to State0, if it is set then we put its register on the output, D2/D3/ and go to State2. We now have two adjacent registers, but need a third to start a range. If the next bit is not set, back to State0 again.

If it is set then we have a range of three successive registers. The action now is to alter the register number at the end of the list and the previous separator: D2/D3/ becomes D2-D4/. We could now stay in State2, when a set bit next time would change the list to D2-D5/. An unset bit would go back to State0 so that a later set bit would start again as a single register e.g. D2-D4/D6/.

However in State2 the separator keeps on being 'changed' to '-' even though it already is '-', so I include a State3 that simply changes the register number.

A range cannot start with a Data register and end with an Address register, so we must start again in State0 after 8 bits.

For completeness here is the State Transition Table for this algorithm:

State IN	State OUT for bit value →	
	0	1
0	0	1
1	0	2
2	0	3
3	0	3

Traces

Example List is %0100 1111 1110 1100 (spaces just to make it more readable) which corresponds (right to left) to the String 'D2/D3/D5-D7/A0-A3/A6'

TF Services

Compswitch

A UK 4-way trailing socket designed to switch off computer peripherals automatically when the computer is switched off, or (in the case of an ATX computer) when it auto-powers down. *Compswitch* has one control socket, and three switched sockets. Can be used with lights/hifi/monitors—ie a QL monitor can be used as a switch control.

Cost **£24**

superHermes

A major hardware upgrade for the QL. All Hermes features (working ser1/2 at 19200, independent baud rates/de-bounced keyboard/keyclick) IBM AT kbd I/F // HIGH SPEED RS232 at 57600// serial mouse port and 2 other RS232 inputs// 3 I/O lines // EEPROM

Cost (including manual/software)**£90** (£92/£93)
IBM AT UK layout Keyboard.....**£11** (£13/£15)
Serial mouse.....**£8** (£8.50/£9)
Capslock/scrolllock LED**£1** (£1.50/£1.50)
Keyboard or mouse lead**£3** (£3.50/£3.50)
High speed serial (ser3) lead.....**£4** (£4.50/£4.50)

Hermes available for **£25** (£26/£27) Working ser1/2 and independent input, debounced keyboard.

SuperHermes LITE: All Hermes features (see above) + an IBM AT keyboard interface only.

Cost (incl keyboard lead)**£53** (£54/£55)

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

Minerva

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/ "warm" 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

MK1...**£40** (£41/£43) MK1L...**£65** (£66/£67)

MINERVA RTC (MK1L) + battery for 256 bytes ram.
CRASHPROOF clock & I²C bus for interfacing. Can autoboot from battery backed ram. Quick start-up.

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** (£40/£41)
4mbytes RomDisq.....**£65**(£66/£67)
8 mbytes RomDisq.....**£98** (£99/£100)
Aurora adaptor.....**£3** (£3.50/£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** (£35/£36)

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/£44)

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

Relays (8 3a 12v 2-way mains relays (needs 2a power driver).....**£25** (£28/£29)

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

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/£32)

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

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

Data sheets.....**£2** (£2.50/£3)

Control software & manual (for all I/F).....**£2** (£2.50/£3)

QL SPARES

Keyboard membrane no longer on sale
1377 PAL **£3** (£3.50/£4)
Circuit diagrams **£3** (£3.50/£4)
68008 cpu or 8049 IPC..... **£8** (£8.50/£9)
8301/8302 or JM ROM or serial lead **£10** (£10.50/£11)
Power supply (sea mail overseas)..... **£12** (£19/£23)
Other components (sockets etc) also available

Prices include postage and packing (Airmail where applicable) Prices are: UK (Europe /Rest of world). Payment by cheque drawn on bank with UK address/postal order or CASH! I can no longer accept card payments as UK only does PDQ transaction. SAE or IRC for full list and details 22 Feb 03

29 Longfield Road, TRING, Herts, HP23 4DG

Tel: +44 (0) 1442-828254

Fax/BBS: +44 (0) 1442-828255

tony@firshman.co.uk

http://www.firshman.co.uk

Bit	Value	Reg	State in	String	State Out
0	0		0	' '	0
1	0		0	' '	0
2	1	D2	0	'D2/'	1
3	1	D3	1	'D2/D3/'	2
4	0		2	'D2/D3/'	0
5	1	D5	0	'D2/D3/D5/'	1
6	1	D6	1	'D2/D3/D5/D6/'	2
7	1	D7	2	'D2/D3/D5-D7/'	3 → 0
Note the State change to zero at the end of the Data registers					
8	1	A0	0	'D2/D3/D5-D7/A0/'	1
9	1	A1	1	'D2/D3/D5-D7/A0/A1/'	2
10	1	A2	2	'D2/D3/D5-D7/A0-A2/'	3
11	1	A3	3	'D2/D3/D5-D7/A0-A3/'	3
12	0		3	'D2/D3/D5-D7/A0-A3/'	0
13	0		0	'D2/D3/D5-D7/A0-A3/'	0
14	1	A6	0	'D2/D3/D5-D7/A0-A3/A6/'	1
15	0		1	'D2/D3/D5-D7/A0-A3/A6/'	0

Now, at the end of the List (updated Bit number is 16), drop the last separator and exit the Loop to give the final String: 'D2/D3/D5-D7/A0-A3/A6'

Finally a trace to show that it gives the correct result for the 'problem' quoted by Norman: For D2/D3/D4/D5 the binary List is: %0000 0000 0011 1100

Bit	Value	Reg	State in	String	State Out
0	0		0	' '	0
1	0		0	' '	0
2	1	D2	0	'D2/'	1
3	1	D3	1	'D2/D3/'	2
4	1	D4	2	'D2-D4/'	3
5	1	D5	3	'D2-D5/'	3
6	0		3	'D2-D5/'	0
7	0		0	'D2-D5/'	0 → 0

No need to go through the address registers, the result is 'D2-D5'.

The Program

I found no need to alter the algorithm in any way to get this assembly code version to work, so here it is.

With QPC I use the GST Qmac assembler, so there is a header in the code here to suit. The `_BIN` from assembling this can be set up, CALLED, and traced with Qmon as described in George Gwilt's excellent and useful article 'Debugging with Qmon' in QLToday Vol 8 Issue 4 page 11, and of course I have done that.

For testing I have put a register list in D2, and formed the result as a QDOS string: length in a word, followed by the ASCII bytes of the string. When in Qmon D2 can be changed to test other register lists, and of course I have done this too.

The heart of the program is the loop which is traversed once for each bit in the list. The States of the algorithm are realised by putting in A3 the address of the outgoing State from each pass through the loop. The State for the next pass is then entered by jumping to the address in A3.

Since whenever an unset bit is found we go to State0, we can put a check for this at the start of the loop, going to the code for the other States only if a set bit is found.

```

SECTION CODE          ; Required by Qmac ..
DATA      128        ; .. to get executable code.
bra.s     START      ; Skip data area ..
; .. set up here for ease in tracing, for me anyway!
STRLEN    ds.w       1          ; String length ..
STRING    ds.b       48        ; .. and contents
;
START
; Test list:  $4FEC = %0100 1111 1110 1100 = D2/D3/D5-D7/A0-A3/A6
;                                     (bits read right to left)
;
MOVEQ     #0,d2       ; Clear top word for tidiness
move.w    #$4FEC,d2   ; List
lea       STRLEN,a1    ; Address of Length..
lea       STRING,a2    ; .. and of String
; End of test set up
;
; Start up conditions required
; A1      Address of output string length word
; A2      Next free in output string
; D2.W    List word
;
; Local usage
; A3      State address
; D0.W    Bit number in input list
; D1.B    Register number in ASCII
; D7.B    Register letter
;
; Initialisation
clr.w     d0           ; Bit number
move.b    #'D',d7     ; Reg letter for bits 0-7
clr.w     (a1)         ; String length
lea       STATE0,a3   ; Start in State zero
;
SLOOP
; Check current bit
btst     d0,d2         ; Set Zf for current bit
bne.s    BITSET
;
; Here bit is zero, stay in or go into State0, no other action.
lea      STATE0,a3
bra.s    LOOPENDING
;
; Here bit is set, action depends on current state
BITSET
; First turn register number into ASCII character ..
; .. for bits 0-7 this is the ..
move.b   d0,d1        ; .. bit number ..
; .. for bits 8 to 15, clear bit 3 (subtract 8) ..
; .. does not affect D regs so do it anyway ..
andi.b   #$07,d1      ; .. extract reg number ..
ori.b    #'0',d1      ; .. and convert to ASCII.
;
; go to the code for the current State
jmp      (a3)
;
STATE0   lea      STATE1,a3
bra.s    ADDREG       ; Add this reg to output
;
STATE1   lea      STATE2,a3 ; Fall through, to ..
;                                     .. add second reg to output
;
ADDREG
; Add this reg to the output at the end of the string ..
; .. 3 characters e.g. 'D0/'
; (post increment mode, i.e. (a2)+, automatically updates the pointer)

```

```

        move.b    d7,(a2)+      ; Register letter ..
        move.b    d1,(a2)+      ; .. and number ..
        move.b    #'/',(a2)+    ; .. and separator.
        addq.w    #3,(a1)       ; Update string length
        bra.s     LOOPENDING

;
STATE2  lea      STATE3,a3      ; Alter this and last two regs ..
; .. to make a range at the end of the string.
; String is e.g. :
;
;               -4  -3  -2  -1  (A2)
;               -----
;               |   |   |   |   |
;               ... / | D | 2 | / | D | 3 | / |
;               |   |   |   |   |
;
        move.b    #'-',-4(a2)    ; .. change separator
;
; becomes:
;               -----
;               |   |   |   |   |
;               ... / | D | 2 | - | D | 3 | / |
;               |   |   |   |   |
;
; Fall through to change register number ..
;
STATE3
; No need to change State here -- stay in State3
; Simply alter register number at the end of the string ..
; .. as the previous separator is already a '-'
;
        move.b    d1,-2(a2)      ; change register number.
;
;               -4  -3  -2  -1  (A2)
; to get, e.g.  -----
;               |   |   |   |   |
;               ... / | D | 2 | - | D | 4 | / |
;               |   |   |   |   |
;
; String length and Pointer A2 unaltered while still in a range.
;
; Fall through to ..
;
LOOPENDING
        addq.w    #1,d0          ; Next bit number
        cmpi.w    #8,d0         ; End of D regs?
        bne.s     NOTBIT8

;
; Here we have reached the end of the Data registers ..
        move.b    #'A',d7        ; Reg letter for bits 8-15
        lea      STATE0,a3      ; Start A regs in State0
        bra.s     SLOOP

;
NOTBIT8
; Ok, so it's not bit 8 -- check for all done ..
        cmpi.w    #16,d0        ; .. end of reg list? ..
        bne.s     SLOOP         ; .. if not, carry on

;
; Here we have reached the end of the loop ..
; .. final task, remove the last 'separator'
        subq.w    #1,(a1)        ; .. reduce string length by one ..
        subq.l    #1,a2          ; .. and move the pointer back.
;
        moveq     #0,d0          ; Exit with no error
        rts
;
        END                    ; Avoid Qmac warning!

```

George Gwilt has published a nice program to do this, also in QLToday Vol 9 Issue 1, but I think mine is sufficiently different from his to ask for it to be published as well.

RWAP Services NEWS!

New Websites!

We are proud to present our new websites! Starting September, we have separated our business in three separate websites.

You can find them at:

<<http://www.rwapadventures.com/>> (Adventure Games)
<<http://www.rwapservices.co.uk/>> (General Site) and
<<http://www.rwapsoftware.co.uk/>> (Software)

Our old address: <<http://hometown.aol.co.uk/RWAPSoftware/>> is of course still functioning but will be deprecated in the future.

New Products!

QWORD 1.0 NOW WITH DIGITAL
SOUND ON QPC2!

The wait is now over! Q-Word version 1 is finally available!

Platforms:

QPC/QXL, Q40/Q60, Aurora (with SGC)

Prices:

All versions without P-Word £20.00p
All versions with P-Word £30.00p

Notes:

Q-Word DOES NOT require SMSQ/E with GD2 support -OR- SMSQ/E at all on the Aurora or Qx0 machines. It works on the highest colour depth everywhere regardless of Operating System.

The Aurora version is available on either HD or ED disk. For the latter add £1.00 to the price. ED version is uncompressed and can be run directly from the floppy. All other Floppy versions are compressed. QPC/QXL version comes on CD. Non CD versions DO NOW support digital sound on QPC2

Quantum Leap Drives

After many years of unavailability, here they are again! These are high quality Mitsubishi constructed/IBM badged drives with full warranty.

Unlike previously sold ED drives, these do not require FLP_JIGGLE and have no problems formatting 720K disks. However for the latter a switch is included with the cable.

ED Bare unit (no cable)	£ 29.99p
ED Boxed unit (complete with cable/PSU)	£ 98.99p
Single unit Cable (with switch)	£ 4.99p
Dual unit Cable (with two switches)	£ 5.99p

(More options available, contact us for details. Also available mass quantities of ED and DD disks!)

for Windows

For QLers that run Windows or with incompatible hardware for Talent Games, we now have re-released these adventures so that they can run on your Windows-equipped PC. No Emulator, floppies, microdrive backups etc. required, just a one-click install! Ofcourse the full QL line is still available! (See side column)

Talent Games for Windows ea. £ 10.00p
(Each Game includes a runtime installation of QLAY-2 by Jimmy Montesinos)

RWAP Services

35 Chantry Croft, Kinsley, Pontefract, West Yorkshire, WF9 5JH, United Kingdom Tel: +44 1977 610509 (From the UK dial: 01977 610509)
Website: <http://www.rwapservices.co.uk>

We accept:



(For PayPal please add 4% to the total price. Cheques in £ sterling made payable to R. Mellor)

Old Favourites!

Utilities

Sidewriter v1.08	£ 10.00p
Landscape Printing (EPSON printers)	
ImageD v1.03	£ 10.00p
3D object generator	
Q-Help v1.06	£ 10.00p
Superbasic On-Screen help system	
Q-Index v1.05	£ 5.00p
Keyword-to-topic finder	
ProForma ESC/P2 Drivers v1.04 for ProWeSs	£ 8.00p
Printer Driver	

Applications

Flashback SE v2.03 (upgrade only)	£ 5.00p
Database	
QL Cash Trader v3.7	£ 5.00p
Accounting/Finance	
QL Payroll v3.5	£ 5.00p
Accounting/Finance	
QL Genealogist v3.26	£ 20.00p
Genealogy	
Genealogy for Windows	£ 50.00p
QL Genealogist to Windows version upgrade	£ 25.00p
QL Cosmos v2.04	£ 5.00p
Planetarium	
Q-Route v2.00	£ 25.00p
Route Finding	
Upgrade from v1.xx	£ 5.00p
Britain map v1.11	£ 2.00p
Big Britain map (needs 2Mb) v2.03	£ 5.00p
Various Britain Area maps (ask for details)	ea. £ 2.00p
Ireland map v1.00	£ 5.00p
Belgium map v1.01	£ 2.00p
Catalonia map v1.03	£ 2.00p
P-Word UK English Dictionary (500.000 words!)	£ 15.00p
Dictionary	

Leisure

Return to Eden v3.08	£ 10.00p
Adventure	
Nemesis MkII v2.03	£ 8.00p
Adventure	
The Prawn v2.01	£ 8.00p
Adventure	
Horrorday v3.1	£ 8.00p
Adventure	
West v2.00	£ 5.00p
Adventure	
The Lost Kingdom of Zkul v2.01	£ 5.00p
Adventure	
All 6 games above	£ 25.00p
D-Day MkII v3.04	£ 10.00p
Strategy/War Simulation	
Grey Wolf v1.08	£ 8.00p
Graphical Submarine Simulation	
War in the East MkII v1.24 (upgrade only)	£ 5.00p
Strategy/War Simulation	
Open Golf v5.20	£ 8.00p
Sports Simulation	
QuizMaster II v2.07	£ 5.00p
Quiz	
Stone Raider II v2.00	£ 5.00p
Arcade Game	
Hoverzone v1.2	£ 5.00p
Arcade Game	
Deathstrike v1.5	£ 5.00p
Arcade Game	
Flightdeck v1.0	£ 10.00p
Flight Simulation	
All 6 games above (Open Golf, QuizMaster II, Stone Raider II, Hoverzone, Deathstrike and Flightdeck)	£ 28.00p

Notes on Software requirements

The following programs have a minimum SGC card requirement: P-Word, Qword, Big Britain MAP for Q-Route

Julian & Gregorian Calendar

Stephen Poole

```
100 ::
110 REMark PERPETUAL CALENDAR (Julien & Gregorien). v19mai92, by S.Poole
120 REMark Sosigene-Moret tabulation method. (01 01 0001 to 31 12 2499).
130 CLEAR: INIT_: STOP
140 :
150 DEFine PROCedure INIT_
160 LOCAL _1$, _2$, _3$, _4$, _5$, _6$, _7$: init
170 END DEFine
180 :
190 DEFine PROCedure init
200 LOCAL Daychr$, e$, wd$, ym$, unit$, ten$
210 OPEN#1, con_16: WINDOW 512,256,0,0: CSIZE 1,0: PAPER 0: OVER 0
220 _1$='1234567': _2$='2345671': _3$='3456712': Daychr$='mTwtFsS'
230 _4$='4567123': _5$='5671234': _6$='6712345': _7$='7123456':
    e$='???????'
240 wd$='MonTueWedThuFriSatSun':
    ym$='JanFebMarAprMayJunJulAugSepOctNovDec'
250 unit$='000000000111111111122222222233':
    ten$='1234567890123456789012345678901'
260 main_
270 END DEFine
280 :
290 DEFine PROCedure main_
300 LOCAL r, q$, z$, crtmt, daymth, wkdynbr, lpyr, qq$: main
310 END DEFine
320 :
330 DEFine PROCedure main
340 LOCAL q_$, Jan1, flsh, roff, coff, ps
350 REPEAT r
360   CLS: AT 16,8: INK 7: INPUT!!'INPUT date as 8 chars, e.g:
    25041948!'q$,
370   IF LEN(q$)<>8: NEXT r
380   z$=weekday$(q$): AT 16,8: CLS 3: AT 16,48: PRINT z$: INK 2
390   IF z$=e$: BEEP 12345,67: ky$=INKEY$(#1,50): NEXT r
400   crtmt=q$(3 TO 4): daymth=q$(1 TO 2): wkdynbr=dayno: lpyr=ly
410   qq$=q$: qq$(1 TO 4)='0101': q_=$weekday$(qq$)
420   Jan1=dayno: flsh=10: roff=2: coff=4: ps=25
430   CALENDraw Jan1, lpyr, roff, coff, ym$, Daychr$, unit$, ten$
440   Curson daymth, coff, flsh, wkdynbr, ym$, unit$, ten$, Daychr$
450   AT 16,0: PRINT' Hit ESCape to quit or any other key. '
460   IF CODE(INKEY$(#1,-1))=27: EXIT r
470 END REPEAT r
480 END DEFine
490 :
500 DEFine FuNction weekday$(n)
510 LOCAL i$, i: RETURN WEEKDAY_$(n)
520 END DEFine
530 :
540 DEFine FuNction WEEKDAY_$(n)
550 LOCAL l, f, d, m, c, y, lp: l=LEN(n): IF l<8: RETURN e$
560 FOR f=1 TO l: IF n(f)<'0' OR n(f)>'9': RETURN e$
570 d=n(1 TO 2): m=n(3 TO 4): c=n(5 TO 6): y=n(7 TO 8)
580 :
590 IF d<1 OR d>31 OR m<1 OR m>12 OR y<0 OR y>99: RETURN e$
600 IF d=31: SELECT ON m=4,6,9,11: RETURN e$
610 lp=((y+c=0) OR (y MOD 4) OR (y=0 AND c=14 AND c MOD 4))=0: ly=lp
620 IF m=2: IF lp: IF d>29: RETURN e$: END IF: ELSE IF d>28:
    RETURN e$
630 IF c=15 AND y=82 AND m=10: IF d>4 AND d<15: RETURN e$
640 IF c=15 AND y>82 AND m>10 AND d>14: c=-c
650 :
660 SELECT ON c
670   =5,12,16,20,24: i$=_1$:   =4,11,19,23,-15: i$=_2$
680   =3,10:   i$=_3$:   =2,9,18,22:   i$=_4$
```

QL Today has already printed several programs for calculating the Gregorian Calendar, which succeeded the Julian calendar.

The Romans used their calendar mainly to calculate when taxes and interest on loans should be repaid.

The Julian Calendar was invented for Jules Caesar by Sosigenes in 46.A.D. to get rid of all the accumulated errors in the previous one, as the orbital period of the earth does not correspond exactly to any actual calendar. But again, by the middle ages, the calendar was out of synchronisation and Pope Gregory XIII adopted the system invented by Moret, which became known as the Gregorian system.

But to recalibrate the system, ten days had to be 'written off', so in France days from 5th.Oct.1582 to 14th.Oct.1582 did not actually occur. In other countries, the Gregorian Calendar was adopted at different dates, so you will have to adapt lines 630 and 640 to suit your country. To make things clear, c=century, y=year, m=month and d=day. (Look up the word 'Calendar' in an Encyclopædia for your own country to find out which dates were eliminated).

The Sosigenes-Moret method basically uses a series of look-up tables to calculate the day of the week, so there is no maths involved. The program references each table, according to the previous result at the junction of each row and column, until the final result occurs. So it is very rapid and accurate from 1.jan.0001 to 31.dec.2499.

You must enter all dates as 8 numeric characters, such as 12032004. The result is printed as a year-calendar, so you do not have to keep entering dates if they occur in the same year. Do not enter any dates in year 0000, as Caesar forgot to include this one, which is why the second Millenium began in 2001, and why Roy Wood got involved in much anguish in the columns of this magazine!

Unfortunately I do not have details of when each country adopted the new system, so I shall have to leave that up to readers. (Sorry).

So, if you want to know on what day the Battle of Hastings occurred, or when was the famous battle on St. Swithen's day, then just use this program.

Or if you want to know if such and such a year was a leap-year, just enter any day of that year and look to see if the calendar prints out the 29th of february...

Of course the Gregorian Calendar applies mainly to Christian countries. So if you live in an Islamic, Chinese or other community you will need to use an even more complicated method to get rapid conversions.

```

690 =1,8,15: i$=_5$: =0,7,14,17,21: i$=_6$: =6,13: i$=_7$
700 END SElect : i=7
710 :
720 SElect ON y
730 =0, 6,17,23,28,34,45,51,56,62,73,79,84,90 : i=1
740 =1, 7,12,18,29,35,40,46,57,63,68,74,85,91,96: i=2
750 =2,13,19,24,30,41,47,52,58,69,75,80,86,97 : i=3
760 =3, 8,14,25,31,36,42,53,59,64,70,81,87,92,98: i=4
770 =9,15,20,26,37,43,48,54,65,71,76,82,93,99 : i=5
780 =4,10,21,27,32,38,49,55,60,66,77,83,88,94 : i=6
790 END SElect : i=i$(i): s=m
800 :
810 IF lp:IF m='1': s=4: END IF : IF m=2: s=8
820 SElect ON s
830 =1,10: i$=_1$: =5 : i$=_2$: =8: i$=_3$: =2,3,11: i$=_4$
840 =6 : i$=_5$: =9,12: i$=_6$: =4,7: i$=_7$
850 END SElect : i=i$(i)
860 :
870 SElect ON d
880 =2,9,16,23,30:i$=_1$: =3,10,17,24,31:i$=_2$: =4,11,18,25:i$=_3$
890 =5,12,19,26 :i$=_4$: =6,13,20,27 :i$=_5$: =7,14,21,28:i$=_6$
900 =1,8,15,22,29:i$=_7$
910 END SElect : i=i$(i): dayno=i
920 :
930 i=i*3-2: dy$=wd$(i TO i+2): i=m*3-2
940 RETURN dy$&&' '&d&' '&y$(i TO i+2)&' '&n(5 TO 6)&n(7 TO 8)
950 END DEFine
960 :
970 DEFine PROCEDURE CALEndraw(_1,_2,_3,_4,_5,_6,_7,_8)
980 LOCAL m$,l$,u$,t$: CALEndraw_
990 END DEFine
1000 :
1010 DEFine PROCEDURE CALEndraw_
1020 LOCAL n,l,r,c,f,m,d
1030 n=_1: l=_2: r=_3: c=_4: f=1
1040 m$=_5: l$=_6: u$=_7: t$=_8
1050 FOR m=1+r TO 12+r: AT m,1: PRINT m$(f TO f+2): f=f+3
1060 FOR f=1 TO 31: AT r-1,f+c: PRINT u$(f): AT r,f+c: PRINT t$(f)
1070 INK 4-(n,5)
1080 FOR m=1 TO 12
1090 d=31: IF m=2: d=28+1
1100 SElect m=4,6,9,11: d=30
1110 FOR f=1+c TO d+c
1120 AT m+r,f: PRINT l$(n): n=n+1
1130 IF n=8: n=1: END IF : INK 4-(n,5)
1140 END FOR f
1150 END FOR m
1160 END DEFine
1170 :
1180 DEFine PROCEDURE Curson(_3,_4,_5,_6,_7,_8,_9,_A$)
1190 LOCAL w$,a$,b$,c$: _1=crtmth: _2=roff: CURSON_
1200 END DEFine
1210 :
1220 DEFine PROCEDURE CURSON_
1230 LOCAL w,r,l,c,y,x,j,f
1240 w=_1: r=_2: l=_3: c=_4: y=w+r: x=1+c: j=w*3-2
1250 w$=_7(j TO j+2): a$=_8(1): b$=_9(1): c$=_A$(_6): OVER -1: INK 4
1260 FOR f=1 TO _5
1270 AT r-1,x: PRINT a$: AT r,x: PRINT b$
1280 AT y,1: PRINT w$: INK 2: UNDER 1: AT y,x
1290 PRINT c$: UNDER 0: INK 4: ky$=INKEY$(#1,ps)
1300 END FOR f: OVER 0
1310 END DEFine
1320 ::

```

QPC2 TIP – Starting Windows Files

Dilwyn Jones

The author of QPC has thoughtfully provided a BASIC extension called QPC_EXEC which allows you to start a Windows program or file. This comes in useful for people like me who mostly use a PC for QPC2 and every once in a while need to fire up a PC program, e.g. to use a PC word processor to create a document containing graphics which a text-based word processor like Quill could not do. Windows includes a mechanism like File Info 2 which allows us to execute a file which is not a program.

Such files have programs 'associated' with them. The type of file is determined by the ending of the filename. For example, a Windows file ending with .DOC is normally associated with the Word or Wordpad programs.

I keep a letterhead document called LETTERHD.DOC to save me typing my address and phone number etc into the PC word processor every time. This is saved in the C:\My Documents\

folder on my PC's hard drive. To quickly and conveniently start this from the QL side, all I need to do is define a hotkey to send a QPC_EXEC

command to SBASIC:

```
ERT HOT_CMD('d','QPC_EXEC "C:\My Documents\letterhd.doc"')
```

So when I press ALT d in QPC2, it fires up Microsoft Word (or Wordpad or whatever program used on systems where Word is not installed) which in turn loads the file called LETTERHD.DOC. I don't even have to ALT TAB to Windows, select the Desktop, go to the Start menu, or click on Word's icon to start it! It works both with QPC2 in full screen mode and with QPC2 in windowed mode on my system at any rate.

I was quite amazed when I realised this was possible and just how well it worked and how much time and effort it saved me! Of course, it's not just word processor files it works with, you can use this with many file types which have associated programs.

Launchpad users can set up a similar thing. Goto the Utility menu and select the QPC:Launch DOS/Windows File command. In the window which appears, select options 4, 5 or 6 at the bottom to set up any of the three 'favourite' filenames for this purpose. Enter the Windows path/filename (e.g. C:\My Documents\LETTERHD.DOC in my example above). After that, you need only click on the 4, 5, or 6 icons alongside to copy the filename into the selection box above, then click on OK to start the file!

QPC2 PRANKS

Dilwyn Jones

Here are some ideas for pranks you can play on your QPC2- and QPAC2-using friends, e.g. as a birthday joke.

1. WINDOWS RULES OK?

The first one relies on your friend having a recent version of QPC2, which includes the QPC_WINDOWTITLE extension. Add this line to his/her boot program at a suitable line number:

```
1 QPC_WINDOWTITLE "WINDOWS RULES OK!"
```

Just make sure you are at a safe distance away if your friend is not too keen on Windows!

The text you supply after the QPC_WINDOWTITLE will be printed in the title bar above the QPC window if it is running in Windowed mode, just before the QPC2 copyright message.

There are more serious uses of this facility. It was intended to give a short identifying title to each instance of QPC2 for people like Jochen who are determined to get their money's worth from QPC2 by running several copies of QPC2 and manage to get themselves confused when they can't remember which is which. It could even be used as a simple message facility in conjunction with a short SBASIC program used as a reminder, e.g.

```
REMark max 45 minutes at the keyboard at  
a time  
period = 60*60 : REMark 1 hour's worth  
of seconds  
time = DATE
```

```

REPeat loop
IF DATE >= (time+period) THEN
  BEEP 5000,0 : REMark audible warning
  QPC_WINDOWTITLE "!TIME FOR A BREAK!"
  EXIT loop
END IF
END REPeat loop

```

```

10 REPeat LOOP
20   IF RND(1 TO 1000) = 1 THEN
30     EXEP BUTTON_SLEEP
40   END IF
50 END REPeat LOOP

```

Either compile this program and make it run in the background or (for SBASIC users) run it as an SBASIC job. Alter the RND range in line 20 make it zap the programs more or less frequently as required. Just sneak up to the computer and start this program running without your friend's knowledge!

2. JOBS PRANK

This one is a pretty evil trick to play on any pointer environment and QPAC2 user with enough memory to have several programs running - be prepared to risk losing a friend though!

If your QPAC2-using friend likes to make full use of QL multi-tasking and task switching, perhaps like me he/she has 5 or 6 programs on the go at a time. Typing into a wordprocessor, Launchpad running in the background, database waiting to be updated, a game of The Wall waiting to be played, a Qascade menu and a couple of other programs running... All of a sudden, the program disappears into the button frame without warning...

3. CUP HOLDER

The author of QPC2 has provided audio CD control facilities for you to play music CDs. Some SBASIC extensions are included to control start and stop etc. There's a CD_EJECT and CD_CLOSE pair of commands to open and close the CD-ROM drive tray respectively. This provides a great opportunity for a prank. Simply run a little timer program which randomly ejects and closes the CD-ROM drive tray, or simply time a message which invites the user to feed the PC for example:



Your North American Q60 and QL Source!

Quantum Leap ED

The megabytes are back!



Once again, ED drives are available for SGC/GC equipped QLs!

Quantum Leap ED Drives are high quality units manufactured by Mitsubishi for IBM (these units were used on the RS6000 workstations). They work great with all disk sizes (720K requires a capacity switch, included with the cable)

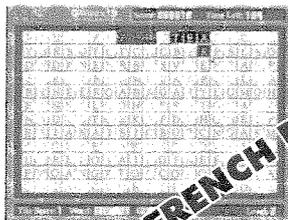
- Very reliable. No FLP_JIGGLE required.
- 100% new (no 'remanufactured' units or pulls)
- Individually tested on real QL hardware.
- 6 months unconditional warranty.
- Available in boxed and bare units

Contact us for details! - Shipping worldwide except the UK. For UK Customers, contact RWAP Services.



Quantum Leap has partnered with SINTECH in Germany to bring you quality QL Membranes. Now you have no excuses for keeping your QL in storage! Made of high quality material with extra long leads these are guaranteed to last you a long-long time.

RWAP Software



NOW WITH FRENCH DICTIONARY






Q60

The only Authorized Reseller in America!

PLUS:

Linux
m68k

Dilwyn Jones
Launchpad

The
QLCDROM
COLLECTION

Phoebus Dokos - Quantum Leap 2306 Byron Ct., Indiana, PA15701-2323, USA
email: ql.css@dokos-gr.net Website: www.dokos-gr.net, Phone/SMS: +1 (724) 467 2404
(SMS acceptable from most European carriers)
We accept: *PayPal*, Cashier's Checks, Money Orders and Personal Checks (US Funds). For PayPal, add 3.5%. Checks payable to: "Phoebus Dokos"

QL Membrane	US\$29.00
Does not include Shipping & Handling.	
Shipping Costs:	
USA	US\$ 3.85
Canada	US\$ 4.50
Europe	US\$ 5.00
Rest of the World	US\$ 9.00

51

```

time = DATE
period = 60*RND(1 TO 60) : REM random number of minutes up to 1 hour
REPEAT loop
  IF DATE >= (time+period) THEN
    CD_EJECT
    BEEP 5000,0
    message = RND(1 TO 3)
    SElect ON message
      =1:PRINT #0,'Please may I have a chocolate biscuit!'
      =2:PRINT #0,'Oops, sorry.'
      =3:PRINT #0,'Time for your tea break, I will hold your cup for you.'
    END SElect
    period = 60*RND(1 TO 60) : REMark random delay before next time
    PAUSE RND(50 TO 1000) : REMark random pause before closing drive
    CD_CLOSE : REMark close the drive tray
  END IF
END REPEAT loop

```

To Eindhoven for only 8 pounds

Al Boehm

RyanAir is a no-frills airline that is very inexpensive. It has flights from London (Stansted) to Eindhoven for 8 GBP! There are also direct flights from Barcelona to Eindhoven but that's all there is direct to Eindhoven for the summer 2004 through 30 October season. However, there are about 45 cities that have RyanAir flights to Stansted.

What's the catch?

Well, Stansted Airport north of London may be a little hard to get to although there is a train from Liverpool Street Station in London. RyanAir only allows one checked bag of 15 kg (6 euro extra for each kilo over with max of 29 kg) and one carry on of 7 kg.

All food and drinks from the stewardess must be bought. A cup of coffee cost me 2.50 Euros this May!

The 8 GBP price was good if you bought your Eindhoven ticket 12 June. The price will go up as the date draws near. RyanAir specifies it is not a connecting airline. Thus if you bought a RyanAir flight from Cork to Stansted then from Stansted to Eindhoven, you can not check your luggage all the way through but have to collect it and stand in line to put it on the next flight. And they

will not guarantee a flight connection. Also the times may not be as handy as you would like. RyanAir uses mostly small airports somewhat away from the major cities which can make them hard to get to. But I will take a small airport with 8 gates over a 61 gate airport zoo anytime.

Other than these limitations, I can recommend it as a pretty good airline. Schedules and tickets can be obtained on:
<http://www.ryanair.ie/>

There are two scheduled flights on Friday 15 Oct 2004 to leave Stansted 0645 and arrive at Eindhoven 0845 and the second to leave Stansted 1830 to arrive at Eindhoven 2030. Coming back on Sunday 17 Oct there is only one flight that leaves Eindhoven at 2055 and arrives Stansted 2055 (one hour flight and one hour time zone change.)

This is what their site says about the Eindhoven Airport: Eindhoven Airport is small, modern and efficient, and has cash machines, a restaurant, car hire and taxis.

Bus: The Hermes bus company operates a twice-hourly service between the airport, the Eindhoven city centre/Eindhoven Central Station. For the schedule, please call 0031-(0)900-9292. Single tickets can be bought from the driver, and the 'strip card' system in use throughout the Netherlands is valid on this service.

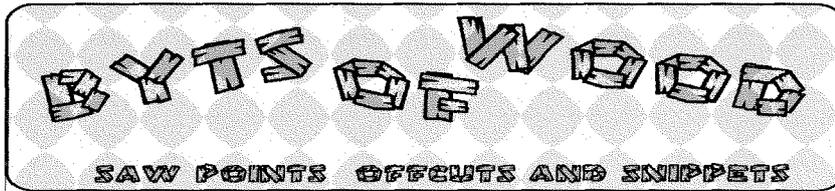
Taxi: Taxis are available from outside the airport.

The Next Issue

Issue 3 of Vol. 9 will be published a bit later than usual - at QL 2004 in Eindhoven!

We feel this special event justifies the delay, and we also hope to have another "goodie" included with this issue.

If you cannot make it to the show (a big miss, it already seems) then please don't be suprised that the issue will arrive later.



One of the more popular programs on British TV of late has been something called 'The Life Laundry'. It is, of course, yet another of those examples of low cost low brow drivel that TV companies delight in and call 'Reality TV'. If this is the reality of those commissioning the programs I can only assume that they lead very sad lives. The premise of the program is that two 'advisors' look at the subject's life and then tell them what they have to get rid of to make themselves happier and more fulfilled individuals. I suspect the answer to that would be to have a TV crew and the two drippy, opinionated presenters put their dreary existence on TV but that is another story. I mention it here because it has a reflection, albeit a circuitous one, on my recent experiences with QDT.

QDT is rapidly approaching release and is now in a pretty usable state. I am, in fact, using it on both of my machines and it works well. In the course of an installation of QDT the user's hard drive is scanned for programs and they are sor-

ted into folders based on a set of rules written into the installer. When I had completed this on my Laptop at the US show I found that the folder marked 'Unknown' was quite large and part of my function as one of the beta testers was to try to identify what these 'unknowns' were and help tighten the rules to make the final installation smoother. All well and good except that I had no idea what some of the things I had on the drive were.

Past Gems and Clinker

As I trawled through all of these programs so I could give Jim Hunkins a list of the oddities it had thrown up, I realised that there were, in the midst of all the clutter, one or two little programs I had forgotten about. Most of my QL computing is via QD, QSpread, LINE-Design and DATAdesign with Text 87 making a reappearance in its new High colour friendly mode. I used to use a great many more programs but I had largely forgotten about them until this little excursion down memory lane.

One of the programs that came to the fore was called 'Smash' this is a wonderful little utility that is of great use to people who run their systems on limited memory or want to run from small media like ED disks or just a ROMDisq. It will take an executable program and squash it down so it occupies less space but will still run as before. Of course you cannot configure the program once it has been 'smashed' because the config block is no longer accessible to the configuration programs. This means that you have to do all of the configuration before running Smash but once it is all happy the 'smashed' code will take up less space on the media and run faster too.

I also re-discovered FComp. This is a very useful utility which will compare two files and show the differences for you. It seems a little more versatile than Compare which also seems to fall over on my current system. Another useful little program is 'Thinfo' which will show the THINGs installed on your system and give you information about them. Good both for troubleshooting and for general information.

Another great re-discovery was Dilwyns PROCMAN, a program I had forgotten long ago. Most of this can be done from

QD these days but it was a neat way to get a set of routines from a BASIC program and it would have come in handy when I had to re-write my invoicing program and did most of the procedure extraction by hand. If only I had remembered it was there.

I also took a look at SCRview. This was a QL '_pic' file viewing program from 1993. Nice interface and very neat pointer. It was written by 'The Beginners Club' of Italy. Maybe you guys could do an update that would use the high colour screens. That would be useful. In fact a high colour screen grabbing utility would be a nice addition to the toolbox as well - while you're at it.

In amongst all of this there was a lot of 'clinker'. My original hard drive was in a Qubide/Aurora/Super Gold Card system and arrived on my first QPC2 system by doing a backup and restore, first with the Norback program and later with The Knight Safe. This, of course, meant that it had all of the Qubide programs on it as well as the SuperHermes and ROMDisq ones. I also realised that I had managed to get things like multiple versions of Menuconfig, Menu Rext and other stuff like this scattered over the drive.

Going through all of this has got me to tidy my drive up a bit at the very least and has led me to re-examine older programs that I will probably find quite useful in the future.

You can if Icon

One other thing that QDT has led me into doing over the last couple of months is spending ages playing around with the icon drawing tool. I have always been a bit of a sucker for computer graphics and one

of the first programs I bought for my QL was QL Paint. It was my involvement with LINE-Design that led me into re-writing the manual for PROGS and I have always found the production of these things to be great fun. QDT comes with a standard set of icons for its components and some of the more recognised programs. The user, however, has control over the icons and it is possible to use any icon file of a set size. It also comes with an icon editing and design tool that works with the new colours and had Wolfgang Uhlig chomping at the bit wanting to buy QDT just to own the utility. I would not say I have produced a Mona Lisa for the computer user but it was fun drawing the stuff.

And Talking in Colours (to paraphrase the Rolling Stones)

Whilst I was at Eindhoven a month or so ago I had a chance to tackle Wolfgang Uhlig about, what I thought, was an omission in QCoCo. I had noticed that, since the arrival of the new colour versions of some of our programs I had been having a small but annoying problem with QD. I use QD a lot for many different tasks and one of these is to produce this column as well as other text for insertion into LINE-Design. Since my fingers can spell less well than my brain I always run the finished item through the spell chucker. In the last few versions of QD this has thrown up a box with mis-typed words displayed in it. In the low colour version this is fine but the hi colour versions always produced a box with yellow text on a blue background or something similar. This was unreadable and I

spent many hours with QCoCo trying to find the setting to change the background.

I found the ink colour for it which was 'Info Window - 2nd Ink' but the background colour eluded me. Since Wolfgang was at Eindhoven I got him to look at it, thinking he had missed out a setting. He was equally puzzled because he had used the parameters given to him by Marcel and they were all accounted for.

The explanation was a little bizarre. It seems that somehow, during the writing of QD, the background for this item was not linked to the usual 'Info Window - paper' but to the separator from QSpread. If you go into QCoCo and change the separator colour then the background for the spellcheck window changes too. This took some investigating and I am still impressed that he thought of it but there you are. Jochen says this is a bug and will be fixed but, in the meantime, if you want to change that colour that is where it is.

Pick Your Own

Another useful, and much under applauded, utility included with QCoCo is the Colour picker. This produces a Rainbow styled display of the available colours which you can roam across with the cursor. As you do this you can see the HEX number in a box beside the main window displayed as either Truecolour or System Colour.

At the bottom of the display you can see the 16 'Favourite Colours'. These are the colours displayed when you first choose to change the colour of an object in QCoCo. You can use the spectrum box to change any of these colours

and thus alter your 16 'Favourite Colours' You can also 'send' the current Colour's HEX number to the stuffer buffer so you can insert it into any program you may be writing or any configuration you may be doing. Another useful and elegant piece of programming from an under-appreciated programmer. This should be in everyone's toolbox - especially since it is free.

EPSON and the Printer thing

I felt a great sense of foreboding when I wrote the printer review that you will find on other pages of this magazine. There I was saying 'This printer works fine with most QL programs and we all know what those words mean don't we?

As I wrote the review you can hear the sound of EPSON's low end laser printer construction line shutting down. Such is the awesome power that a review in QL Today seems to wield. Nevertheless, unheeded of those poor workers whose jobs I am closing, I had to proceed and write the review.

After I had written the review and sent it off to the magazine I found a few other things which some of you may wish to know about so I have added them in here. One of the more interesting things is that you can buy a serial interface for this printer (Epson Part No. C12C823081 around £70 + VAT) and this should enable you to print direct for a QL without a serial to parallel interface or a Super Gold Card. OK, so I know that means that you will have a QL, worth all of £20 connected to a printer that has now cost you around £280 and has approximately 8 times the memory of the computer

but it is an intriguing thought.

Among the other available accessories are

C12C802051 - Duplex Unit to allow printing on both sides of the paper, around £80 + VAT

C12C823914 - Network adaptor, puts the printer on an Ethernet, around £120 + VAT

C12C823972 - 802.11b Wireless network adaptor, allows wireless printing, around £180 + VAT

None of these are cheap options and the last two are not a lot of use to QL users although Q60 users could share the printer with other computers on an ethernet and QPC2 users could do the same on a wireless link via the PC's interface.

Print (re-)#

Rich Mellor brought the printer driver thing up on the user group as I was putting the finishing touches to this column. He mentioned two of the ideas that I had put forward in previous columns as possible solutions to our printer dilemma. I had discussed my ideas at length with Marcel and Jochen while we were on the US trip this year and it would seem that the only really viable solution is Marcel's idea of a ghost-script driver. There really is no other easy option.

The idea of a box which could be programmed with the drivers for your printer does not make a lot of sense when you think that the printer drivers provided with most printers need a version of Windoze to run. Few printers these days come with DOS drivers and even if you managed to get a DOS driver running on a simulated version of DOS in the box someone would have to write a DOS program to accept the input from the QL. In

order to get that to work you would need to have an output from the QL that it would understand. To get this work from a large selection of QL programs you would need to use one of the EPSON drivers as used by Quill etc. These are based on, at best, 24 pin DOT matrix devices so you would wind up with a modern 1200 dpi printer producing 24 pin output. And that is just the text! What you do about graphics is a whole different metal food receptacle of wiggly things.

And Speaking in Animal Terms

To continue with the animalistic analogy Nasta let the furry mouse consumer out of its hessian container when he sent a post to the user group which was only intended for private consumption. Since this is already in the public domain I can now mention it here. As a result of the needs of the people he works for and also at the prompting of some members of the QL Community he is working on a small device which will run SMSQ/E.

To quote him:

'General specs are as follows:

- * 66MHz 68SZ328 CPU (performance should be about 50% faster than SGC)
- * VGA monitor output (640 x 480 resolution or slightly over - theoretical limit is 1024x512 + there is a virtual screen feature) graphics in 256 colours (65536 is possible but only for lower resolutions due to memory bandwidth limitations)
- * Integrated small LCD display (6-7" diagonal) option to make a palm size computer, stylus digitizer

hardware included for touch sensitive displays.

- * 32 or 64M of RAM built in (not expandable)
- * 2M (at least) of Flash built in
- * MMC or SD flash card for storage (Sony Memory Stick is also possible but as this is less proliferant(sic), more expensive, and larger, it will probably not be used)
- * 2 serial ports (or one serial and one IRDA) with high baud rate support.
- * Matrix keyboard (8x8, probably compatible with QL keyrow assignments)
- * USB device port (NOT host controller - the board would be a device to be connected to a host for data transfer)
- * Mono sound in/out
- * Real Time Clock
- * Low power
- * (possible extra features like a QL compatible expansion bus, CF or IDE port, parallel port, dimensions that make it fit a regular QL case, etc).

Regarding this piece of hardware, I am trying to sort of skid around on the edge of a blade - trying to make it both what my employer wants it to be, as well as a product for the QL community. He is willing to pay for the hardware side of the development, and this hardware could then be made available to the QL community essentially at cost, which is what the community needs. On the other hand my employer needs the expertise people of the community have regarding porting of the OS and some applications. The end product could then be marketed in several markets.

Don't ask me about costs yet,

I have no idea as it's all under development. I know that boards are going to be made in neighbouring Slovenia, which is now part of the EU.'

Since all of the above appeared on the mailing list I am sure he will not mind me repeating it here for those who are not currently on that list. This is a very interesting project and one which should be supported by as many people as possible. It raises a lot of points which have been put forward before by people such as Arnould Nazarian.

If we could move an SMSQ/E driven computing device back into a more commercial market we would be in a better position to attract some of the more professional programmers back to do some work. For a small device a lot of our programs are more than adequate and, having a more viable user base, we could work on improving SMSQ/E. Nasta has mentioned that the proposed box would require some work which could be of some use to all of us. Talking about the proposed project he says:

'As we discussed, he wants to build a small machine that would drive a CRT or LCD monitor, could be easily programmed, and would run some games. The idea of running it under SMSQ/E was discussed and he seems very interested in this as it's small and relatively easy to program. I've also roped in another 68k programmer that used to program Atari STs etc, and he is very impressed by the docs I gave him on SMSQ/E - he also downloaded the source. I plan to drag him by his heels if needed, to QL2004 later this year... he's

the kind of guy that is capable of doing anything if only he finds it interesting ;-)

This is, as I said, a project we need to support any any way possible. It is also very much in the planning stage so don't start bombarding Nasta with requests for one. Mind you, if you think you can help I am sure he would like to hear from you.

QL 2004 - again

Since he mentions QL 2004 that is a good enough cue to remind you all that we need to support it as best we can and it would be good to see as many of you there as possible. Given the proposed program of events and the people who have already committed themselves to attending it would seem to be one of the better QL shows in recent years. I believe that a full prospectus of events is published on another page in the magazine. We have been trying to get nominations for the awarding of a Golden Clive award to the person or persons who have done the most for the QDOS/SMSQ community in recent years. These nominations have been a bit slow in coming in so please VOTE NOW! you can send a postcard to either of the QL Today addresses in the UK or Germany or email Jochen, Dilwyn or myself. Don't forget, this is your opportunity to reward someone who has worked hard to keep your favourite system alive.

So, book your ferries, planes, dog sleds or whatever other mode of transport is to your liking and get yourself over there. I won't mention the Chinese.

QL2004

EINDHOVEN

16th October

LIGHT CITY CHALLENGE!

The QL2004 team invite you to celebrate the QL's 20th birthday at an international workshop to be held in Eindhoven, the "Light City" of the Netherlands.

We shall be looking back on 20 years of the QL, but also looking forward. Our Light City Challenge is to take the QL into its 21st year and beyond.

VENUE

Pleincollege St. Joris, Roostenlaan 296, Eindhoven, Netherlands.

Saturday 16th October from 10.00 to 17.00.

The Pleincollege St. Joris is one of the oldest and most popular QL workshop venues with a large hall for the main meeting and smaller rooms for demonstrations and lectures. The hall is well provided with electric sockets, but please bring your extension leads etc. For demonstrations we can make use of projectors, and it is helpful if your system can be connected to these.

There are no cafes or restaurants in the immediate vicinity of the show venue, but the show hosts, the Dutch user group Sin-QL-Air, will provide light refreshments at cost price throughout the day.

ACTIVITIES

Most of the main traders have indicated they will be present at the show, and there will be ample room for you to set up your own system. In addition we are planning several workshops and a QL Forum.

A QL Internet Connection Workshop will explore the latest progress in developing emailing from your QL, and a Colour Workshop will look at tools and techniques for using the new colours in your own programs.

We also hope to have some Q60 related activities and news of expansion cards for native hardware. We would also like to see some progress made in improving QL printer compatibility.

To end the formal proceedings, there will be a QL Forum with a panel of "experts". We are still inviting nominations for this panel, and would like some questions for the panel in advance.

BY CAR

Travel to Eindhoven on the Venlo - Antwerp Motorway (A67 - E34) and leave at Knooppunt Leenderheide following the signs Centrum. You are now in the Leenderweg. At the first roundabout turn left into Floralaan West. Stay on this road to the first traffic lights where you should turn left. You are now in the Roostenlaan. Look for the signs to Pleincollege St. Joris, which is on the right hand side of the road. There is ample parking space on the school campus.

PUBLIC TRANSPORT

There are frequent trains to Eindhoven from all corners of the Netherlands.

For train times and prices go to www.ns.nl, click on the English link and use the snelplanner.

To catch a bus to the show venue, turn right on leaving the station platform and go to the bus station at the rear of the railway station. You need bus line 7 in the direction of Waalre, which is a half hourly service. Ask for the stop Roostenlaan/Floralaan. The journey is over 2 zones, (i.e. 3 strips of your strippenkaart) and takes just over 10 minutes. When you leave the bus, turn left, walk over the Floralaan and keep walking straight on. Pleincollege St. Joris is about 5 to 10 minutes from the bus stop on your righthand side.

If you prefer to use the taxi, then you should turn left when you leave the station platform and go to the front of the station. (Please note there is no longer a treintaxi service in Eindhoven.)

**More information on the
QL2004 website - see below.**

Geoff Wicks, 56 Peveril Crescent, West Hallam, Derbyshire DE7 6ND, U.K.

Tel: +44 (0)115 - 930 3713

email: gwicks@beeb.net

Web: <http://members.lycos.co.uk/geoffwicks/ql2004.htm>



The QL Show Agenda



Don't miss QL 2004 - see reverse side!

QL Meeting - (D) Berchtesgaden

Saturday, 2nd of October, 10:00 to 17:00

Hotel Schwabenwirt, Königsseer Str. 1

Same venue as last year!

Although J-M-S will be moving (see News and Ad), we'll make it to the show, of course! TF Services, Q Branch etc. will be there too, as usual.

Norwich Sinclair Show - (UK) Norwich

Saturday, 6th of November

Alec Bussey Centre off Rowington Road, Norwich NR1 3RR

More information, including map, is on the show website at

<http://www.speccyverse.me.uk/orsam/>

QL Meeting - (UK) Byfleet

Sunday, 28th of November, 10:00 to 16:00

Byfleet Village Hall

Same venue as all the years before!

The Hall is just inside M25, between jns 10 & 11, and just South of A245.

• From M25, jn11; go East towards Weybridge, then turn right (South) onto A318 and follow it to Brooklands (now a business park), then through the old racetrack, turning right onto A245. Turn left at second roundabout, left at little roundabout, and Hall is on right.

• From M25, jn10; take A3 towards London, left onto A245 towards Woking at next junction, (Painshill). After A318 joins, turn left at second roundabout – see below. (Or you can take the earlier left fork into Byfleet, go right at first small roundabout and left at the next, by the green.)

• From A3, either direction; leave at Painshill junction with A245 Cobham/Woking head for Woking.

By train; Byfleet & New Haw is on the Waterloo Woking line. It's 12 to 15 mins walk. Don't go through Brooklands – carry on down the old road, across the A245, and on to the green, then turn right.

Free Parking, and all the usual attractions. If that isn't enough, Brooklands Museum (aircraft and motor racing) is just up the B374; or there's a Bus Collection on the A245 to Cobham; or I'll explain how to get to the RHS Gardens at Wisley (RHS Membership needed on Sundays).

