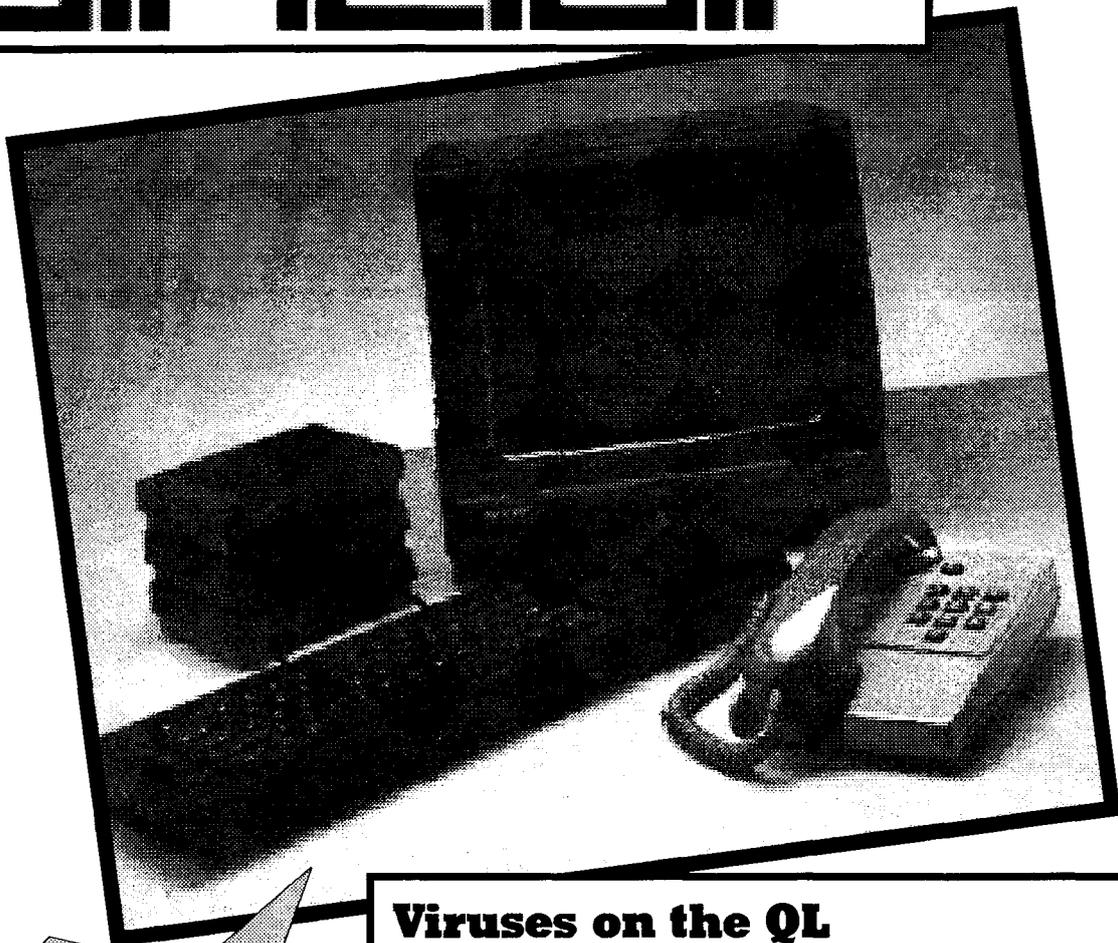


QL Today

Volume 1
Issue 3
Sept./October
1996

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

sinclair



**MORE PAGES
THAN EVER
BEFORE!**

Viruses on the QL

- do we need to be frightened?

What is CONFIG Level 2?

**Reviews: Brandnew SBASIC Pointer
Environment kit & QMenu**

**How to put your QL into a tower
case**

... and much more!

QL Today

German office & Publisher:

Jochen Merz Software
Im stillen Winkel 12
47169 Duisburg
Germany

Tel. +49 203 502011
Fax +49 203 502012
Box1 +49 203 502013
Box2 +49 203 502014

English office:

Miracle Systems Ltd.
20 Mow Barton
Yate, Bristol
United Kingdom BS17 5NF

Tel. +44 1454 883602
Fax +44 1454 883602

Editor:

Dilwyn Jones
41 Bro Emrys
Tal-Y-Bont, Bangor
Gwynedd
United Kingdom LL57 3YT
E-Mail: DILWYN.JONES@BBC.CO.UK

Tel. +44 1248 354023
Fax +44 1248 354023

QL Today is published bi-monthly, our volume begins on 15th of May. Subscriptions begin with the current issue at the time of sign up. Subscription rates are as follows:

Germany	DM 70,-
England	DM 60,- or £25
Rest of the World	DM 70,- or £30

Payment in DM (drawn on a German bank) can be made by either Cheque or Euro-Cheque, Payment in £ (drawn on an English bank) can be made by Cheque. Cheques should be made payable to Jochen Merz Software (German office) or Miracle Systems Ltd. (English office).

Credit Card holders may subscribe by either calling or sending their Credit Card number and Expiry date. Credit cards will be charged in DM (German office) or in £ (English office).

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) in ASCII, Quill or text87 format. Pictures may be in _SCR format, we can also handle GIF or TIF. 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: 15 April	Issue 2: 15 June
Issue 3: 15 August	Issue 4: 15 October
Issue 5: 15 December	Issue 6: 15 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 is copyrighted and all material published remains the property of QL Today unless otherwise specified. Written permission from QL Today is required before the reproduction and distribution of any/all material published herein. All copyrights and trademarks are hereby acknowledged.

Contents

- 3 Editorial
- 3 JMS-Box 1 & JMS-Box 2
- 4 QSpread V1.30 News
- 4 More News
- 5 QL Fair in London - Mark Knight
- 8 Hints and Tips - Floppy Disks - D. Jones
- 7 QXL2 and the falling memory prices - Stuart Honeyball
- 7 Beginner's Basics - correction to part 2 - Stuart Honeyball
- 8 Di-Ren Report on IQLR - Robin J A Barker
- 10 Recursion - Dilwyn Jones
- 18 A QL Meeting in Norway - Albin Hessler
- 18 Viruses and the QL - Tim Swenson
- 20 Harddisks and Backup - H.P. Huyg
- 24 QMenu - Dilwyn Jones
- 30 QL Service List - Part 2 - Robert Klein
- 33 My QL Rebuild - Mark Knight
- 37 Config Level 2 - Jochen Merz
- 42 Writing a Zine on the QL - Tim Swenson
- 43 The SBASIC PE Kit - John Miller
- 46 Bugs 'n' Fixes - Peter Tyler / Dilwyn Jones
- 46 Dynamic Buttons - Per Erik Forssén
- 48 Spy & Master Spy - Norman Dunbar
- 51 Letter Box
- 52 80MHz QXL - Caution! - James Hunkins
- 53 Compatibility - Tony Tebby
- 56 The QL on the Internet - E.C. Herrnsdorf
- 58 Hints for the QXL-Card with MS-DOS - Franz Krojer
- 59 EPSON Stylus Color II - Jochen Merz
- 60 ProWesS SBASIC Interface - Wolfgang Lenerz
- 62 PUZZLER - Win QL Writers Pack
- 62 SuperGOLD Cards from Quanta
- 63 Last Minute News
- 64 Beginner's Basics Part 3 - Stuart Honeyball

Advertisers

in alphabetical order

FWD Computing	61
Jochen Merz Software	25
Miracle Systems Ltd.	19
PROGS - Van der Auwera	39
QBOX USA	47
QBranch	41
Quanta	31
Qubbesoft	45
W.N. Richardson (EEC)	55
TF Services	59
S.J.P.D. Software	29
Geoff Wicks	53

EDITORIAL

To start on a negative note, we apologise for the poor print quality in many of the copies of Issue 2. Jochen got his megaphone out and shouted at the printing company, so I hope it will have been sorted out by now (at least they promised to do it better next time). Another step on the learning curve for this poor little trainee magazine editor!

Letters and questionnaires received seem to suggest that on the whole we're not doing too badly, but not quite managing to please everybody all of the time. Importantly, we seem to have got the mix of beginners' and more advanced articles more or less right. Keep the letters coming, we want to keep improving **QL Today**.

One of the things requested in the letters and survey forms was a letters page. We're pleased to say that there is one in this issue! Thank you to everyone who's sent in a survey form. I'm busy going through them all and I'll write an article about the results in our next issue. It looks like everyone wants as much news about the QL as possible, so we'll do our best to oblige! Software authors, hardware and software traders, send us your news! The readers want to know about your products!

We have recently seen a flurry of activity on the QL hardware front, with all sorts of devices being launched or announced. It is to be hoped that the software scene will not lag behind. I very much hope that the Aurora graphics card will lead to more new graphical software for the QL once authors have familiarised with it. Keep reading **QL Today** for the latest news!

You can reach me by E-Mail now. The address is: Dilwyn.Jones@bbc.co.uk

Dilwyn Jones

IOMEGA ZIP for QXL Users

Bruce Nicholls informs us that has had an Imoega Zip 100MB supper-floppy operating with his QXL. This device plugs into the parallel port of a PC and looks to software like a mini hard disk. Formatting from SMSQ produces a QXL.WIN file on the 100MB removable disk, which can be exchanged and referred to as WIN2_ (or WIN3_, depending on how many drives you have on your system).

So on the QL scene it seems we can use the EZ-135 system for QLs via the QUBIDE system and the Iomega Zip is suitable for use with the QXL, as long as your PC has a suitable parallel port. ■

JMS-Box 1 & JMS-Box 2

By the time you read this JMS-Box 1 has been converted to run under Phil Borman's excellent PBOX bbs software. This results in many advantages for me, the sysop,

- Maintenance of the bbs is much easier for me (and it is only one system now).

- The utilities for handling the bbs are much better.

and mainly for you, the user:

- both lines will be served at the same time. If you have a V.34 modem (28.800) then better use 0203-502014. If you have a ZyXel U-1496 then you better to use the 0203-502013, as this will give you a 16.800 connect (and the 502014 will handle 14.400 with the ZyXel.

- both lines will handle the same data. All uploaded files are available in box 1 and box 2, as well as messages and everything else. Messages written in box 2 can be read in box 1 and 2 and vice versa. Both boxes will "know" you under the same name and password.

- The menus are better organised.

- Much better file handling. Tagging is easier, download via ZMODEM makes life easier and transfers faster and more reliable.

- Proper full-screen ANSI-menus are possible as well as cut-down scroll-type menus, whatever you prefer.

- Proper full-screen editor is available as well.

... and much, much more. Why not have a look!?

Ah, and if you forgot to register in box 2 and you find that box 1 does not recognise you anymore, then don't worry: register now and you're "in" again.

Once again, please remember the following when you used Box 1 in the past and you dial into Box 1 or 2 now:

- Set your terminal program to ANSI, otherwise you will see lots of funny characters mixed into the text, e.g. ESC [40 m or so.

- Set your protocol to ZMODEM! ■



Jochen at work

QSpread V1.30 News

As you probably know, the last update on QSpread happened in December 1995. Oliver Fink, the author of QSpread, did not find any more time to do more work on QSpread, but generously offered to pass the sources of QSpread to me so that development can continue. He was actually very busy working outside of Germany so that he could not even pass the sources to me. Now, being back, Oliver sent me the sources.

We are now in the position to maintain and update QSpread. In order to fulfil most of your wishes I do need your help. All the bug reports which you sent to me during 1995 and beginning of 1996 were passed on to Oliver, and I did not keep a copy, unfortunately. Oliver did send me the sources, but he did not return all the letters from the customers which I passed to him.

I would like to ask you, the QSpread user, to write to me about any problems you have had with QSpread, but only referring to V1.29 (this is the most recent one and the one we have the sources of). Please do not report bugs of V1.28 or before, as they might have been fixed. It is quite a lot of work to go through Megabytes of assembler sources not produced by oneself, therefore it would be nice if time and work was not wasted by trying to fix bugs which are already fixed.

You do not need to write for the following problems and wishes:

- Split window into sections which are smaller than the maximum cell width (panning will crash).

- Formulae entry longer than one line. That's done in Version 1.30.

- All strings starting with pi (like piranha) to be evaluated into 3.141.... Fixed in V1.30 too.

- INSTR and string comparison (yes, we know you would like to get this). As we cannot tell how difficult it is to implement it, we'll leave it for later and start with bug fixes.

- If condition: text-result (would be nice to have this).

- Being able to reference to text-cells. V1.30 now copies the contents of text cells.

A new QSpread demo exists which offers all features found in V1.30, except for that you cannot print, save and export. This is far less cut-down than the previous QSpread demo and up-to-date as well, so that you can have a look at all features.

Version 1.30 is available now and not only contains a number of fixes but also some improvements. The new version, together with a manual supplement, is available from Jochen Merz Software for DM 16,- (please return the master disk, as usual).

For comments on QSpread please write to Jochen Merz Software. ■

More News

COMING SOON:

THE NEW-LOOK THESAURUS

Version 3.00 of QL-THESAURUS will shortly be released. This is a major revision of the program incorporating new features requested by users. These include:

- Backwards scrolling of both word lists and word groups

- The ability to "pick up" the chosen word for entry into a word processor

- Simplified antonym searching

- User configurable ink and border colours

- A revised data-base.

The upgrade terms will be published in the September JUST WORDS! advertisements.

Geoff Wicks, Bertrand Russellstraat 22, 1097 HL Amsterdam, Netherlands.

QBRANCH News (Roy Wood)

QBranch have announced their "first birthday" present to the QL community! As part of their efforts to get older QL software titles re-released, they have arranged with Triptych Publishing that three of the early QL business packages, called Decision Maker, Entrepreneur and Project Planner will soon be released as PD software, courtesy of QBranch. Roy Wood is at pains to point out that Triptych Publishing will not be providing support for these programs, that will only be available from QBranch. So all enquiries to QBranch ONLY please!

QBranch, P.O.Box 7, Portslade, East Sussex, BN41 2ND, Great Britain. Telephone (+44) 1273-386030 (or 01273-386030 in Britain).

QUBBESOFT P/D News (Ron Dunnett)

Qubbesoft are pleased to announce price reductions on some of their hardware range - the Qubide interface itself is down in price from £65.00 to £55.00, and the Syquest EZDrive 135 IDE Removable Medium is down from £165.00 to £135.00. Ron Dunnett claims that they are "selling like hot cakes", so it would seem that Qubbesoft have Dunnett again (sorry, Ron, I couldn't resist that one!).

To help boost sales further, Qubbesoft P/D have announced two EZ135 cartridges packed with clipart and PD software. The first is called 'EZ Clipart 1' (pronounce EZ with an American letter Zee and you'll understand the name) and costs just

£25.00 for a cartridge full of Line Design clipart!

And EZ PD Collection consists of Qubbesoft's entire PD software library (as per their issue 22 catalogue), including such gems as the C68 compiler, for example, also for only £25.00. It says something about QL software that Qubbesoft's huge library will all fit on a single 135MB cartridge - certain other computers may need 135MB PER PROGRAM! If you prefer to supply your own cartridge, Ron will copy either collection onto that for a copying fee of just £10.00.

A third EZ cartridge collection called EZ-Literature will be available in September, comprising a large collection of classic literature, from classical Latin to Shakespeare, from Sherlock Holmes to religion, a complete library on a single cartridge! Should keep QL users busy during those cold winter nights! This cartridge will also cost £25.00.

Ron Dunnett also announced that he and author Dave Walker have been working on an upgrade to the QL and PC file transfer program, Discover, to enable that program to read files directly from a PC formatted EZ135 cartridge onto QL floppy disks. Of course, a Qubide interface is required for this on a QL, and you may also need a ROM upgrade for the Qubide from Qubbesoft P/D.

PM data Address change

P. Monstad of PM data informed us that the new address and telephone number is:

PM data Tel./Fax (+47) 52783413
Søndenålia 27
4200 Sauda
Norway

Page Designer 3 V1.17 Update

Quo Vadis Design have announced that Page Designer 3 is now at V1.17. Author Barry Ansell has added a MOVE item to the program, allowing its outline to be moved around the screen on a QXL or Atari ST-QL, for example, where the display is larger than the normal 512x256 QL screen - at last! As yet, it is not possible to change the size of the area of screen used by the program, though the move item itself is bound to be a very useful facility as more and more users have access to higher resolution displays. V1.17 of Page Designer 3 has been tested on a prototype Aurora graphics

card and found to be compatible as of early August

1996, according to Bruce Nicholls.

Disk upgrades for minor version changes (e.g. V1.16 to V1.17) are free provided return postage is provided. Only disk 1 needs to be returned for this upgrade. A major upgrade to PD3 needing a new manual (check first with Quo

Vadis Design if you have an older version) costs £15.00 if applicable.

Quo Vadis Design,
57 Shaftesbury Road,
Romford, Essex, RM1 2QJ,
Great Britain
Tel/Fax (+44) 1708-755759
email: 101366.3534@compuserve.com

QL Fair in London

Mark Knight

On Saturday 22 June there was a QL fair held at Saint Quintin's Avenue in London, organised by Tony Firshman of TF Services. The show started at 10:00 a.m. and finished at 5:00 p.m. and traders in attendance were: Di-Ren, Miracle Systems Limited, QUANTA, QUBBESOFT PD, Quo Vadis Design, Q-Branch, TF Services and W N Richardson & Co. Ltd.

I arrived right at the start and for a while it seemed there were going to be very few QL enthusiasts in attendance, with the traders outnumbering the customers. To everybody's relief the hall suddenly seemed to fill up around lunch-time and the show looked well attended compared to similar events I have been to. The atmosphere, as at most QL shows, was friendly and sociable with most traders on first name terms with many customers and with each other.

There were many new QL products being shown as well as lots of updates for old favourites. On the QUANTA stand it was even possible to obtain

updates for the XCHANGE suite, something I thought would have died through lack of interest some time ago. After enquiring I discovered that there are hundreds of people still actively using Psion XCHANGE and naturally they want the bug fixes and enhancements on offer to QUANTA members. This shows how wrong you can be and also how well designed XCHANGE was, for it is still serving some people well more than ten years after it was written. QUANTA were also selling the Super Gold Card, probably the best selling piece of hardware at the show, though Ron Dunnet of QUBBESOFT PD reports that his QUBIDE hard disk interface continues to be a steady seller.

The main items of interest for me were the Di-Ren PC-AT keyboard interface, the Super-Hermes board from TF Services, the new version of the QXL card from Miracle, and finally the long awaited Aurora Graphics card from QUBBESOFT PD.

The Di-Ren keyboard interface is the cheapest way to give the QL a really good keyboard and works extremely well. If you have not tried a QL with an upgraded keyboard or you couldn't afford the expensive products previously available I would recommend trying this product at a show or just go ahead and buy it. It is very simple to fit and it just works, there is no fussing with software or new keyboard drivers as it is all built in to the interface.

Super-Hermes is a possible alternative and although this is much more expensive it provides a great deal more than a keyboard interface. Using Super-Hermes you can provide your QL with better behaved ser1 and ser2 as well as ser3 to ser6, with one of the latter being capable of baud rates up to 57,600 baud. Another advantage with Super-Hermes is that the QL serial ports can operate with different baud rates for each serial port and even with different rates for input and output. With up to 6 serial ports one can be dedicated to a mouse, the high speed one to a modem and you can still use a serial printer or a Miracle Centronics interface in one of the original ports. You also have a 1.5k battery backed memory which can be a special BOOT program for your QL to make it boot from a network or do some other startup trick. Looks excellent, ask for details from TF Services if it interests you.

The long awaited and much delayed Aurora Graphics card was at the show being demonstrated running the pointer environment at 1024 by 768 pixels on a QL with an SVGA monitor connected. This was a prototype only and Ron Dunnet admitted that the final shipping product will not be available until September or later, with September being the target. As well as the high

resolution 4-colour modes (including but not limited to 640 by 480, 800 by 600 and 1024 by 768 pixels) there will be others with more colours. There will be 8, 16 and 256 colour modes in various resolutions with the 256k of screen memory onboard and Aurora will provide such an enhancement to the QL that it will be enormously popular. This card is also one reason why the Super Gold Card is currently popular as to take advantage of the new screen modes a Super Gold Card is needed as well as Aurora.

As is usual at these events there were some talks, with Ron Dunnet talking about Aurora, Tony Firshman talking about Super-Hermes and Stuart Honeyball talking about the new version of the QXL card. This is a card to plug into a PC to make it capable of switching to a QL compatible mode using the hardware on the card. The old QXL card had a clock rate of 20Mhz while the new one runs at 25Mhz, giving the already amazing speed a 25% boost. The cut-down version of the 68040 that is used on the QXL card runs Miracle's SMSQ operating system software with SBASIC, an enhanced SuperBASIC compatible interpreter. The combination of the fast processor and the new QDOS compatible operating system makes the QXL very fast indeed, and it is able to use PC graphics cards giving up to 800 by 600 pixels on an SVGA monitor in 4 colour mode. While there I tested some of my software on a PC fitted with the card and was stunned by how much faster it was than my Gold Card system.

Attendance at the talks was good especially the Aurora graphics card which emptied the main hall. I got the impression that although there were a lot of people at the show over the course of the day not that many were buying, perhaps holding onto their money until Aurora becomes available. This might have bothered traders though I didn't manage to talk to them about it and my impression could have been a mistaken one. I will say that the atmosphere was good and there was hardware and software being sold, with QUBIDE, Super Gold Cards and the Di-Ren keyboard interface being deservedly popular in the hardware department. In software the Pointer Environment software seems to rule now and lots of this was being sold by Q-Branch and Quo Vadis Design.

I was glad I went, coming away with several bits of software and a couple of contacts who will be able to help me test some software I am writing when it is ready for final testing. I also enjoyed myself which is surely a good reason to go to such shows, as well as the chance to see what you are buying in advance. Thanks are due to Tony Firshman for organising this one! ■

QXL2 and falling memory prices

Yates, U.K. - Stuart Honeyball

The price of memory has been tumbling over the past few months in contrast with the last 2 or 3 years when it was gradually increasing with a rather steep rise last year. It was at the peak that I decided to stop making the QXL because the production cost just about matched the retail price. The price of memory (DRAMs) is now about 1/4 of what it was a year ago. In the light of this it only seemed fair to pass this benefit on so the QXL2 is now available only with 8M RAM but at the old 4M price, i.e. £295 in the EU or £255 for the rest of the world.

So why do memory prices fluctuate so wildly? It's obviously due to the laws of supply and demand so why does the supply differ from the demand so markedly? The reason lies in the long time delay between the taking of a decision to build a manufacturing plant (known as a wafer fabrication facility or "fab") and the start of production. This time is typically 2 years. A fab is designed to churn out a few million chips per month and may cost \$1,000,000,000 to build but will produce the chips it was designed to produce for only 5 years or less. The cost of the chips is heavily dependent on the initial investment so when demand is slack you would save little by reducing output and when demand exceeds expectation you can't increase production to satisfy it. With several manufacturers around the world all making the same thing prices ride the roller-coaster. The only option open during a slump is to switch from making DRAMs to something else but there may be no real alternative products that sell in quite the same quantity. Note that Intel who pioneered the DRAM quite that particular race and make highly profitable single sourced '86 processors instead.

Experiment has shown that the original 20MHz QXL's can reliably be run at 25MHz. This is accomplished simply by replacing the 40MHz oscillator with a 50MHz one. For those of you who have an original 20MHz QXL and would like it to go at 25MHz then bring it along to one of the meetings and I will endeavour to upgrade it for you there and then, free of charge. ■

Beginner's Basics - correction to part 2

Yates, U.K. - Stuart Honeyball

Thanks go to John Hall for kindly pointing out that last issue's article had some errors in it - sorry. These were in the right hand column on page 14.

Firstly there was a rather trivial error at the top of the column where the procedure name is given in line 10 as `Multiply` yet at the end of the procedure definition, line 30, it has changed to `Mult`. This highlights one of the small differences between the new interpreter `SBasic` and the original `QL` interpreter `Superbasic`; in `SBasic` this is allowed whereas in `Superbasic` it gives an error. The author uses a `QXL` running `SMSQ` which incorporates `SBasic`.

The other error was more serious and concerns the definition of the function `Product` written in terms of the procedure `Multiply`. The correct version of this function is:

```
100 DEFine FuNction Product(x,y)
110 LOCal Product_x_y,Temp_x
120 Temp_x=x
130 Multiply Temp_x,y
140 Product_x_y=Temp_x
150 RETurn Product_x_y
160 END DEFine Product
```

Last issue's function definition looked as though the procedure `Multiply` was returning a value - something a procedure does not do.

You may be wondering why I have included the variable `Product_x_y` when the lines 140 and 150 could be condensed into just a single line:

```
140 RETurn Temp_x
```

the reason is that the name `Product_x_y` has a more relevant meaning where it's used than `Temp_x`, and vice-versa. Making programs readable may be at the expense of efficiency but they are more likely to work and are easier to write and edit. ■

*** THE SURVEY WINNERS ***

I couldn't even afford an old bucket, therefore I numbered the surveys and used the BASIC statement `PRINT RANDOMISE(1 TO maximum)` to draw the winners. Here they are:

The main winner is: Mr. Lennart Forssén, Sweden. Congratulations, QMENU is on its way to you!

We had so many replies, therefore 5 *QL Today* Calendars also go to:

Stephen A. Hall, United Kingdom. John Miller, United Kindgom. John Wakefield, United Kingdom. Rainer Fieseler, Germany. Charles Fothergill, United Kingdom.

HINTS AND TIPS - FLOPPY DISKS

Dilwyn Jones

Q. I have a 5.25 inch disk drives which formats disks to 720 sectors (360 kilobytes). Is it a 40 or 80 track drive?

A. It could be either, as the information you provided is not sufficient to distinguish the two types of drive. 720 sectors could indicate that the drive is either (a) 40 track double sided, or (b) 80 track single sided, since both these types return the same capacity when formatting disks. There is a way to distinguish between them, as long as your disk interface supports single sided disk formatting, using a '*' character as the 11th character of the medium name in a format command. Place a new disk in the drive and enter this command:

```
FORMAT '1234567890*'
```

(and press ENTER). This will format the disk, forcing it to be single sided format. If your drive is a single sided type, this will return the normal number of sectors. If it returns half the normal capacity, your disk drive is a double sided variety. Most users with 720 sector drives tend to assume that these are 80 track single sided types, but experience has shown me that they almost always turn out to be 40 track double sided types. This is very important when ordering programs from a software supplier. When you ask for 720 sector format disks, you must correctly specify if your drive is 40 or 80 track, as your drive will fail to read the wrong format. If your supplier sends the program to you on a 40 track double sided disk, your single sided 80 track drive may fail to read it, and similarly, if you receive an 80 track single sided disk, your 40 track drive won't be able to read it, but a 40 track single sided disk of 360 sectors will often work correctly on both types of disk drive.

Q. Some of the disks I have received from QL software traders report a very low sector count, but the program seems to run OK. Are the disks faulty, or is the trader cutting corners with the disks?

A. If the program copies and runs without errors, then all is probably well. Some traders cut down on the time it takes to format floppy disks by doing what is called a Short Format. This means that they calculate how much space is needed for the program and format the disk to just that capacity. This may mean using single sided

formatting as detailed above, or just formatting to a give number of tracks, using the FLP_TRACK command (see your Toolkit 2 or disk interface manual for more details). For example, the supplier has determined that the program needs approximately 250 sectors on a DSDD disk, plus a little bit of spare space to allow for minor revisions in future revisions in case you return the disk for an update one day. So he/she decides that 270 sectors is a suitable figure. Knowing that there are 9 sectors per track, the calculation is the number of sectors required divided by 9. Thus $270/9 = 30$, so 30 tracks need to be formatted. This could either be 30 tracks on one side only, or 15 tracks per side. So, either of two commands could be used:

```
FLP_TRACK 30 : FORMAT 'FLP1_PROGR_NAME*' :  
REMark single sided
```

```
FLP_TRACK 15 : FORMAT 'FLP1_PROGR_NAME' :  
REMark double sided
```

This could help QL Today contributors. We never miss a chance to encourage people to write for the magazine! When sending your disk to us, just format it for the minimum number of sectors required, so save yourself some time! ■

Di-Ren Report on IQLR - 30.5.96

Robin J A Barker - Di Ren

This document is a report of all the details currently known by me in relation to the unexpected cessation of trading by IQLR, its trading practices and my involvement as the subscription agent. I feel it necessary to produce this document in order to clarify my position in this unfortunate incident.

Earlier this year, Bob Dyl asked me to act as a subscription agent for IQLR, replacing the service previously supplied by Stuart Honeyball of Miracle Systems. After some consideration I agreed to act in this capacity.

The function of the agency was broadly divided into distinct operations.

First, to accept, acknowledge and submit subscriptions directly to Bob Dyl's UK bank account. Secondly, to receive and despatch several hundred magazines to the UK, Europe and as later transpired, all world-wide destinations with the exception of the United

States and Canada (postage from the UK is much cheaper than America). For performing this function I was to receive £3.00 for every Credit Card Subscription received. Cheques and cash payments did not return a commission.

Mail Order Credit Card transactions are charged to myself at a rate of 5% per transaction. Other costs involved were acknowledgement postage and paperwork, not to mention my time. It will readily be appreciated that as many of the subscriptions were Cheques and Cash, there was simply no cash incentive to perform this agency task. I accepted the job as subscription agent for IQLR simply because I felt it would provide a service to the QL community.

On Monday 29th April 1996 I received an Email from Bob Dyl's family informing me he had suffered a heart attack and that due to his condition considered the continuance of IQLR to be unlikely. Upon receipt of this communication I immediately sent an Email to all database subscribers and informed several other people of this development. Since this time, several pieces of information regarding the financial status of IQLR and the subscription database have become known and regrettably these are not encouraging.

Firstly there is the question of the subscription database. I, and many others, were led to believe there were around 1,500 subscribers to IQLR. It would now appear that at any time, the total subscription base never exceeded around 400. *[To be precise, the database contained 366 subscribers plus 48 subscribers through JMS - Editor].* The basis for this conclusion is as follows. The current database now in the possession of Stuart Honeyball and Jochen Merz is around 360. The subscription agent was responsible for posting around 300+ magazines to all places world-wide with the exception of the United States and Canada. This means there would have to be 1,100 subscribers in the USA and Canada. We have no evidence to support more than 80 subscribers existed in these countries.

Next comes the financial status. Myself, Stuart Honeyball and others were informed by Bob Dyl that a special account was in operation to protect subscribers money in the event of a failure of IQLR. This account was designed to release the portion of the subscription fee relevant to the current magazine issue. Thus, if

IQLR failed, the balance of subscription fees would be intact. There is no current evidence to prove the existence of this account. In any event, IQLR has no funds to repay outstanding subscription fees and additionally leaves traders such as myself with substantial advance advertising fee losses (£250.00 for me).

Bob Dyl currently maintains many subscribers have been repaid and their names purged from the database. As yet, there is no evidence to support either of these claims.

The total income of IQLR since the beginning of the year is estimated to be well over £4,000. This includes 151 subscription fees taken by myself and traders advertising fees. Note the high proportion of subscription fees in comparison to the database total submitted in the last few months.

I should make it clear here that there is no reason to suppose IQLR would not be functioning as normal if it had not been for Bob Dyl's heart attack. Postage and production costs would have been nominal and easily maintainable from ongoing subscriptions.

Since the closure of IQLR, a new magazine, entitled QL Today, although not linked to IQLR for reasons of liability, has been started by Stuart Honeyball and Jochen Merz to fill the gap left by IQLR. Jochen Merz is the publisher. Stuart Honeyball has pledged to replace issues due from IQLR by the new magazine at no charge for those who subscribed through him (via Miracle Systems - mainly last year). IQLR subscribers who wish to continue with QL Today who subscribed through me (from the beginning of 1996) and other methods, will only pay half (or less) the proportionate QL Today subscription fee of £25.00/£30.00 for outstanding IQLR issues. Adverts are not discounted. *[In fact, they are - Editor]*

My personal feelings about this set of circumstances are mixed. Whilst I was very sorry to learn of Bob Dyl's illness, the apparent mismanagement of IQLR's funds and subsequent close down has turned what I believed to be a favour to the QL community, into something of a nightmare. Worse, I am now informed that other members of the QL community guessed all was not well with IQLR some time ago. It's unfortunate this did not filter through to me at the beginning. ■

Recursion

Bangor, Wales - Dilwyn Jones

Recursion is one of those subjects that most people with a little programming experience will have heard about, but few will fully understand. Ask those who have heard about it to define Recursion, and they'll probably say something like 'a routine which calls itself'.

That's a reasonable description, though there is a bit more to it than that. If a routine just calls itself continuously (like this short example) it will just keep going forever, or until the computer runs out of memory.

```
1000 DEFine PROCedure MENU
1010   MENU
1020 END DEFine MENU
```

In addition to being a routine which calls itself (or is described in terms of itself), a recursive routine must have a mechanism for ensuring that it tests for when it must call itself and when it must return from the procedure. As recursion is a difficult subject to explain clearly and briefly in simple words, I will present a few example routines to demonstrate uses of recursion in programming in superbasic.

Most programmers who understand and use recursion are able to spot tasks that require the use of recursion almost by instinct, but few will be able to explain precisely how they arrived at that decision. Nonetheless, once mastered, recursion can be a powerful and efficient programming tool.

Recursion can be thought of as a process where the overall task is broken down into smaller, more manageable tasks in an ordered form. Thus the solution to such a task or problem is described in terms of solutions to similar smaller problems. At this point, many people's brains tend to blow a fuse trying to follow the explanations, so we'll consider a very simple example to try to illustrate what this means.

Suppose you have an army. The opposing army has the same number of soldiers as you, so you don't like the idea of a head-on battle where there is a high risk that many of your best soldiers may be killed or injured. So you decide on the strategy of breaking up the enemy into smaller groups and taking them on group by group to simplify the attack. This could be called 'divide and conquer' - reducing the problem by splitting up into several smaller sections until it is all finished.

```
define procedure Conquer (groups)
  if all divided then return
  split up into another section (groups-1)
  conquer that group
end define
```

Take the simple example of printing a series of numbers from 1 up to the value 'n' (any whole number). Normally, we'd approach such a problem simply by putting a PRINT statement in a FOR loop. But by writing it as a recursive routine, what we are attempting to do is so simple that we can concentrate on the theory rather than the problem itself. We'll start by breaking down the process into simple component tasks. We have to print numbers, each of which is one larger than the one before it.

```
define procedure print_number (value)
  if we have reached the limit, end this
  procedure
  print the number to the screen
  print the next number (i.e. value+1)
end define
```

That was the breakdown. Now we can code it into a proper basic procedure.

```
10 CLS : Print_Number_Up_To 10
20 :
100 DEFine PROCedure Print_Number_Up_To (value)
110   IF value = 0 THEN RETURN
120   Print_Number_Up_To value-1
130   PRINT value
140 END DEFine Print_Number_Up_To
```

The test in line 110 checks if it is time to stop going round in loops. Line 120 calls up the procedure 10 times, decrementing the value by 1 each time, just like a counting loop. This goes round 10 times (until value=0). This has effectively set up the same procedure to print 10 different parameters, each call has a parameter which is different by the amount stated from the previous 1. Once the 10 values been calculated, there are 10 procedures, each ready to print its own value. After the tenth definition, when the value of the parameter "value" has become 0, the RETURNS start to come into effect such that there are 10 returns to be made. Remember that each return goes back to the most recently defined procedure call, so the printing happens in the reverse order to that in which the numbers were defined.

Line 10 first calls the procedure with the highest value (10 in this case). For this first call, the value is 10, so the return is not executed. The program now comes to the line containing the call to print a number up to the value specified. The next number to define is 9, so the procedure is called again with value-1, or 10-1. This takes us back to the line

containing the RETURN statement. We still haven't reached 0, so we call the procedure again, with value-1 once more. This time the value was 9, so the next number is 9-1 or 8. This goes on until 0 is reached, then the returns start taking effect. The most recent procedure had a value of 1, so this gets printed first. Then we finish that procedure call

line	action
10	call to Print_Number_Up_To with value = 10
100	procedure definition, value of 10 given to variable called "value"
110	expression checks if "value" has reached 0
120	not having reached 0, we call the procedure again with the value of 10-1, or 9.
100	back at the procedure definition. This time, value=9
110	check to see if we have reached 0. No we haven't.
120	call procedure again with value-1 (9-1, or 8)
100...	and so on until value=0
110	value = 0, so we return from this procedure to where we were in the previous procedure call (remember, we last called from line 120, with a parameter value of 1
130	print the current value (1)
140	end define - end of this procedure call so we return to the last one, where we had made a call from line 120 with a value of 2. Returning to the next statement after line 120 which contained the call
130	print the current value (2)
	and so on until all 10 numbers are printed

Of course, this approach to printing numbers is like taking a sledgehammer to crack a nut. But I find it does help to explain the technique, because it is a short and fairly simple to follow routine, short enough for you to follow it step by step, writing out each step on paper to help you to understand it.

Recursion is often used where a similar routine would be needed for a series of operations or calculations where there exists a slight difference in each call to the routine, but the routine is sufficiently able to cope with these variations and the end conditions are sufficiently well known to be clearly definable. This is very important in recursion - the test for when a routine needs to stop calling itself is vital to ensure correct operation and successful termination. This next example is not a working program (do not try to run it as a basic program!), but it does help to show in a much simplified way the general workings of a recursive routine.

```
DEFine PROCedure Recursive_Routine (parameter)
  LOCAL list of local values
  ...any code required here
  IF end condition THEN return
  Recursive_Routine with variable parameters
  actions to take while unravelling the
  recursive calls
END DEFine
```

and go back to when value was 2 - we go back to just after the line which called the procedure, so the next thing after each return is the PRINT statement which prints the 10 values in turn. I hope that made sense. In case it didn't, here is a trace list of where the program goes:

The routine establishes any variable data required each time round the routine, checks if the end condition is met or not, then calls itself again with the slightly differing data as required. One thing which may not be obvious is the need for local variables in the procedure. If you need to create temporary variables to hold values which differ during each call of the procedure, you need different variables for each call so that each call does not destroy the values we worked so hard to set up.

At this stage, what may not be obvious is just why anyone would want to use recursion. When I first struggled to come to terms with recursion years ago, it seemed to me as though recursion was just a technique invented by someone very intelligent who wanted to impress anyone who read his/her code by writing an otherwise simple piece of code in a way no one else could possibly figure out. It always seemed as though there was a simpler (but usually longer) way of doing things with less obscure code. In a way this is true - there is usually a non-recursive way of doing the task implemented as a recursive call, but it may be longer, less elegant and less efficient (not to mention slower). The art lies in being able to spot those cases where recursion would be of benefit to the programmer. To some extent, this comes with experience and with a knowledge of 'standard' applications and routines where recursion is used. In terms of thought processes, you have to spot repeating patterns in what needs to be done, or

need to spot how a task can be split up and done more efficiently in sections (the "divide and conquer" approach). I remember reading in a text-book some time ago that recursion is an act of faith - since it is very difficult to follow exactly what a recursive routine does, it is hard to be sure that the routine will perform exactly as you expect it to. When you wrote the routine yourself, that is not too bad, as you knew what you were trying to do at the time. But if someone else wrote it, your brain may well get to the "blow a fuse" stage while trying to follow what the routine is doing!

I will present several example recursive routines, some using procedures and some using functions. I hope that at least one of these routines will help you to see how recursion works, and why it can be such a useful programming technique.

(i) Factorials. This mathematical routine is one of the classic examples of recursive programming techniques. It involves calculating the total of a series of products of numbers from 1 up to n. n! is pronounced as 'n factorial', the product of all the whole numbers from 1 to n

$$n! = n * (n-1) * (n-2) \dots * 2 * 1$$

(ii) Sub directory reading. The routine enables one command to read through all sub-directories on a hard disk (or high capacity floppy disks). It only works on systems with 'level 2' directories (so-called hard directories, created with the MAKE_DIR command from basic, for example).

(iii) A fast string sorting routine, called appropriately a 'Quicksort', where the data is repeatedly split into small groups, each of which is sorted, and the data further divided and sorted until the whole list ends up in order. The sorting routine itself divides the list to be sorted and calls itself to sort the small groups created.

(iv) A graphics flood fill routine, where irregular shaped outlines can be filled with colour. It is not as fast as similar routines you will find in commercial programs, but performs adequately. Unfortunately, a toolkit extension is required in order to read the colour of pixels already on the screen. I have used my own extension, PIXEL% x,y (where x and y are co-ordinates across and down the standard QL 512x256 screen), but equivalent routines such as the one from Simon N. Goodwin's DIY Toolkit collections may also be used as long as you change the name of the extension accordingly, and supply the correct parameter. There are in fact two routines, a very simple one which illustrates the basic principle (it actually works, but is not very efficient), and the second which includes an element of buffering (or stacking) to reduce call overheads and make the

routine work more quickly than the first example.

(v) the Towers of Hanoi puzzle, where there are 3 spikes and 3 rings on the first spike. You have to transfer all the rings from spike 1 to spike 2, subject to a few rules: one ring only to be moved at a time, and no ring may be below a larger ring on a spike. Realistically, it doesn't take much to work it out in your head, but this routine demonstrates how to use recursion techniques to allow the computer to elegantly solve the puzzle. The routine helps you to solve this puzzle for a given number of rings, not necessarily 3. The principle is to always end up with the next largest ring on the destination spike while using the spare peg to build up a stack of those rings larger than the next ring to go on the spike.

1. FACTORIALS

This routine lets us calculate the product of a series of numbers up to a given value. The term "n!" (pronounced n-factorial) is used to represent this product.

$$n! = n * (n-1) * (n-2) \text{ etc to } \dots * 2 * 1$$

So 5! is 5*4*3*2*1, which is 120. This is not easy to code as a FOR loop, since the values do not form a tidy sequence. It could be coded up as follows:

```
INPUT'Factorial of > ';n
LET factorial = 1
FOR number = n TO 1 STEP -1
LET factorial = factorial*number
END FOR a
```

and this is a perfectly legitimate way of writing such a program. But let us see how we can do this by recursion. When the calculation is written down in stages as above, we can see that it involves several stages of multiplying the existing total by the next number in sequence. So the sequence involves (i) calculating the next number, (ii) updating the total value by multiplying the total by the next value until we get down to the last value, 1.

```
100 INPUT 'Factorial of > ';n : PRINT Factorial(n)
110 DEFine FuNction Factorial (num)
120 IF num = 1 THEN RETurn 1
130 RETurn num * Factorial (num - 1)
140 END DEFine
```

This is actually longer than the other routine, though this is partly due to the function definition. To avoid having a running total, the function returns the first number multiplied by each descending value defined each time by calling the function with the same value less one, thus

building up a running total as an expression fairly similar to the example given above. There is a special case when the countdown reaches 1 - we have to return just plain 1 (line 120) rather than calculating the next in the series (line 130).

2. SUBDIRECTORY READING

This routine may only make sense to those who have subdirectory facilities on their QL. This includes Gold Card, Super Gold Card, QXL, and Atari ST-QL emulators, plus Trump Cards and SuperQBoards fitted with a level 2 upgrade roms from Jürgen Falkenburg or Jochen Merz.

Here is an outline of how we'll build the routine.

```

procedure extended_dir (drive,directory)
local variables
open directory
if failed, return
read back all names in directory
  have we finished? if so, close directory and exit this loop
  fetch next filename
  is this a directory name?
    yes, do extended dir on this directory -->> recursive call
    no, just print the filename
  end if
end point of loop reading back names
end of procedure

```

and this is how I chose to write the routine in Superbasic:-

```

100 REMark extended dir of all sub-directories
110 Extended_DIR 'win1_',''
120 :
130 DEFine PROCedure Extended_DIR (drive$,directory$)
140   LOCAL loop,ch,d$,fp,n$
150   ch = FOP_DIR (drive$&directory$) : REMark open channel to directory
160   IF ch < 0 THEN RETURN : REMark unable to open directory
170   fp = 14 : REMark file position in directory for filename
180   REPEAT loop
190     BGET #ch\fp : IF EOF(#ch) THEN CLOSE #ch : EXIT loop
200     GET #ch,d$ : REMark get directory entry name
210     IF LEN(d$) > 0 THEN
220       REMark a directory length of 0 may be a deleted file
230       BGET #ch\fp-9 : REMark file type byte
240       IF CODE(INKEY$(#ch)) = 255 THEN
250         REMark this name is a subdirectory, so we need to DIR this
260         REMark if you want directory names printed, add this
270         REMark PRINT d$;' ->'
280         Extended_DIR drive$,d$
290       ELSE
300         PRINT d$
310       END IF
320     END IF
330     fp = fp + 64
340   END REPEAT loop
350 END DEFine Extended_DIR

```

When you do a DIR of a floppy or hard disk containing directories, you get the files in the current directory and the names of any subdirectories shown as the name followed by a '-' symbol. To get a list of files in one of those subdirectories, you have to do another DIR. Then if that list contains more subdirectories you have to do more DIR commands on those and so on until you have the complete list. So hopefully you will see that the main task, of getting a complete list of all files in all directories consists of looking at each directory in turn, starting from the root specified. So we keep going down any path until finished and return to start another path once we have finished the first and so on until we have listed all files.

Note that the routine as it stands does not print the directory names, only the filenames. To print the directory names, like a DIR command, remove the REMark statement from line 270. Remember that the call to the routine must be made with separate drive and directory names, since the drive name is used separately later on in the routine.

3. QUICKSORT ROUTINE

The quicksort routine is one of the fastest sorting routines which can be programmed from Superbasic. I have encountered two variations. The first is simpler to use, but not as fast as the second variant, which has certain special requirements which make it a little less convenient to use.

Quicksorts work by shuffling pointers to sections of the data to be sorted and sorting these smaller sections. This reduces the total number of passes through the data which slows down lesser sorting routines such as the Bubble Sort (I hope to

deal with sorting techniques in a future article in QL Today). I have already used the phrase 'divide and conquer' to describe how some recursive routines work and this is also true for this sorting routine.

This is not an easy example of recursion to follow - if you want to try, assign a small array of data to be sorted and try to trace by hand the flow of operation of the program, printing out values of the main pointer values as it goes.

This sort is designed for strings; to sort numeric arrays simply change the strings in the routine (the names ending with \$) to numeric variables (integer, ending with %, or floating point variables and arrays as required). The second parameter is the subscript of the lowest array element to be sorted, usually 0, and the third parameter is the subscript of the highest element to be sorted, NOT the total number of items (i.e. usually total number of elements-1 as long as the first subscript is 0).

```

100 Q_Sort array_name$,0,entries%-1
110 :
120 DEFine PROCedure Q_Sort (array$,bottom,top)
130   LOCAL sort_loop,low,high,ptr
140   low = bottom
150   high = top
160   ptr = bottom
170   REPEAT sort_loop
180     IF low >= high THEN EXIT sort_loop
190     IF array$(low) > array$(high) THEN
200       REMark need to swap these strings
210       temp$ = array$(low)
220       array$(low) = array$(high)
230       array$(high) = temp$
240       REMark how do we shuffle pointers to sections
250       IF ptr = low THEN
260         low = low + 1
270         ptr = high
280       ELSE
290         high = high - 1
300         ptr = low
310       END IF
320     ELSE
330       IF ptr = low THEN high = high - 1 : ELSE low = low + 1
340     END IF
350   END REPEAT sort_loop
360   IF ABS(top - bottom) < 2 THEN RETURN : REMark can't sort 1 item!
370   Q_Sort array$,bottom,ptr - 1
380   Q_Sort array$,ptr + 1,top
390 END DEFine Q_Sort

```

4. GRAPHICS FLOOD FILL

A graphics flood fill is the facility to fill an outline (regular or irregular) on the screen. Whatever the colour to be filled in, draw the outline (making sure there are no gaps where the colour filling might 'leak' out) then choose a point inside this outline to start filling from and the routine will gradually fill the shape. Imagine the outline of a house with windows and doors and roof already designed. Place a point somewhere on the wall and it will fill in the walls with colour, going around the windows and door as long as there is nowhere for it to leak through!

The basic principle is that whatever line we start filling from, we search left to find how far we can fill, search right to see how far we can fill, then fill that line and look above or below that line to see if we can go in that direction, if so we fill that line with our chosen colour and keep doing this until we can't go any further. We then retrace our steps back to a line where filling could continue from. With an irregular shape, what will usually happen is that it fills up as far as it will go, then fills down from where it started, then find another point from where filling can work from and so on. This sort of fill is often called a FLOOD fill because when you see it in action it looks like water flooding all over the place, filling in little bits as it goes.

It differs from the QL FILL command in that it can cope with so-called re-entrant shapes (irregular shapes which can bend back on themselves). The QL FILL command works in a different way to this sort of fill, and that means it will (usually) be faster than a flood fill routine, but the flood fill routine will cope with a wider variety of shapes to be filled, thus able to fill irregular (and quite complex) shapes, whereas you may need more than one QL FILL command to fill in a complex shape fully and correctly. The other major difference is that the flood fill works only AFTER the outline has been completely drawn, whereas the FILL command stores points and lines to fill as the outline itself is drawn, normally filling as the

x=x-coordinate to start filling from (0 to 511)
y=y-coordinate to start filling from (0 to 255)
direction=direction in which routine fills, -1=up, +1=down
colour=ink colour number of colour to use for filling

```
100 WINDOW 512,256,0,0 : CLS : INK 7
110 BLOCK 100,100,50,50,7 : BLOCK 96,98,52,51,0
120 BLOCK 60,80,70,60,7 : BLOCK 56,78,72,61,0
130 Fill_Outline 100,55,-1,7 : REMark -1 = upward, +1 = downward
140 STOP
150 :
160 DEFine PROCedure Fill_Outline (x,y,direction,colour)
170 LOCAL loop,a,lp%,rp%
```

outline is closed. To the user this makes little difference; the end result is usually similar, but to the programmer there is a great deal of difference in coding the two routines.

To make the flood fill work, we need a routine to examine the colour of a given pixel on the screen. Many toolkits of basic extensions provide one of these (e.g. the DIY Toolkit PIXEL function). In this routine I have used an extension called PIXEL% which requires the following parameters - change these to suit the extension you use. Unfortunately standard QL superbasic lacks such a function.

x = x co-ordinate across the screen
y = y co-ordinate down the screen
Both x and y assume that we are in a window covering the whole screen, i.e. 0,0 is at the top left corner of the screen.

The outline of the routine is not too difficult to understand, but the finished routines are horrendous to try to understand, especially as the more advanced of the two routines includes a buffering system to reduce the massive number of recursive calls done by the system, also helping to speed up the second routine compared to the first.

```
procedure fill_from start point x,y
  if not a fillable point then return
  fill to the left of this point
  fill to the right of this point
  from left to right
  fill_from points above this line
  fill_from points below this line
  end left to right scans
end of procedure definition
```

This can be written in the following form, using the extension PIXEL%(x,y) mentioned. The first 3 lines draw a box inside a box with a gap between them, which the routine then fills in. The second line in the procedure Fill_Outline (IF y<0 OR y>255) protects the program when the colour fill tries to go off the top or bottom of the screen. The routine is called by supplying 4 parameters to the Fill_Outline procedure name:-

```

180 IF y < 0 OR y > 255 THEN RETURN
190 IF PIXEL%(x,y) = colour THEN RETURN
200 lp% = x
210 REPEAT loop
220   IF lp% = 0 THEN EXIT loop
230   IF PIXEL%(lp%-1,y) = colour : EXIT loop
240   lp% = lp% - 1
250 END REPEAT loop
260 rp% = x
270 REPEAT loop
280   IF rp% = 511 THEN EXIT loop
290   IF PIXEL%(rp%+1,y) = colour : EXIT loop
300   rp% = rp% + 1
310 END REPEAT loop
320 BLOCK rp%-lp%+1,1,lp%,y,colour
330 FOR a = lp% TO rp%
340   Fill_Outline a,y+direction,direction,colour
350   Fill_Outline a,y-direction,-direction,colour
360 END FOR a
370 END DEFINE Fill_Outline

```

The problem with this routine is that it reaches a dead end, then has to retrace its steps back to fill in an opposite direction, which takes a long time for some shapes. We can improve this by adding a buffer which filters out some points in the opposite direction which cannot be filled, which in turn means it doesn't have to go back over as many pixels which it has already filled. There is still a delay after filling in the last point, but there is some time saving in using a stack or buffer like this. The fill points in the opposite direction are

updated as the routine goes in one direction, so the stack (actually an integer array) grows to remember fillable co-ordinates away from the original fill point, then once the routine comes to a halt in that direction, it starts to read pixel co-ordinates from the stack to remember where to go back to. It then works in the opposite direction, but should it again find points away from the direction of fill which it can later fill, it will also add those to the stack. And so the process continues until it has all finished.

```

100 WINDOW 512,256,0,0 : CLS : INK 7
110 BLOCK 100,100,50,50,7 : BLOCK 96,98,52,51,0
120 BLOCK 60,80,70,60,7 : BLOCK 56,78,72,61,0
130 Fill_From 100,55,-1,7 : REMark -1 = upward, +1 = downward
140 STOP
150 :
160 DEFINE PROCEDURE Fill_From (fx,fy,direc,col)
170   LOCAL xst%(512),yst%(512),stack%
180   REMark this routine requires a 'stack'
190   DIM xst%(512),yst%(512)
200   stack% = 0 : REMark nothing on stack at first, of course!
210   Fill_Outline fx,fy,direc,col
220 END DEFINE Fill_From
230 :
240 DEFINE PROCEDURE Fill_Outline (x,y,direction,colour)
250   LOCAL loop,a,lp%,rp%
260   IF y < 0 OR y > 255 THEN RETURN
270   IF PIXEL%(x,y) = colour THEN RETURN
280   lp% = x
290   REPEAT loop
300     IF lp% = 0 THEN EXIT loop
310     IF PIXEL%(lp%,y-direction) <> colour THEN xst%(stack%) = lp% :
yst%(stack%) = y-direction : stack% = stack%+1
320     IF PIXEL%(lp%-1,y) = colour : EXIT loop
330     lp% = lp% - 1

```

```

340 END REPEAT loop
350 rp% = x
360 REPEAT loop
370   IF rp% = 511 THEN EXIT loop
380   IF PIXEL%(rp%,y-direction) <> colour THEN xst%(stack%) = rp% :
yst%(stack%) = y-direction : stack% = stack%+1
390   IF PIXEL%(rp%+1,y) = colour : EXIT loop
400   rp% = rp% + 1
410 END REPEAT loop
420 BLOCK rp%-lp%+1,1,lp%,y,colour
430 FOR a = lp% TO rp%
440   Fill_Outline a,y+direction,direction,colour
450 END FOR a
460 REPEAT loop
470   IF stack% = 0 : EXIT loop
480   stack% = stack% - 1
490   Fill_Outline xst%(stack%),yst%(stack%),-direction,colour
500 END REPEAT loop
510 END DEFINE Fill_Outline

```

The above routine suffers from the drawback that very complex shapes can cause it to run out of stack space. In the vast majority of cases, 512 stack spaces will be adequate. These fill routines have the distinct advantage that they show on screen most of what is happening, so you can get a feel for how the routine thinks. After a while, even with complex shapes, you find you can predict how it will fill in the shapes!

5. THE TOWERS OF HANOI PUZZLE

Figure 1 is a diagram showing the appearance of the 3 pegs and rings of the Towers of Hanoi puzzle. What you have to do is to move all the rings from the first peg onto the second peg following certain rules. You can only move one ring at a time. No ring may be placed on a peg which contains a smaller ring (i.e. large rings can't sit on top of a smaller one). So you have to use the spare third peg as a temporary peg while you work

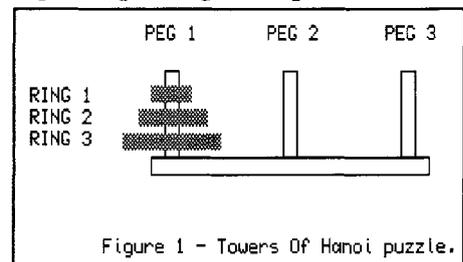
```

100 REMark Towers of Hanoi puzzle
110 INPUT'How many rings > ';rings
120 Hanoi rings,1,2,3
130 :
140 DEFINE PROCEDURE Hanoi(r,peg_a,peg_b,peg_c)
150 IF r = 0 THEN RETURN
160 Hanoi r-1,peg_a,peg_c,peg_b
170 PRINT'Move ring ';r;' from peg ';peg_a;' to peg ';peg_b
180 Hanoi r-1,peg_c,peg_b,peg_a
190 END DEFINE Hanoi

```

By now, you should have enough understanding of recursion to be able to reason out how and why this works by using a pile of coins to follow what is happening and at the same time following the flow of the program to understand what it is doing. Write out a generalised flow of what has to be done, see the patterns emerging and realise that although the problem is easy to solve in your head, writing a program to solve it is not quite that easy, especially writing a program this short to produce a solution! ■

out how to juggle the rings around until eventually the second peg contains the rings in the same order as they start on the first. This computerised recursive solution simply tells you which ring to move to which peg. You can, if you wish, add routines to it to animate the solution. If you are really ambitious, and have seen Tony Firshman and Laurence Reeves' little robot on the TF Services stand at QL shows controlled by a QL fitted with a Minerva and interface, you may like to consider how you'd set about programming the robot to do the solution to this puzzle! The program can cope with any number (within reason!) of rings, with ring 1 always being the smallest and the highest ring number is always the largest ring. For small numbers of pegs, you can follow the moves given with, say, a small pile of coins as long as you can remember which coin is which ring number! More than 3 or 4 rings and the instructions become quite long, because of the need to continually shuffle rings from peg to peg to avoid getting a larger ring on top of a smaller one.



A QL-Meeting in Norway

Albin Hessler

More than 25 years ago I had the chance to travel around Norway as my parents lived there for several years. So for several years I have longed to visit this beautiful country again. From Southern Germany this is a two day journey with the car. My family does not like such long travels by car, so I could not persuade them to go there until now.

Since several years I have been in contact with Arvid Borretzen, an old QL enthusiast who programmed with EASYPTR and often called me for special questions. We have met at QL-meetings in Munster and Eindhoven and there the idea was born to visit him in Norway.

This year we found a cheap charter flight to Norway at the end of May. So we decided to take this opportunity. We planned to hire a car to travel around through the fantastic Norwegian landscape and to visit Arvid. He invited us to stay at his home for some days, which we accepted thankfully. The flight went to Fargernes, a small airport (was it really an airport?), at least there was a runway on the Fjelle in the middle of Norway. From there we had to go about 300 kilometres to the west coast where Arvid lives in Leirvik on the island of Stord, about 50 kilometres south of Bergen. It was a really stunning journey. We went over the Hardanger Vidda, an impressive large high mountain area still covered with snow and ice, passed Vöringfossen, a huge waterfall and then we came down to the Hardanger Fjord, a part of which we followed for a rather long time. This part called Sörfjord is the Norwegian fruit-growing area where all the fruit-trees were just blooming as we visited.

The travel did take longer than we expected, so we only reached the ferry boat which took us over to the island, at eight o'clock in the evening. In the boat's saloon a guy came to me and introduced himself as Pal Monstad. Pal is a well-known QL-programmer and I had seen and phoned with him before. He had spoken to Arvid, so he knew that we should take this ferry. He is from the same Island, so he took the chance to take the same ferry on return from his teachers' college. Arvid, his wife Jorianne and the two sons were already waiting for us and welcomed us with a late dinner.

The next day, after a trip around the island, we inspected Arvid's office. He is the chief of the local food and health control authority, so *British Steak Enjoyment* has no chance in the "Yttre Sunnhordland". There are some QLs installed for

evaluations of chemical food analysis. Most PCs are equipped with QXL-cards and some Ataris with QL emulators are also used.

For the evening Arvid arranged a spontaneous QL-meeting in his office. Together with Ole Haukeland and Pal Monstad there were four of us. We had enough to talk about! I had the QPC beta software with me, and the others were interested in seeing how it works. So we made speed tests on several PCs and tested some of Arvid's software. They were quite impressed about QPC.

For me it was interesting to see a really professional environment where QDOS/SMSQ is still used and certainly will be used in the future, thanks to the QXL and the QPC emulator. As you can imagine, we ended very late and at Arvid's home we continued to bother his PC with *sympiotic* operating system questions until three o'clock in the morning.

Our travel back to Gol, where we wanted to stay for some more days, went over Bergen and the Sognefjord. The trip was very impressive again. Of 300 kilometres at least 50 km were tunnels, the longest measured 11 kilometres.

Meanwhile I sit here back home in Germany in an awful sultry heat longing for the clear Scandinavian weather! ■

Viruses and the QL

USA, Tim Swenson

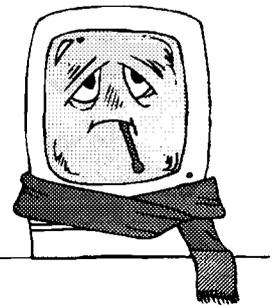
Now that more and more QLers are getting online, be it the Internet or BBS's, a few of them are starting to worry about what has plagued their "real" computer friends for years, viruses.

Many have heard of the horror stories of people losing hard disks and floppies to viruses. Ads for anti-virus products don't help.

The chance of getting a virus infecting your QL is very very low. Most of your data loss problems will come from either bad disks or the idiot behind the keyboard (hey, I have zapped a few of my own files). There are a few points to know about why the QL is very safe against viruses.

The QL is a Different Beast

Viruses are written for a specific operating system and will not work on other operating



Miracle Systems

QXL2

(QL hardware emulator for the PC)

25MHz 68EC040

8M bytes of RAM

ISA half card

SMSQ and SBASIC

Very fast

2 year warranty

*** Price (inc. VAT) ... £295.00**

(Outside EU £255)

or your GOLD CARD + £195

(Outside EU £165)

or SUPER GOLD CARD + £100

(Outside EU £85)

14 day money-back guarantee

Prices include VAT (except outside EU) and postage

Please make sterling cheques payable to "Miracle Systems Ltd"

quote your Visa/Mastercard number and expiry date

Tel and Fax: +44 (0) 1454 883602

Miracle Systems Ltd, 20 Mow Barton, Yate, BS17 5NF, UK

systems. An MS-DOS virus will not infect a Macintosh computer and vice versa. The virus code exploits certain operating system calls to infect disks. If the calls are not there, the virus goes nowhere. Plus, the MacOS does not know how to run an MS-DOS program, so the virus never gets executed.

The same goes for the QL. For a virus to affect a QL it has to be written specifically for the QL. Any file you download off a BBS or the Internet that does have a virus will be safe to use on a QL. The virus is in a totally alien environment and is just data to the QL.

QDOS is ROM Based

Some of the most deadly viruses are Boot sector viruses. These viruses infect the boot sector of hard drive or floppy disk. Any time you try to boot off an infected disk, you are loading the virus. Now the problem comes in finding a non-infected disk to boot off of so that you can start with a clean computer and get rid of the virus. If you choose the wrong disk, you will still be infected.

With the QL, its operating system in ROM, all you have to do is turn it on and not put a boot disk in. Now you know you are clean and can start to fix the problem.

Viruses Only Work When Executed

The key thing to remember with any computer platform, viruses must be executed for them to work. If you download a data file that has a virus in it, you are safe until you actually try to execute that data file. Viruses can not autoexecute. Now once a virus gets executed and gets in memory, it may be tough getting it out, but it had to have been executed at one point. So, if you don't know about a file or program that you've downloaded, run a virus scanner before you execute it.

No Known Viruses Written for the QL

Of course, one of the safest things to do to avoid viruses, is to have a computer so obscure that no virus writer would write a virus for it. It is this obscurity that helps keep the QL safe from viruses.

Before you worry too much about viruses when you download data to the QL, realize how safe you are and worry more about data loss from other means. Spend your time backing up files to other disks instead of worrying about viruses. ■

Harddisk and Backup

H.P. Huyg

1. Introduction

It's strange, when the QL had 128KB memory, one could work with it, then it got 896KB one needed these and one couldn't do anything on the basic machine. Then we got 2MB on the GoldCard and now 4MB on SuperGold and up to 8MB on the QXL. My 4MB QXL has, when everything has been set up, but before any program is executed, fewer than 2MB available and my SGC has around 1MB! With the external storage we get a new phenomenon: whilst the capacity of the Microdrive is in the same order as that of the QL, and the floppy capacity is similar to GoldCard/SGC, with the event of the hard disk the ratio can be 1 to 100 or over, my QXL accesses a 406MB hard disk. Given the reality that most systems have only one hard disk, it means there is a severe backup problem. Backup is not needed in an ideal environment and with perfect people operating in it. If you can claim both, do not read on. Although I suspect that sometimes "loss of data" is used as an excuse for not delivering the goods, the problem is a real one.

2. Why backup is needed

2.1. Hardware failure.

Although the reliability of microcomputer systems has improved significantly over the years, you can be sure that one day your system will fail and your data will be corrupted and/or become unavailable. You are NOT able to predict that day.

2.2. Theft.

Floppies and Microdrives can be kept at a secure place, computer and hard disks not. Possibly the probability of becoming a victim of theft is higher than of encountering a hardware failure.

2.3. Data corruption.

There are two kinds: unintended corruption and deliberately generated damage.

2.3.1. Unintended corruption.

Most, if not all software is written in languages like 'C' or Assembler, where protection against faults or errors is non-existent (in the latter case) or not sufficient (any stack-using application without adequate overrun protection is a potential time-bomb). The fact that programs, packages and the like seem to grow in time to gigantesque sizes makes predictions for the future not sunnier.

2.3.2. Intended damage.

As far as I am aware we live in the QL world in a clean environment. With QXL on PC's it is different. Whilst anti-virus packages 'protected' you against about 800 virae and vermins two years ago, today any package inoculating you against fewer than 3000 viruses is not worth acquiring. If you just work on your own, you have no problem. If you get software from anywhere, be prepared. If you download from BBS or Internet, watch out.

2.4. Self-protection.

By far the most common cause of "data loss" is the user itself. By just deleting that directory, or just performing some operations in the wrong sequence there has been much more data loss than that caused by all the other categories together. Humans are not machines and they cannot be programmed. Thank heavens. We need some backupmechanism though.

3. Hard disk organization

One should realize that what we use today, has been started some 15 years ago. This is so for the QL, and even more so for the PC. Hard disks were for mainframes. Floppies had a capacity of 360KB. What has come later on the market has grown from those times.

A hard disk has logically a directory structure and it has files in them. Less logically, the available storage is a collection of slots (clusters in PC-jargon), of 512 characters (bytes). So when you write a file of 150 characters, you'll use up a whole slot. Somewhere on the disk (floppy, hard) there are two tables (actually it is one table, but that is a detail), one containing all the available slots, the other all the used slots. That thing is known by experts as the File Allocation Table(FAT). Each time some space is needed, a number of slots are taken and assigned to the concerned file. Now, for historical reasons PC's always had problems going over 16 bits (even on the QL in Superbasic an integer is also maximum 15 bits, about 32000). This then would be the maximum number of slots, which implies a maximum disk size of 32 million. On the QL the same technique has been used, although there is no magic 16-bit limit (it is 31- or 32-bits, more than enough for some time to come). Please remember that 15 years ago nobody foresaw that a PC with Gigabyte disks could be sold like bread and butter. So there was a problem. Problems are there to be solved and the solution was to increase the size of a slot, which, on bigger disks is now commonly well over 7KB, giving a maximum 'capacity' of around 530MB. This means that

on a PC my 100 'mke_'-files, average length 200 bytes, would require over 700000 bytes, a wastage of more than 97%! PC's older than 1993 cannot cope with a bigger capacity. Current PC's have clusters of even 32KB. However, nothing stops you, by means of a utility program, to split your disk into two or more logical hard disks, on the PC called c:, d:, This is for QXL owners very interesting, since on each of those logical units one can have a QXL disk: win1_ on c:, win2_ on d:, etc. The slot-size on the QXL-disk is 2KB, just like on ED-floppies. The QXL disk has its own FAT.

Why, for heaven's sake would one want to have more than one winx_? There are many reasons for this, the first of them is called performance. This FAT must be somewhere and it must be manipulated somehow. This is not the same thing if it is for a few hundred slots as for tens of thousands of them. To illustrate this on the QXL (on the PC the problem is similar) the following test. My 406MB IDE disk with 11ms access time is split into three logical units: c: 150MB, d: 100MB and e: 150MB, and I have win1_ (20MB), win2_ (20MB) and win3_ (80MB), the last one for test purposes. The actually used capacity of win1_ was 9MB. With my magic bulk copy routine (on a file by file basis) I copy win1_ to win3_ex1_, win3_ex2_, ... , win3_ex8_. Thus I added each time 9MB, in one case I added in the middle of it all (that is why the results are not totally straightforward). The following table give the copy results and the time it took to just read all the directories on win3_ after the copy process.

TABLE: Filling a 80MB disk.

size MB	directories	files	copy 700 files (in seconds)	read all directories (in seconds)
9	27	700	328	2
18	55	1400	379	8
27	83	2100	420	17
36	111	2800	474	30
45	139	3500	512	44
54	167	4200	558	62
63	195	4900	632	76
72	223	5600	745	96

(I am very proud of my super duper copy program, because the speed of copying 9MB on the SAME disk is the same as the tape-streamer does, see 4.2.) If one considers the 2nd and the last line (the 1st might have benefitted from buffering & caching) as more or less representative, then it takes from 9 to 18MB half the time as from 63 to 72MB! Reading 55 directories with 1400 entries is pro rata three times faster than reading 225 directories with 5600 entries. In other words: small is beautiful. Validation timings (not in the table) increase as well, though slightly slower than

the copy times. The moral of this story is that it is MUCH faster to work with small (logical) hard disks than with one large one with the same total capacity. What is shown here for copy is valid for all work using hard disk extensively, it is valid for PC's as well, but with other ratios.

4. Backup possibilities

It all depends which cases (see 2.) one wants to cover. If all have to be coped with, there are two alternatives: removable disks and tape-streamers. With systems on a network and a server this applies to the server as well.

4.1. Removable disks.

One needs at least two units, one of which should allow removable disks, and a bunch of disks. The vital data should be on the removable disk and programs, workareas, etc on the fixed disk. The fixed disk should have an "over"-capacity to contain (temporarily) the contents of the essential userdata. This is the simplest and most elegant solution. Techniques/methodology are quite simple, one could consider these disks as extended floppies and if a backup is made regularly & systematically and a copy is locked away then most is well. An added benefit is that ALL essential data can be safely stored away when the system is not in use. Strangely enough, this solution has never caught on well, to my mind it shows that most systems are not professionally run. Possibly the size & price of them is a handicap (chicken and egg problem?)

4.2. Tapestreamers.

Basically a streamer is a unit which can read/write a special cassette. A few years ago they did cost thousands of dollars/pounds, but today one can buy units with a capacity of 400-1600MB for around \$150 - \$400. There are some products derived from DAT (Digital Audio Tape), DCC (Digital Cassette) and MiniDisc (a small writable CD), but they are all much more expensive & messy to use/connect. For some obscure reason the software which comes with a streamer makes it an embedded system. A streamer is not a unit, it is not visible to applications, cannot be programmed, it is thus a matter of take it or leave it, but if you take it then you must use it as prescribed. It cannot be used on a SGC, although normally they are on a standard floppy-'channel', because there is no software for it. On a PC it works on a unit, directory or file-level. Now, win1_ is for the PC just one file. Thus to make a backup, one makes a backup of the whole win1_, even if only one bit has changed. This is one reason to have small

win1_ ... etc, as the copy time increases with the size (not quite linearly, because compression is big for 'unused' parts of the file). Given the above, there appears immediately a second reason to have more than one winx_ on your hard disk. Suppose that in a drunken moment you have overwritten this essential file containing vital statistics and you cannot live further without them. Your last backup was two weeks ago and since then a series of other files have changed also. If you restore this with the streamer then you restore the whole lot, including the other files, result: you have gone from bad to worse. Tape streamer software is 'untouchable', but you could copy the lot back to say, win2_ with the streamer, and from there you just copy back your statistics (albeit 2 weeks old) back to win1_. There is yet another reason to have more than one winx_ If you consider your system then you can observe that there are (hopefully) lots of directories which are 'static' and a few which are 'dynamic', they change with the weather. Why making a copy of the whole lot if half of it 'never' changes? Example: the 'C'-package is over 4MB, but only the modules you write change, the rest not, when a new release of that package arrives, you should make a backup (before AND after installation!).

You won't believe it, but there are more arguments for having multiple winx_'s. It goes back to our FAT. It is a well known fact that the longer a hard disk is used (in the sense that files are written and modified), the allocation of slots becomes more of a random affair, this is called fragmentation, one doesn't see it, but your system becomes slower and slower. Once in a while it is useful to have everything put in a contiguous manner and the backup streamer software won't do that for you. The procedure would be: copy the lot (say win1_) to cassette; restore the lot to win2_; possibly make a full comparison between win1_ and win2_, the streamer software can do it for you; format win1_; copy the whole contents with a FILE-copy program back to win1_ et voila it feels if the system runs in an accelerated turbo-mode (as mentioned before, my magic copy program can do this very fast).

5. Practical considerations with tapestreamers

5.1. Does it fit on your PC?

Although there are external models, I would not recommend them, they are more expensive and there may very well be conflicts with connections, e.g. the printer port, but it can work. The best and normal way is somewhere in the PC, so you need

the space for an opening to the front. Installation is quite straightforward, but you must be prepared to remove and plug ribbon cables. If you are scared of it, ask your son, he will be quite happy to ruin your system, and he can do it. My main problem was that the unit must be fixed solidly on both sides, it is easy on one side, but the other is ALWAYS inaccessible, and there is a moment you have more parts outside the PC than inside. I have double labeled all my connections inside the PC! (it must be fixed on both sides, because cassettes stick half way out of the unit when in use and you need a considerable force to plug and unplug them, unplugging the unit rather than the cassette is embarrassing).

5.2. Which model?

The cheapest ones use the DC2120 cassettes, they store 120MB uncompressed (up to 200MB compressed). The blurb on the unit gives always a much higher capacity, but that is mostly with tapes you can't find anywhere, or which are far too expensive. Try to find a model which supports running under DOS as well (they all function under Windows), see 5.6. If you see a price, then double it for the cassettes, you need 7 .. 10 of them as a minimum. There is a trend to go to the TRAVAN series of cassette-units, they are more expensive, but they can cope with today's Gigabyte disks.

5.3. Cassettes.

Most are preformatted, but this is 'high tech'-nology and errors are not uncommon. You better format yourself, that takes about 2 hours for DC2120, and can take up to 5 hours for higher capacity tapes. Cleaning the units every 10-hours of usage is not a luxe but a necessity.

5.4. Partitioning of hard disks.

A must, as explained before, but do it AFTER you have installed your tapestreamer and you are SURE backup copies can be properly restored. I consider three logical units as a minimum, most QL software can cope with it, just like flp1_, flp2_,

5.5. Catalogue.

Always specify that the catalogue (i.e. the involved disk-directories) is written to tape as well. This saves time if you have to go to another system (with QXL!) to continue your work. It is handy too if you have to start from scratch again (see 5.6.) with your hard disk. Once I installed OS-2 and I got so many troubles that the system got clogged up completely, even on restart or startup, so that the only recovery was to format the hard disk from BIOS and then to restore from tape.

5.6. Restart from scratch.

If possible install the backup software both under DOS and under Windows, but have only ONE catalogue directory. I run the QXL from under DOS, but normally the backup operation is run from Windows. So I always start up under DOS. The reason to be able to do the backup program from DOS is the following: after having the whole system up and running, one makes a copy of the 'root' -directory (which should be almost empty), the DOS-directory and the backup-directory on a few floppies. Now, if anything happens, or you change your mind and you want to re-arrange the layout of your hard disk (that will make the disk contents inaccessible), then you can rebuild your system, after formatting, with just those floppies and your backup cassette. You don't have to go through the whole rigmarole of installing all those packages, including DOS & Windows, and your system is identical to the one when you took your backup. This is interesting if you buy a bigger disk, or if you have to replace your system for whatever reasons. Make sure you have a QXL-card available!

5.7. Backup timings.

There are many systems, I have to count on 5 - 6 minutes to do a backup for every 10MB effectively to be saved. All streamers use compression techniques, and the same cassette can contain several copies of a backup. I have two kinds of backup cassettes: one for the whole unit, more than 70MB, this takes well over 30 minutes, and I do it irregularly (only when I install new PC software), the other to copy just win1_, 20MB, of which 10 really used, which takes 5-6 minutes. This I do at least once a week!

6. Conclusions & Recommendations.

- There are only two proper backup systems for hard disks:
 - removable disk systems, and
 - tapestreamers.

Partitioning of hard disks is essential, especially with higher capacity disks.

- See if you can separate programs from data.
- Have a fixed schedule of backup and adhere to it! I wished I could do it!
- Parts of back-up data should be kept at a different place than where the computer is (fire, theft)
- Keep multiple 'generations' of backup.



The Menu Extension (QMENU) - Part 1

Bangor, Wales - Dilwyn Jones

A few years ago during my first period of attempting to write a program using the pointer environment, any comment from me on the subject may have been unprintable. In those days you could hack away in machine code or use the dreaded QPTR Toolkit. No disrespect to the author, it's not that there was much wrong with it, but it wasn't the easiest package for a novice programmer to learn to use.

Sometime later I bought a copy of QMenu from Jochen Merz Software and while it didn't exactly change my life, it did make it a bit easier to write pointer driven programs from SuperBasic. Combined with the excellent QLiberator basic compiler, I was soon writing a great deal of programs using The Menu Extension. If you want to see what can be achieved with this software, check out programs such as QD from Jochen Merz, or Sidewriter or Deskjet-A5 from Quo Vadis Design.

The 'review' part of this article will be fairly short, before going on to list facilities and give examples of use, concluding with a complete file handling program which shows just how much can be achieved in quite a little code with QMenu.

QMENU Description

QMenu is a system based on code Jochen Merz used when writing his QD editor. After a while, Jochen packaged it up as a separate programming tool in its own right. To quote from the first page of the manual, "This software package offers the user a truly simple to use interface, which instead of continually confronting one with overlays and menus, offers simple to use, practical menus.... Whether it's for a short program for your own use, or for a commercially viable project, this is the one for you."

That quote sums it up. If you have a reasonable knowledge of SuperBASIC, and some familiarity with the PE (pointer environment) such as experience of using QPAC2, you can quickly master QMenu. There is not much by way of examples in the manual, although the relevant information is all there, but the program disk contains several short example basic and machine code programs.

The package includes a set of pre-defined menu

formats for a variety of commonly needed facilities. The menu designs are broadly fixed and the user cannot change much, other than colours and a limited control over menu sizes. By supplying a list of parameters, you control how each menu responds and to some extent what it does - you can supply values or text or lists to the menus, drive details and so on. Until you actually get used to the system, it is difficult to appreciate exactly what a package like this is capable of doing. Figures 1 to 11 will show what the menus look like. In addition to menus, the package includes facilities like the Scrap (a sort of clipboard for cutting and pasting data between packages which know how to handle data from the Scrap.)

QMenu is an excellent product within the limits of what it sets out to achieve. It doesn't have (or purport to have) all the facilities of a full pointer programming tool such as the QPTR Toolkit or Easyptr, but it is a lot easier to learn to use. As long as you have some experience of writing superbasic or machine code programs, and have some experience of pointer driven programs and their associated terminology, you will soon be writing programs using QMenu. You probably won't be producing that ultimate graphics program with it for example, but for general use it is likely to meet all needs. Its outstanding merit is that it is possible to produce pointer driven applications of moderate complexity very quickly.

The manual is probably the weakest part in my opinion. All the essential information is there (except how to use OUTLN, see below), but there is little by way of printed examples, so it is a bit daunting when you first see the instructions, I wish that some short example routines could have been included in the manual, and that Jochen Merz could have seen fit to include more information to get the first-time pointer programmer going. To be fair, this package is not that complex and it is not aimed at those with little or no experience of Pointer Environment. If you know how to handle procedures, functions, arrays and so on, you should be able to master the basic interface at least. It can be used from superbasic (or sbasic) without knowledge of machine code.

Would I buy it? Yes, I did some time ago. And I have used it in many programs written for my own use and a few programs commercially released through Quo Vadis Design. Which reminds me to mention that when you give away or sell your programs which include the Menu Extension, there is a royalty to pay. Actually, two sets of royalties - you need to pay Tony Tebby for use of the basic three files PTR_GEN, WMAN and

HOT_REXT (the equivalent of one Pound Sterling per set supplied) and pay Jochen Merz 3 German Marks for the licence fee to supply a copy of the MENU_REXT file from QMenu. This is not a large amount to pay for software of this calibre, and the pricing of programs which are sold using this software takes the royalty payment into account.

The main body of code is held in a file called MENU_REXT. This contains both the assembler and superbasic interface. From assembler, menus are called by using the Menu Thing. A Thing is basically an area of memory with a name, which a program uses by looking through a list (the "Thing list") of similar items to find the one with the name expected. Exactly what any particular Thing does depends on the application, you cannot tightly define what Things are (other than named areas of memory) because they can be written to do just about anything, hence the name, and applications which use them have to know how to use the routines or data in a given Thing. Don't worry too much about Things, as the QPAC2 manual says, "Things rarely go bump in the night," you can use QMenu perfectly well from superbasic without knowing about Things. Each of the assembler routines is clearly documented with all the information you are likely to need. I won't go into too much detail about the machine code interface here, as the average user is probably more likely to use the superbasic extensions, and the example basic routines I'll present here show what the menus look like and what they can do.

There are menus for selecting items from lists (List Select and the simpler but less flexible Item Select), menus for file handling (File Select, Dir Select, Ext Select and View File), ones for error reporting (Report Error and File Error), and string entry (Read String). There are also a few useful little functions which don't fit into these categories, such as button handling, character selection and default configuration reading (GET_DEFAULT\$ function). All these can be used from both basic and machine code.

There is, however, one menu which can only be used from machine code, due to what it does. The Do And Report menu is used to report the progress of an action (an operation which takes a long time such as formatting a disk) or to report a final result. It is a shame that something like this could not be implemented in the basic interface - the facility to leave a menu drawn on screen while something else is going on (even if only a "Please Wait..." type message) would be useful.

Another facility not currently implemented which would be very useful is the facility to do a

direct read of the pointer position on the screen. This would be useful for graphical applications such as drawing lines, for example.

A 'menu', for those unfamiliar with the term, is a list of items, from which you can select one or more items. For example, a program might ask a user to respond to a question with Yes, No or press ESCAPE.

Using The MENU Extension

To use the Menu Extension from SuperBASIC, you need to load the standard pointer environment files PTR_GEN (the Pointer Interface), WMAN (the Window Manager which controls what pointer environment menus etc look like) and HOT_REXT (the Hotkey System II). You may already have these files if you have purchased other pointer driven packages, as they are usually supplied with software which require them. Jochen Merz also supplies them with QMenu.

The simplest of BOOT programs will load these three files and the MENU_REXT file, and start the hotkey job with a HOT_GO command. Another possibility is to use the OUTLN_REXT file. This consists of an extension called OUTLN, which defines the outline, or area of the screen to be used by the program. This is not strictly needed in superbasic as the outline tends to be set automatically to cover the area used by the basic windows. But you may need it if you wish to compile the basic program for example. This example boot program uses the TK2_EXT and LRESPR commands from Toolkit 2

```
100 TK2_EXT : REMark ensures Toolkit 2 on
110 LRESPR FLP1_PTR_GEN
120 LRESPR FLP1_WMAN
130 LRESPR FLP1_HOT_REXT
140 LRESPR FLP1_MENU_REXT
150 LRESPR FLP1_OUTLN_REXT : REMark may not
    be needed
160 HOT_GO : REMark activate hotkey job
```

SMSQ users, for example, have an OUTLN command built in (latest versions), and the QPTR and Easyptr basic toolkits also have an outline command, so OUTLN_REXT may not be needed. A little hint for QLiberator users (not in the QMenu manual) - OUTLN_REXT can be linked to a compiled basic program if required with a simple \$\$ASMB statement, this version being suitable for the copy I have but may vary from version to version if Jochen ever decides to change it.

```
REMark $$asmb=FLP1_OUTLN_REXT,0,54
```

will ensure that the whole standard QL screen display is stored when you switch between programs. This is called 'setting the outline' for the program. Without this, you may find that your programs appear not to work - the menu may be drawn, a pointer appears, but you can't control anything or do anything. If this happens to you in basic, all is not lost - **BREAK (CTRL SPACE)** will remove the menu, giving the usual 'Not Complete' error report. Incidentally, the **OUTLN** command does not need to cover the whole screen unless the menu concerned needs to use that size section of the screen. What I'm getting at is that if you are using **QMenu** on a large screen display such as the **VGA** or **SVGA** displays on a **QXL**, the **512x256** definition may still be valid unless you deliberately set the height of a menu to take advantage of the larger display area.

Two terms we need to be familiar with are **HIT** and **DO**. **HIT** means you press the **SPACE** bar on the keyboard, or press the left hand mouse button. **DO** means press the **ENTER** key on the keyboard, or press the right hand key on the mouse. This suggests that a mouse or the keyboard can be used to control the menus and this is indeed true.

OK, now we've got the basics out of the way, let us start to investigate the available menus.

The Menus And Their Use From BASIC Programs

There are several features common to all menus, such as the method of specifying origins (top left corners of menus) and colourways (colour schemes). The colours are usually specified as the last of a list of parameters supplied to the basic extensions which call up the menus. There are 4 main colour schemes:

- 0=white and green
- 1=black and red
- 2=white and red
- 3=black and green

In addition, **-1** can be used to tell the extension to use the 'default' colour scheme programmed into **MENU_REXT** (**MENU_REXT** itself can be configured with the usual **CONFIG** program used to configure most pointer driven programs). This is quite an useful option, as it ensures that all your programs which use the default configured colour will have the same appearance. Consistency of appearance of course is often cited as a major advantage of using the **Pointer Environment** - the **Window Manager** helps to make this possible. Some menus allow a

main colour (the colour scheme of the menu frame itself) and a list colour (the colour in which the list of selectable items is drawn).

You can omit colour values. Simply put nothing between the commas or between the last comma and the closing bracket where the colour value should be and the program will use the default configured value.

Before this usually comes the 'menu position'. This is where the top left corner of the menu is drawn, measured in pixel co-ordinates from the top left. An **x** (across) and **y** (down) value may be specified. If you omit these values, the menus will be drawn at the current pointer position (if possible, otherwise the menus will be moved slightly to fit them on screen).

Defaults exist for many of the optional parameters to the basic extensions (shown as parameters in square brackets in the documentation). For simple programs, where the defaults may be adequate, taking advantage of them can simplify programs. For example, instead of spending a lot of time experimenting with menu positions, you can simply omit the position values in most cases and the menus will just pop up at the current pointer position. This might mean that the menus never appear in the same place twice, but in many applications, especially ones for your own use, this may not matter.

Probably the simplest menu to start with is **REPORT_ERROR**. You can use this to report an error and allow the user to acknowledge this before the program does anything else. For example, suppose you want to report the error 'Not Complete'. The code for this is **-1**. Here is a short listing showing how to use the **REPORT_ERROR** menu and Figure 1 is a screen dump showing what it actually looks like.

```
10 REPORT_ERROR -1,100,50,0
```



The menu reports the error 'not complete' (or 'incomplete' if using **SMSQ**), and is drawn 100 pixels across and 50 pixels down, using the

white and green colour scheme.

The user can **HIT** or **DO** on the 'OK' item to continue. Most menus allow you to press the **ESC** key as well.

Here we see how a simple one line basic program is able to do all this; draw a menu, report an error and wait for the user to take an action before the program continues. It gives a clue to the tremendous programming power of a package like **QMenu**. And it gets better!

REPORT_ERROR is implemented as a procedure. Some of the menu extensions are implemented as basic functions, which means they return values. For example, there is another error reporting menu (FILE_ERROR) which in addition to the actions taken by REPORT_ERROR, returns a value used to indicate what course of action the user chose in response to the error reported. For example, the user may be asked if he/she wishes to 'retry' (e.g. put a disk in the drive and try again) or 'abort' (if at first you don't succeed, GIVE UP!). Some errors allow you to specify EDIT or OVERWRITE (e.g. when a file of that name already exists). You can enable or disable the extra options according to the second parameter (the parameters are the list of values in brackets, in this case the '1' means that the extra options are to be shown, use a '0' if not). The value returned enables your program to take different actions according to the user's response. Here, I have used a SELECT routine to print different messages.

```

10 fer = FILE_ERROR(-8,1,100,50,0)
20 SElect ON fer
30  =-1 : PRINT'Not a filing error!'
40  =0  : PRINT'Abort (or ESC pressed).'
50  =1  : PRINT'Retry specified.'
60  =2  : PRINT'Overwrite specified.'
70  =3  : PRINT'Edit specified.'
80 END SElect

```

Figure 2 shows what the menu itself looks like. I used the error -8 as the first parameter; change this value to experiment a little to see how the menu varies with the error code. You can find lists of error messages



Figure 2 - FILE_ERROR menu.

corresponding to given codes in the QL manual, or by using the REPORT_ERROR menu to examine them one by one. The example listing given does nothing in practice except confirming what each value returned is meant to signify. Notice that the program cannot tell the difference between the user pressing the ESC key or selecting the 'Abort' option. In practice, both normally mean the same thing in a program, so this is not a serious shortcoming.

The READ_STRING menu function is essentially a pointer version of an INPUT statement in SuperBASIC. You can supply a prompt (a message to the user) and specify a restriction on the length of string which can be typed in. In addition, the cursor may be placed at the end or at the beginning of the string supplied for editing, in which case the first letter or number key you type clears the string and you start typing something anew, unless you

use a cursor left or right key press when you can modify the string shown. Figure 3 shows the menu displayed by the short program shown below. All it does is pretend to be a menu where you are asked to specify a filename for a file to be saved and then

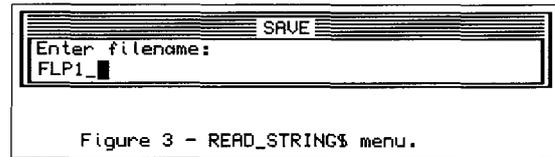


Figure 3 - READ_STRING menu.

prints the name you specified to the screen (you will notice that once you press ENTER to complete the entry, the menu disappears; the menus in QMenu all work like this, they go away of their own accord when finished. You cannot have menus which overlap from superbasic when using QMenu, other than certain viewing or message windows which can pop up from within other menus as the result of errors, or the 'cup of tea' Please Wait menu which appears temporarily during the Tree facility in the FILE_SELECT menu - this is one restriction in the package for those wishing to write advanced programs, but QMenu is not meant to be that complex anyhow. Many users may also claim that multiple overlapping windows are at best untidy, at worst confusing.

```

10 CLS
20 LET suggest$ = 'FLP1_'
30 LET t$ = READ_STRING$('SAVE', suggest$,
'Enter filename : ', 65536+41 ,,,0)
40 PRINT'You entered ';t$

```

The first parameter is a heading for the menu. The variable 'suggest\$' holds a default drive name to save the user some typing. The prompt parameter tells the user what to do, while the 41 specifies that the entry cannot be longer than 41 characters (5 for the drive name and 36 for the filename). Adding 65536 to this value forces the cursor to appear at the end of the suggestion presented for editing. Typical uses for this menu includes entering filenames, data required by a program, editing existing strings, anything where you'd use an INPUT statement, but with the added advantage of being able to restrict the length of the string entered, edit existing strings and control of initial cursor position. Numeric values can only be entered as strings, but it would be easy to transfer numeric values to a numeric variable later after checking it was really a number entered!

There are two other categories of menus - those that deal with files and those which allow you to select something from a list. Let us now look at the latter, starting with the simplest to understand and

S.J.P.D. SOFTWare.

WWW Page <http://www.di-ren.co.uk/sjpd/homepage>
CompuServe ID = 101325,2750
Internet Email stephen.johnson@almac.co.uk

36, Eldwick Street,
Burnley,
Lancashire,
England.
BB10 3DZ.



Tel / Fax +44 (0)1282 701767.

QL HACKERS JOURNAL.
INTERNET GUIDE TEXT FILES.
UNGIF Release 094.
7 DISKS OF RELIGIOUS CLIP-ART.
XCHANGE Version 3.90L.
ZEXCEL. SPECTRUM EMULATOR.
INFOCOM ZIP COMPILER.
GHOSTSCRIPT 2.6.1.
QL INFOZIP RELEASE 004.
MOLECULAR GRAPHICS V5.13.
PRETTY GOOD PRIVACY.
C68 Release 4.21b.
"C" TUTOR (SJPD Version.)
PERSISTENCE OF VISION.
M. EDWARDS DEMO DISK.
G. WICKS DEMO DISK.
QL WAR Version 3.02.
Z80 CROSS MODULE ASSEMBLER.
POINTER MICROEMACS.
SALVATION ARMY CLIPART.
ST CLIP-ART.
LINEDESIGN FONTS.
MONOP. (MONOPOLY)
TREK.(STAR TREK)
LIB LIST.

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

Please see catalogue or contact me prior to ordering as
some packages come on more than 1 disk.

*Plus 64 disks of the best
assorted in QL Public
Domain & Shareware.*

*A full range of emulators for
use on either IBM
compatibles or the Amiga.*

*Over 200 disks of classic
literature & assorted text files.
Including the complete works
of Shakespeare.*

**Gold Card &
Super Gold Card Users!**

**IBM Branded
Extra High Density Disks
Now In Stock**

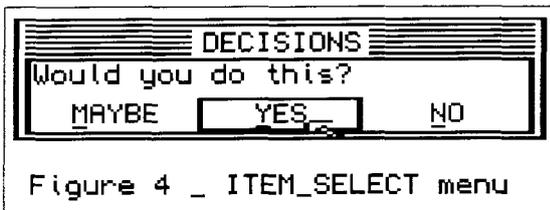
**Only £2.50 Each Or £22.00 For 10.
Fully Inclusive Of UK P & P.**

QL Pre-Copied Disks Are £1.25. Each, Inclusive Of Media And Return Postage.
For A Catalogue Disk Which Details Over 500 Disks Of PD & Shareware & Text
Files Offered. Send A Formatted Disk And Return Postage.

"From The Nurse Helping To Keep The QL Alive."

10/06/96.

use, ITEM_SELECT. This function gives you a simple horizontal menu allowing you to choose



one of up to 3 options (plus the ESC key). Figure 4 shows what it looks like. A simple program allows us to use it to make a simple decision maker, like this. Whatever the menu asks, you can respond with YES, NO or a definite MAYBE!

```
100 CLS
110 decision =
ITEM_SELECT('DECISIONS','Would you do
this?', 'YES', 'NO', 'MAYBE',,,)
120 SElect ON decision
130 =0 : PRINT'ESC pressed.'
140 =1 : PRINT'YES, you would.'
150 =2 : PRINT'NO, you would not.'
160 =3 : PRINT'MAYBE you would!'
170 END SElect
```

By now, you may have noticed that some options have a letter underlined. By pressing this key, you can choose an item via the keyboard without having to move the pointer over it. Also, as the pointer moves over an object, it is highlighted - the menus put a box around the item.

The ITEM_SELECT menu is also useful for presenting YES/NO type menus (e.g. a confirmation request before deleting a file or quitting from a program). Incidentally, you do not need to supply all three parameters. A YES/NO type menu only requires two items, while a single item is useful in the case of a "Press Any Key To Continue" type menu, for example:

```
100 temp = ITEM_SELECT('PAUSE','Hit OK to
continue.','OK',,,,,)
```

Note how many parameters are not specified in the above menu (shown by the long list of commas).

Continued in the next issue.

QL Service List - Part 2

Here comes the next part of the QL Service list. If we have missed somebody in the first part and he/she feels that he/she should have been in the first part then please let us know and we will publish up-to-date information in the next issue.

Software dealers

Albin Hessler Software

Im Zeilfeld 25
D-72631 Aichtal
phone: +49 - 7127 / 952 890
fax: +49 - 7127 / 952 898
- Cueshell, Easyprtr, Disa (dist. by J. Merz)

Athene Consults

0705 / 511 439

CGH Services (Richard Alexander)

01559 / 384 574

(Closed 31 March 1993 - enquiries about past products only.)

COWO Electronic (Switzerland)

c/o Urs König
Münsterstrasse 4
CH-6210 Sursee
010 - 41 - 45 / 211 478
QTop, Thor support.

Deltasoft

7 Tyrell Way,
Stoke Gifford,
Bristol.
- FlightDeck, Image D, AMD Airplan.

Digital Precision

222 The Avenue
London, E4 9SE
(United Kingdom)
phone +44 - 181 / 527 5493 (?)
- Perfection, PC Conqueror, Lightning,
Professional Publisher, Eye-Q, Solution, Media
Manager, Professional Astrologer, QMaths,
CPort, and others. They currently sell the "QL
Collection" which is a collection of all their
programs for a price of £179.

DJW Software

phone 01256 / 881701
- homebanker

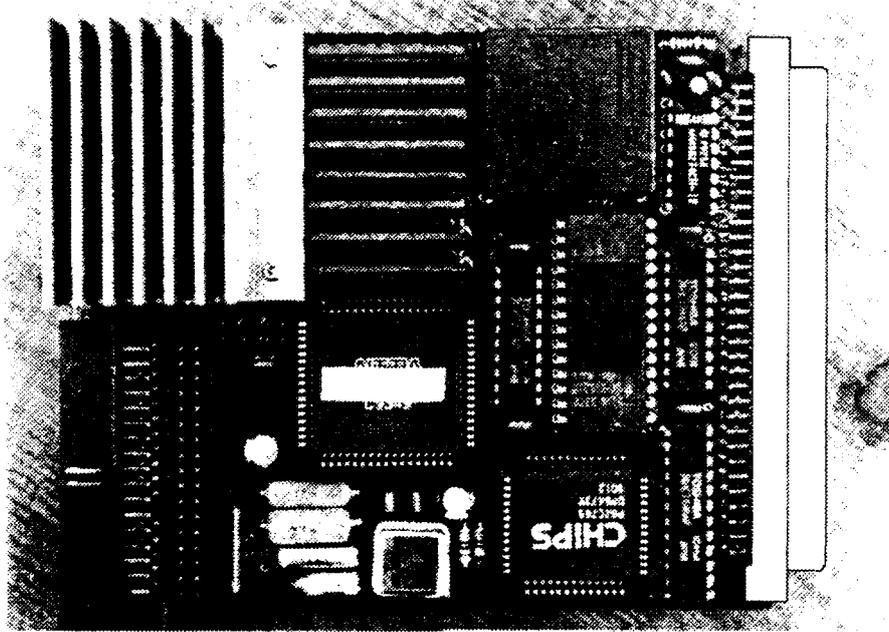
Dilwyn Jones Computing (DJC)

41 Bro Emrys
Tal-y-Bont
Bangor
Gwynedd, LL57 3YT
(Great Briatin)
phone/fax: +44 - 1248 / 354 023
- lots of software. (closed in March 95)

QUANTA

SUPER GOLD CARD

"The Pathway to Future QL Development"



- * *3 Times Faster*
- * *68020 processor*
- * *4M bytes of RAM*
- * *2 Disk drive ports*
- * *CENTRONICS port*
With FREE printer cable
- * *£275.00 + £5 postage*

*To Order please send a cheque
drawn on a U.K. Bank for
£280.00 payable 'QUANTA' to:*

J Taylor
145, Barrowby Rd.
Grantham
Lincs.
NG31 8AJ

*(for more information
phone 01476 563404)*

Di-Ren (Robin Barker)

Jelanda, Wyndley Drive
Sutton Coldfield B73 6EU
(Great Britain)

phone/fax: +44 - 121 / 355 3943

email: support@Di-Ren demon.co.uk

www: <http://www.forthrt.com/~di-ren/homepage.html>

- Fleet Tactical Command, keyboard interface Amadeus token ring network system. Di-Ren also maintains an extensive amount of QL pages on their world wide web site. They're covering virtually every QL related internet address. Check it out.

Ergon Developments (Davide Santachiara)

Via Emilio de Marchi 2

42100 Reggio Emilia

Italy

phone/fax: +39 - 522 / 70409

FIDOnet 2:335/21.11

- ZM-X ZX Spectrum emulator, Open World, other QL software.

Grange Technology Limited

Rosebank, Stream Road

Upton, Oxon, OX11 9JG

United Kingdom

Tel/Fax: +44 - 235 / 851818

- GT-Prolog/QL

Joe Haftke

7 Lansdown Road

Sidcup, Kent DA14 4EF

United Kingdom Tel.: +44 - 181 / 302 6154 -

Printmaster plus; Filemaster plus; Vision mixer plus; Home bills budget; Picturemaster plus; Remind me plus; JDir. (All at £15 each; Upgrades £7 GBP + old Master disk)

Jochen Merz Software

Im Stillen Winkel 12

D-47169 Duisburg

(Germany)

phone: +49 - 203 / 50 20 11

fax: +49 - 203 / 50 20 12

mailbox: +49 - 203 / 50 20 13 (ZyX 16800)

mailbox: +49 - 203 / 50 20 14 (28800)

(orders from 10am to 8pm local time)

- QPC, QL/Atari emulators, QD, QSpread, File Finder, QPTR Pointer Environment Toolkit, System documentation, other PE programs, many games, and others. He also sells SMSQ/E (QDOS compatible operating system) as well as often needed hardware components, e.g. ZX8301, ZX8302, keyboard membranes etc. Accepts Eurocheques, Eurocard, Visa, Access, Mastercard, American Express and Diners Club.

Lear Data Systems

6 Southview Green,

Bentley,

Ipswich,

Suffolk IP9 2DR

- PCB-CAD.

Liberated Software

0181 / 546 7795

- QLib Basic compiler and utilities

Ocean Computer Services

0161 / 740 9002

- Professional Poolster

Pointer Products

01258 / 455 117

- Pointer Environment programs.

PM data

Tel./Fax (+47) 52 78 34 13

Søndenålia 27

4200 Sauda

Norway

- Disk Mate 5

Progs (Van Auwera)

Haachtstraat 92

3020 Veltem

Belgium

phone/fax: +32 - 16 / 488 952

- LineDesign, The Painter, The Clipart, DataDesign, Proforma, ProWesS PROGS Window Manager.

Q Branch

P.O. Box 7

Portslade

East Sussex BN41 2ND

(Great Britain)

phone: +44 - 1273 / 386 030

fax: +44 - 1273 / 381 577

- Lots of Pointer Environment software, Spy, Master Spy and others.

Qubbesoft PD

38 Brunwin Road,

Rayne,

Braintree

Essex CM7 5BU

(Great Britain)

phone: +44 - 1376 / 347 852

fax: +44 - 1376 / 331 267

email: 101634.776@compuserve.com

- QL Home Finance, Public Domain Software, see also hardware section.

Quo Vadis Design
57 Shaftesbury Road
Romford,
Essex, RM1 2QJ
United Kingdom
Tel.: +44 - 1708 / 755 759

email: 101366.3534@compuserve.com
www: <http://ourworld.compuserve.com/homepages/qcd>
- Lots of QL Software including the full range from Albin Hessler and Ergon Development.

SD Microsystems

01462 / 422 897
- General Ledger, Small Traders' Pack/Invoicer and Stock Accounting. Other business software.

SJPD Software

36 Eldwick St
Burnley Lancashire, BB10 3DZ
(United Kingdom) phone: +44 - 1282 701 767
- Public Domain Software. Send formatted QL disk plus return postage for a free catalogue.

Software 87

33 Savernake Road
London, NW3 2JU
(Great Britain)
- Text87 Plus4, Publishers' Pack.

TF Services

Holly Corner
Priory Road,
ASCOT,
Berks SL5 8RL
United Kingdom
Phone 01344 - 890986
Fax & BBS 01344-890987
- Qualsoft QL Terminal Emulator, File Transfer.

Geoffrey Wicks (Just Words!)

Bertrand Russellstraat 22
1097 HL Amsterdam
The Netherlands
phone: (31) (0)20 692 - 1521
Netherlands Bank: Postbank number 4111 942
- Solvit-plus 2 electronic dictionary, english thesaurus, style checker.

Wood and Wind Computing (Bill Cable)

RR3 Box 92
Cornish NH 03745
USA
phone: (US) (603) 675 - 2218

WS Software

0534 / 81392
■

My QL rebuild

Mark Knight

I was recently given one of those PC minitower cases fitted with a switched mode power supply (thanks John!) and I decided it was a perfect opportunity to rebuild my QL in a more tidy form. It would have suited me better to have a black case and keyboard but anything free is welcome when you are a poverty stricken student and this was before I was forced to quit college. Anyway I have successfully rebuilt my QL and I am happy with the result. I decided to write this article to pass on my experience and hope that somebody out there can make use of what I have learned as I built this system.

I had for some time a QL with a Gold Card sticking out of it, a separate box containing a 3.5 inch disk drive and naturally the printer and monitor took up some room too. There was an external power supply for the QL, one in the disk drive unit and leads and plugs for the monitor and printers (I have two printers). This was real computer spaghetti. I long ago became fed up with the QL keyboard and bought a Schon PC-style keyboard, so my QL looked odd as well as scruffy. I know some find the original QL keyboard acceptable, but I was working with MS-DOS computers professionally at the time and coming home to the QL keyboard after a day using IBM and good clone keyboards was a nightmare. The QL of course I prefer, but sorry that keyboard is a joke.

I had a great advantage as I had recently been given a non-working QL (thanks Basil!) with some of its chips missing: this gave me the opportunity to work on mounting the QL circuit board and assorted bits without sacrificing my current system until the new box was ready. I ordered QUBBE SOFT's Q-Plane to allow me to power the QL from the PC power supply and then started thinking carefully. Quick service too from QUBBE SOFT, Q-Plane arrived very quickly and I didn't know the post could be that quick (thanks are due to Ron Dunnet for a speedy despatch).

The first thought I had was that I wanted the QL motherboard along the side of the case not along the bottom as most others seem to have it. This leaves the circuit board vertically oriented and improves the airflow, keeping the components cool. Because the three expansion ports on Q-Plane also lay vertically in this arrangement it leaves my Gold Card and any other expansions that I purchase later oriented the same way. The

air in this case (and most other tower cases) flows in through ventilation slots at the front underside of the fascia and is blown out of the upper rear of the case through the power supply's fan. In the intended use of this case the PC board is likely to generate a lot of heat so this is the way the airflow is supposed to work. Top marks for me there, still, as my college tutors would tell you this is nothing new...

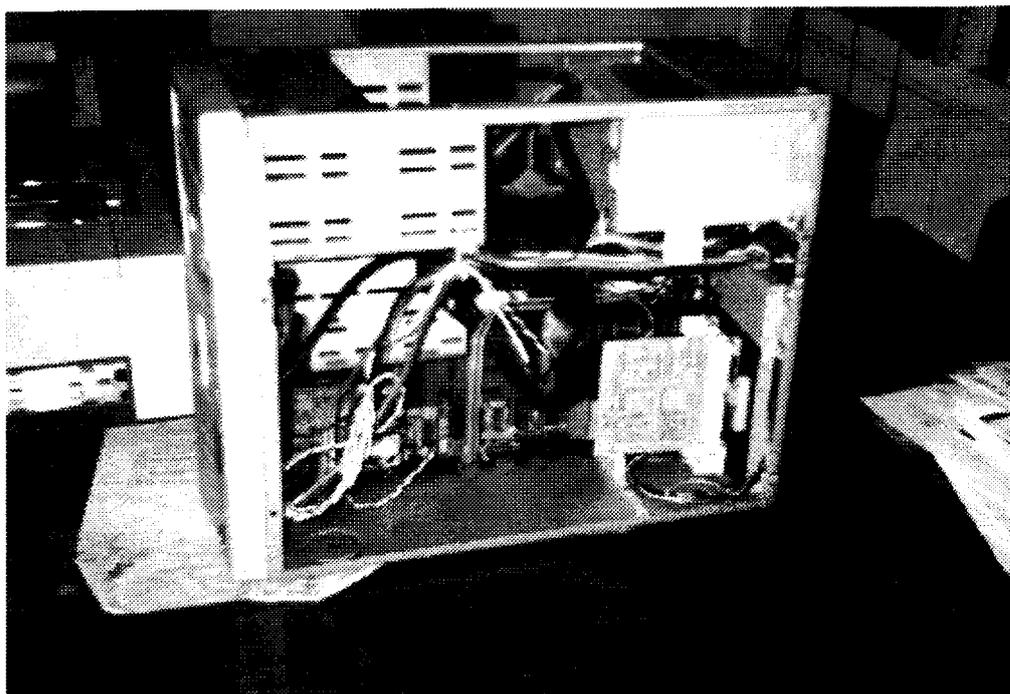
Next came the important considerations of strength of mounting and electrical insulation. I used the bottom half of the QL case as the mounting for the motherboard, cut up with a hacksaw to remove the microdrive portion and other parts not required. This means that pressing in socketed chips and keyboard interfaces is fine as the QL board is held in place by the usual two screws and backed and supported by the same sheet of plastic as when it is in the original case. The remainder of the tough plastic case was kept to cut up for mounting brackets. The QL board is surrounded by fairly thick black plastic in this arrangement and so is well protected from electrical shorts.

The QL case is a particularly good grade of ABS plastic, unusually for Sinclair no corners were cut here. ABS (Acrylonitrile Butedene Styrene, for the curious) has good electrical insulating properties and is very impact resistant. It is also very tough and it will bind blunt saw blades and drill bits quite quickly. I would recommend to anybody using it in the way that I have done to start with new drill bits and saw blades, they are cheap and you will avoid the next experience I had.

Drilling a hole in one of the newly cut plastic brackets with a rather old drill bit I was cautious but not cautious enough. The bit bound in the plastic, the strong motor of the drill dug in and something had to give. The drill bit shattered under the strain and exploded all over my hallway, cutting my right hand slightly and leaving a hole in the wall close to the vice I was using. No real harm was done but it certainly could have been and the

accident would probably not have happened if I had been using a sharper bit. Upon examination my hand was barely scratched but that bit of metal could have gone in my eye...

Having absorbed this lesson and stopped for a day or three to get on with other things (college essays mainly) I came back to the project with renewed energy. I pushed out three of the four punch out panels on the back of the case and fitted connectors for the serial ports and probable future parallel port. I don't have a parallel port but I do plan to buy a Super Gold Card at some future date and I will wire this connector up to take a PC parallel printer lead when this happens.



The connectors for the serial ports are industry standard, a 9-pin D-Sub female connector for ser1 and a male version of the same for ser2. This prevents me plugging the Miracle Centronics interface into ser2, as it now has a new 9-pin D-Sub male plug on it to match the new ser1. My Miracle Centronics interface doesn't work properly in ser2 since I got a Gold Card, though it works perfectly in ser1, so I did this with an eye to setting up in a hurry after moving my system for QL shows etc. The industry standard 25-pin female D-Sub socket was fitted in place for the parallel port but is not wired to anything at present.

The reason for choosing the male plug for ser2 is that this matches a PC standard serial mouse, which has the female connector in the plug, and if I decide to get a serial mouse and the driver software it will plug straight in. Although the ribbon cable is connected at the socket end it is

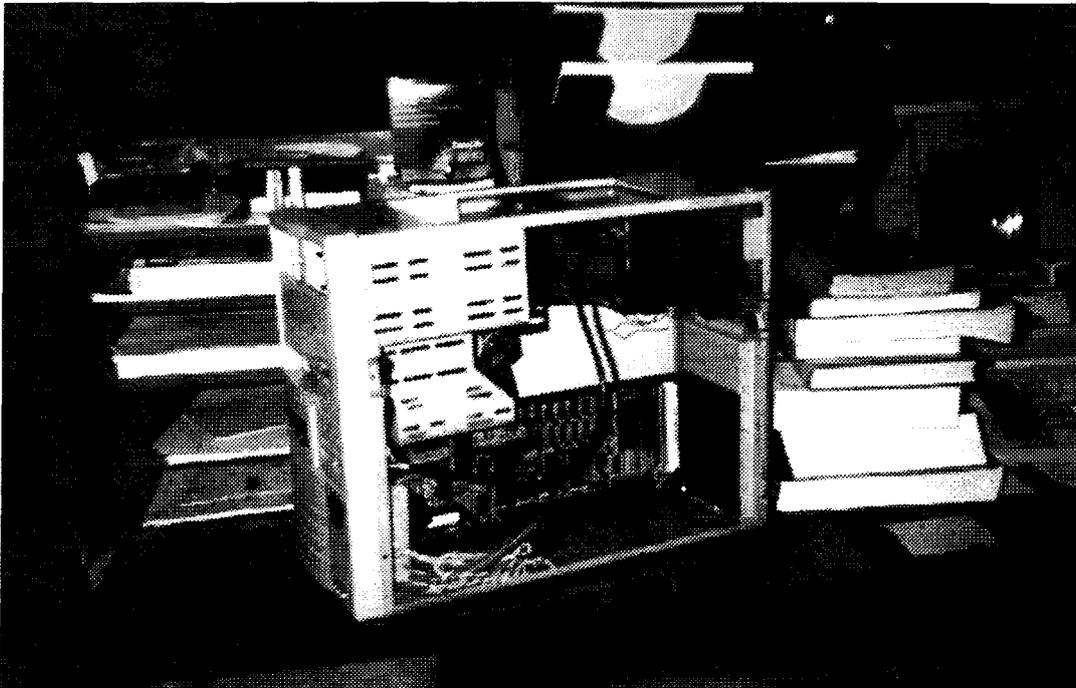
not yet wired to the QL motherboard as I don't have the pinouts for the PC serial mouse. It will be a few minutes work to wire ser2 up when (if) I get a mouse and the data for the connections.

At this stage I looked at the square indent for holding a logo or badge on the front panel, then cut the bit of the QL case that has "QL" written on it, beveled the edges to make it look neater and cemented it in place with rubber cement. I painted it dull grey and then an hour or so later rubbed it down with a cloth to remove the paint from the raised letters, making it easier to read the QL name. I also glued the word "Sinclair" from a QL assembler package onto the top left of the front panel, in the same style as the letters on the QL but much smaller. This Sinclair logo was treated in the same way as the QL one, it is just over an inch long and looks quite neat.

Next I drilled more holes in the metal back panel to accommodate the non standard array of QL sockets. There are three 3.5mm jack sockets, two

big for the job but I bolted on a bit of the old QL plastic case and drilled a hole in it for the new RGB socket (using the new drill bit this time). My existing monitor lead could then go straight into the new arrangement once I got it all working. The keyboard socket was mounted in a hole drilled in the front panel, where all keyboard sockets should be in my opinion. After all, when using a computer the user is in front of it with the keyboard, so why should the keyboard lead have to stretch all the way round to the back of the computer?

At this stage I started fitting internal connectors to the ends of all the leads coming from the new sockets. I did not intend to solder any leads onto the QL circuit board as I wanted to be able to easily unplug everything and remove the board for repair or replacement. So I wired up connectors for the RGB, network, speaker leads, ser1 etc. I decided not to bother with CTL1 and CTL2 as I have never used them, not even once at a QUANTA subgroup meeting or a QL show.



I then had to abandon work again for several days in hospital, a frequent occurrence in my life that I am quite used to. I also wanted to keep my QL in its existing setup for just a little longer, as I decided to print out all the college essays I had been working on before starting the slightly risky job of moving my working QL mo-

to replicate the QL network ports and one that brings the QL speaker output to the outside. This last is a switched jack socket that cuts off the QL speaker when a plug is in the socket, so I can silence the speaker if it irritates me by putting in a jack plug not connected to anything. The possibility of amplifying the QL speaker output has also been tested but it needs filtering, so I will be taking advice on this subject before I do any more of that.

There was already a circular hole for the PC keyboard socket but I used it to mount the QL RGB socket instead. This hole was actually far too

therboard into the new case. Once this was done I removed the duff QL motherboard and put in my working one, removing the 5V regulator and bridging its socket. Q-Plane was connected, the power connectors extended to mate with Q-Plane and then I powered up. At this stage there was no keyboard connected, just the RGB lead to the back panel so I could see the familiar dots when I switched on. Sure enough when I hit the switch the monitor lit up and the "F1 F2" window told me that the project was close to completion. Overwhelmed by my own genius I hurried to complete the construction.

In quick succession I wired up ser1, the network ports and the QL speaker, mounted on a bracket made from a bit of that tough black plastic that got me into trouble before. This turned out to be too quiet until I drilled a couple of holes in the panel just in front of it. I soldered little pins onto the reset switch terminals and connected the reset button from the front panel using the push on connector. The power LED was connected in the same way to pins pushed into the grip socket that usually holds the LED leads and this was tested; it worked first time, great!

Now ready to test the system thoroughly I inserted the keyboard interface, connected the keyboard and took my nerves in my hands to modify the Gold Card. I did only the essential modification, bridging the 5V regulator to allow the system to work with the Q-Plane power supply rails. Starting the system up with the Gold Card plugged into Q-Plane's second expansion slot I was delighted to see it work, though the message "Real time clock failure" was presented as well. I removed the Gold Card to build a support that would hold it in place using square Plastruct tubing bolted together.



Plastruct is really a modeller's material. It is model airplane plastic extruded, and various types are available, from tubes and pipes to "I" and "L" girders as well as the type I used, square box sections. It is intended for building models of bridges, construction sites, oil rigs etc. The stuff is strong, light, an electrical insulator and makes superb supports in electrical and electronic projects as long as heat is not going to be involved. I could have cut up some more bits of the original

QL case instead, but I had the Plastruct lying in a box and it is easier to cut and drill.

Once I had built the support to hold the Gold Card tightly in place and some anchors for Q-Plane the system seemed rock solid. I moved the floppy disk drive into the case next and folded the ribbon cable into the empty drive bay above. Currently this ribbon cable is too long really as it was intended to link the disk drive when installed as an external QL device. I will make a shorter version when my finances recover sufficiently to go to the component shop again.

Now with everything assembled and connected I tested again and all seemed well. Feeling very pleased with myself I took some cable ties and tied all the wires together in neat bundles inside the case then started using my QL in its new clothes to begin this article. After a while I decided to put the required 9-pin D-Sub plug on my Miracle Centronics interface and print a letter that I had written earlier in the week. At this point a disaster that had struck earlier but had not been noticed became obvious. Up to this point and as the project progressed smoothly I had been feeling more and more certain of my own brilliance, and I suppose it

did me good to find I had made another mistake.

The printer would not print! Ser1, so carefully connected, did not work at all. In order to connect the ribbon cable to the QL board I had been quite cunning. I bent back the ser1 socket, broke up the plastic and removed it, then straightened out all six pins. The pins were soldered onto

a little 6 pin PCB connector and the appropriate plug was attached to the ribbon cable that led from the external socket. This allowed me to remove the board easily and still gave a good connection to the new socket. The only trouble with this clever scheme was that it didn't work. After checking all the soldering again and the pin outs twice in case I had simply wired something in the wrong order the thing still wouldn't print.

I didn't panic, I just opened up the QL and examined all my soldering to see if it was obvious where I had gone wrong. It wasn't so I desoldered everything on ser1 and resoldered it. No joy. Oh dear, a disaster! At this point I did start to feel panicked. In the end I removed the QL motherboard on a hunch and found a dry joint on one of the original pins. I probably caused this when soldering the pins to the new PCB connector simply by taking too long about it. The heat would have travelled along the pin and melted the factory original solder, thus ruining the connection. Another lesson learned...

With everything working again I was able to print that letter, then came a bit of madness that added nothing to the functionality of the project but was something to fill a spare hour. There was an unused LED display on the front panel and a "turbo" switch. Nowadays the turbo switch serves no useful purpose as most PC users would not use software that required them to switch the machine down to a low clock rate anyway. The QL and most modern PC motherboards can't switch clock rates in any case, but I did decide to wire up the LED display to the 5V supply provided for it, and then to wire up the turbo switch to switch between the two displays.

This proved to be a tedious business though it wasn't really difficult. The LED display is set up by loads of little jumpers, and with the documentation it is a simple matter to set the jumpers to give "66" and "8" or "120" and "16" etc. The trouble was I didn't have the documentation, so it was trial and error time. After about an hour of fiddling with tweezers I got the LED to display "16", the Gold Card clock rate, and if the turbo button is pressed it switches to "HI". "Very silly, stop it at once"...

I topped off by wiring up a socket to mirror the QL's TV output. This might seem a waste of time but sometimes at QUANTA subgroup meetings I give talks, and this allows me to use a TV facing the audience and a monitor facing me, so I don't have to talk to everybody with my back to them. Eventually I hope to replace the Schon keyboard and interface with the Di-Ren interface and a really good IBM or Cherry keyboard and then I will really have a system to be proud of.

Well, the project is finished for now, so no doubt some of you will be looking at the photographs and wondering why I bothered. My QL now takes up a greater volume of the room and is slightly heavier, so was it worth all that effort?

Yes it was. Although slightly heavier in total and

bigger in volume the QL does take up less table space than it did before and is easier to carry and move. There is just one mains socket powering the QL and the monitor, as the monitor now has a plug on it to fit the mains outlet on the back of the QL's new power supply. Because of the modifications both the QL motherboard and the Gold Card run cooler and are provided with better regulated power from the carefully designed power supply. There is also masses of room for expansion, with plans for QUBIDE and a hard disk, or perhaps the IDE version of one of those Syquest 3.5 inch EZ135 drives that hold 128 Megabytes on a "superfloppy" disk. There is also room for other expansions like a 5.25 inch drive should one ever interest me, and I could even build two micro-drives into one of the two 5.25 inch drive bays if I really want to waste some time.

In short my QL looks better, is easier to move about, more robust than it was and will be more reliable. Expansion has improved enormously with Q-Plane providing three expansion slots instead of the usual single slot. A recent QUANTA subgroup meeting left me convinced I must have left something behind when transporting my system but no, it was all there. I am glad I did it. ■

CONFIG Level 2

Jochen Merz

We felt that a number of things were missing in the definition of level 1 of the QJUMP Standard configuration definition. Therefore, after a number of discussions, the following suggestions were made to be implemented on level 2.

First of all, re-configuring software you already had in previous versions is a very boring thing. Most of the time, all you do is set the old settings in the new file. This has to be made automatic. Therefore, the item structure is expanded to make room for a config-item-ID, i.e.

The configuration level 2 consists of the following information:

- Configuration ID
- Configuration level
- Software name
- Software version
- List of
 - Item ID (long) <== NEW!!!
 - Type of item (string, integer etc.) (byte)
 - Item Selection keystroke (byte)
 - Pointer to item
 - Pointer to item pre-processing routine

Pointer to item post-processing routine
Pointer to description of item
Pointer to attributes of item (item type
dependent)

End word (value -1)

[New Config Macros which handle the extra ID can be found on the current QPTR disk V0.28 or higher and can also be downloaded from the JMS Box 1 and 2.]

The ID should be unique for every item. There may be global ID names, which could be used by many programs (like the colourway setting), there can be unique "registered" ID names (which are preferred) and there may be "unregistered" local ID names. Global ID names should start with an underscore, unique ID names should start with a letter. For unregistered local IDs, the top byte of the ID has to be 0.

For all ID names, a list which is maintained by Jochen Merz Software is created, to avoid multiple name conflicts. If you wish to register for one or more ID names, please write to Jochen Merz Software and enclose an I.R.C. You may suggest one or more name, otherwise JMS will try to find a sensible abbreviation for you.

ID names consist of a longword (i.e. four characters). The first three characters have to be reserved by JMS, the fourth character can freely be assigned by the software house for the various items.

The function of the MenuConfig program

When the MenuConfig program starts up, the user selects the file to configure (which should contain one or more level 1 or level 2 config blocks). Level 1 blocks are treated as before (i.e. they can be printed or configured), but for level 2, there is an additional UPDATE facility. MenuConfig "learns" level 2 configurations and stores the settings of the item for any ID in a separate file, giving a "global" default configuration file. When the user selects UPDATE, the config block is scanned for IDs, and every ID is checked in the global default configuration file. If it is found, the preferred setting is automatically copied in the file which is to be configured. This way, updating programs is MUCH easier and nearly automatic. In fact, it could be made completely automatic (via parameter string).

Another advantage is, that the configuration can be made language-independent. The "learned" configuration of an English file could be used to

configure a German or French file, for example, provided that the same items have got the same ID's. Care should be taken for items, which are language-dependent filenames (i.e. help-files, auto-save filenames etc.), which SHOULD have different ID's, otherwise the German program would save to an English file or vice versa.

Local IDs are not stored by MenuConfig by default. You can configure MenuConfig from V3.21 onwards to enable the save of local IDs, but it may crash your system if you update files with the same "local" ID with different meaning, e.g. a string assignment is done to an ID which was defined as a word. There is no type check!!! We think it is safer not to save local IDs and update as follows: When a user wants to update a file containing local IDs, then MenuConfig has to "learn" the old settings from the old (already configured) version of the file, and these settings are then updated to the new version of the file. The local IDs are not stored anywhere else, as this could lead to ID clashes between different files containing the same local ID for different purposes.

MenuConfig stores the learned settings in a file called MenuConfig_INF on your current PROGRAM default device. It will try to read it from there the next time to execute MenuConfig. You can, of course, tell MenuConfig to load a different _INF file containing other configuration information, for example if you prefer having different configurations for colour and monochrome versions (!). When you terminate MenuConfig and you changed or learned new settings, MenuConfig asks you whether you want to update the _INF file, so that the settings are preserved for the next update.

An additional item type

It became obvious in MenuConfig, that a new item type "nothing" or "all" is required, which does not do anything automatic but calling the pre/post-processing routines. This is useful for testing own menus without having to mess around with unwanted texts. In addition, more information is required to be passed to these pre/postprocessing routines. We think, at the moment, of the following scheme:

A3, which points to a 4kBytes space, is negative indexed and provides the following information:

\$0000	4k	base of workspace passed to pre/postprocessing routine
-\$0004	long	MenuConfig's version
-\$0008	long	primary channel ID
-\$000c	long	pointer to working definition
-\$0010	2 word	primary window x/y size

PROGS

Professional & Graphical Software

Haachtstraat 92, 3020 Veltem, Belgium, tel/fax : +32-16/ 48 89 52

ProWesS *pre-release version
available now*

After more than a eighteen months work, we are very proud that we can announce our new product, which we believe will be an important new environment for the QL.

ProWesS is the "PROGS Window Manager". It contains the complete environment for running ProWesS applications. This is currently the pre-release version. We do not consider the package to be finished just yet, however, especially the ProWesS reader which is part of the package may be interesting to many users, and we would like to get comments about the system as soon as possible.

The ProWesS reader is a major part of the package. It is a hypertext document browser. This means that text files which include formatting commands and possibly links to other files can be displayed and read in this program. This is used in ProWesS to read (and possibly print) the manuals, and display the help files. The hypertext documents which are used by the ProWesS reader are in HTML format, the format which is popular on Internet to display World Wide Web pages.

All registered ProWesS users will get a free update to ProWesS when the full version is available. The package currently contains (apart from the libraries) the ProWesS reader, which allows you to browse hypertext documents (in HTML format), the ProWesS loader, which allows loading applications, including all the required extensions without reset, and some small sample applications (like a calculator). Many more utilities and installation software will be sent to you as the free upgrade to the full version !

ProWesS does not include the programming documentation. This is available via bulletin board and public domain software suppliers. The programming documentation is readable in the ProWesS reader, and partly in DATAdesign (the demo version is be included).

order your copy of ProWesS today ! for only BEF 2400

Payment terms :

ProWesS is available NOW for BEF 2400 (HD, excluding postage). It is normally distributed on high density (HD) disks. However it can be obtained on double density (DD) disks at an extra costs of BEF 100.

If you are VAT registered (specify registration number) or live outside the EEC, the amount to be paid is the total (including postage) divided by 1.21 (no need to pay too much).

Payment can be done by EuroCheque in BEF, or by VISA, EuroCard or MasterCard. Credit card orders can be handled by phone. For credit card, please specify name of card owner, card number and expiry date.

Postage : Costs of postage and packaging have to be added. You can choose the quality. Rate depends on no of programs.

copies	priority mail			ordinary mail		
	Belgium	Europe	World	Belgium	Europe	World
one	100	200	230	100	120	135
two	110	340	420	110	180	215
3 or 4	120	560	770	120	300	370
5 to 8	160	870	1250	160	550	675
more	200	1130	1610	200	800	1005

All prices are in BEF, including 21% VAT

-\$0014	2 word	primary window x/y origin
-\$0018	2 word	work area x/y size
-\$001c	2 word	work area x/y origin
-\$001d	byte	text info window number in working def
-\$001e	byte	work info window number in working def
-\$0022	long	window manager vector
-\$0026	long	pointer to filename of the file being configured
-\$002a	long	pointer to buffer containing file being configured
-\$002e	long	ptr to buffer of default directory
-\$0032	long	ptr to buffer of output device
-\$0040	long	colourway

WORKING COPY

If the configured file contains a flag "<<QCFC>>" BEFORE the "<<QCFX>>" flag (which can be generated with the new Macro MKCFCUT) then MenuConfig offers the user the choice to save a configured version without the config texts, to reduce the required file size to the minimum (as the configuration texts are not required anymore after configuration). Of course, a file treated this way cannot be configured afterwards anymore.

Programmers should take care that the configuration items come BEFORE the configuration texts, otherwise they will be cut away too. So make sure that the configuration texts are always the last section in your file!!!

LIST OF GLOBAL ID'S

_COL	Main Colourway
	Byte range -1, 0 to 3.
_COS	Sub-Window Colourway
	Byte range -1, 0 to 3.
_COB	Button Colourway
	Byte range -1, 0 to 3.

LIST OF RESERVED ID'S

APP.	APPMan
ATA.	ATARI-Rext
BLC.	BASIC Linker
BLG.	BASIC Linker
BLO.	BASIC Linker
CSH.	CueShell
CDK.	CueDark
DDE.	DataDesign
DIS.	DISA V2
DRN.	Disk Rename

EMN.	EASYMenu
EMU.	ATARI-Emulator
EPM.	EPROM-Manager
ESP.	EASYSprite
EXT.	EASYExt
FiF.	FiFi
HLP.	HyperHelp
MBT.	MultiButton
MCF.	MenuConfig
MEN.	Menu Extension
MPK.	MultiPick
OSP.	Operating System Preferences (SMSQ)
PAD.	Notepad
PDF.	Page Designer 3 Fonts 1
PDf.	Page Designer 3 Fonts 2
PDG.	Page Designer 3 General
PDP.	Page Designer 3 Page
PDp.	Page Designer 3 Pattern
PF..	Proforma & Applications
PRP.	PrinterPanel
PRM.	QPrommer
PW..	ProWes & Applications
QBS.	QBASIC
QDA.	QD (Tab Options)
QDE.	QD (Editor)
QDF.	QD (Files)
QDG.	QD (General)
QDS.	QDesign
QMK.	QMake
Q2_.	QPAC2 Main
Q2S.	QPAC2 Sysdef
Q2F.	QPAC2 Files
Q2B.	QPAC2 Buttons
QSN.	QSnap
SST.	Systat
SYP.	System Password
SYS.	System
TAB.	QSpread
TRA.	TRA Extension
WED.	WIN Ed
WSL.	WIN Select

With MenuConfig a number of new string attributes have been introduced. For example, if the attribute is "file" then MenuConfig will pop up the nice and handy file-select menu instead of the boring "read string" input. The new attributes are

```
cfs.file equ $10 ; string is filename
cfs.dir equ $20 ; string is directory
cfs.ext equ $30 ; string is extension
```

```
cfs.progs equ %001xxxxx00000000
; is reserved for PROGS. ■
```

Q Branch

Feeling out on a limb ?

Reach out for Q Branch

Suppliers of Quality QDOS / SMSQware

Thestaurus Upgrades

Geoff Wicks' Thesaurus is now at Version 3 with many new features including the ability to stuff the chosen word to the stuffer buffer. For an upgrade send £ 3.00 + 50p P&P and original disk.

P.O. Box 7
Poriskade
East Sussex
BN41 2ND

Tel. 01273-386030
Fax 01273-381577

UTILITIES.

FiFi, The File Finder	£ 24.00
QPAC 1	£ 20.00
QPAC 2	£ 40.00
QSUP	£ 36.00
Q Typ 2, Spelling Checker	£ 30.00
CueShell 2, File Utilities	£ 40.00
Disk Mate 5, File Utilities	£ 37.00
Q Spread, Spreadsheet	£ 77.00
Geoff Wicks' Style Checker	£ 15.00
Geoff Wicks' Solvit 2	£ 15.00
Geoff Wicks' Thesaurus	£ 15.00
QLerk, Archive accounts	£ 25.00
Printed manual for above	£ 20.00
DBEasy, Archive programs	£ 18.00
DBProgs, Archive programs	£ 14.00
DATAdesign database	£ 67.00
DATAdesign A.P.I.	£ 24.00
PFdata, printing from above)	£ 24.00
PFfontapack, fonts for LINEdesign	£ 89.00
PROforma,	£ 112.00

LINEdesign - The program that made this Ad :
£ 100.00

ARCADE GAMES

Arcanoids	£ 17.00
Firebirds	£ 17.00
Diamonds	£ 17.00
Doubleblock	£ 20.00
Super Games Pack	£ 40.00

POINTER ENVIRONMENT GAMES

Lonely Joker 2	£ 28.00
The Oracle	£ 24.00
QShang	£ 22.00
Bob's Pipes	£ 20.00
Mine Field	£ 19.00
Black Knight	£ 55.00
Brain Smasher	£ 20.00

NEW PRODUCTS !

Archive Made Easy.

Archive Made Easy is an 84 page, spiral bound book filled with information about programming in Archives own language. This comes complete with a 3.5" DII disk of sample programs.
£20.00

3.5" DII disks, preformatted to QDOS standard. Box of 10 disks and labels
£ 3.00 + 50p P&P

Preformatted Disks SuperBasic Linker

This is a brand new program from Wolfgang Lenerz. Designed to make the life of the programmer easier.

£ 24.00
Call for details.

PROGRAMMING.

QD 8	Text Editor	£ 58.00
QBasic		£ 23.00
QD + Qbasic		£ 75.00
QD + Qbasic + QLiberator		£ 120.00
QLiberator		£ 50.00
SPY	Text Editor	£ 15.00
Master SPY	Text Editor - enhanced version	£ 30.00
QPtr		£ 42.00
Easy Ptr	Pt 1 - Base	£ 40.00
Easy Ptr	Pt 2 - SuperBasic Interface	£ 20.00
EAsy Ptr	Pt 3 - C Interface	£ 20.00
QMake		£ 21.00
QMenu		£ 20.00
QMon / JMon		£ 44.00
QLoad / QRef		£ 15.00
DISA 2		£ 40.00
I/O Toolkit		£ 46.00
QDOS / SMSQ Reference Manual		£ 38.00
ProWesS	PROGS Windowing System	£ 50.00
SERmouse - Software and mouse		£ 40.00

SMSQ/E

For the following systems:-

Gold or Super Gold Card	£ 90.00
QXL	£ 90.00

Atari

(Without QJump C.D., or E level Drivers)	£ 90.00
Including Drivers	£ 112.00
(The Atari programs above need a QVME Card to run)	
With Mono screen driver	£ 134.00
All three versions	£ 180.00
With Mono screen driver	£ 134.00
All three versions	£ 180.00

Coming Soon : QPC ! SMSQ/E on the PC !

For full details on the programs listed above send a stamped addressed A5 envelope to the address at the top of this advert.

Demos Demos Demos

We have several demos of the programs available here. Write or Telephone for details or send £5.00 for the complete collection.

Writing a Zine on the QL

USA - Tim Swenson

Zine is short for Fanzine, which is short for Fan Magazine, which really is not a magazine at all, but more of a newsletter. E-Zine is short for Electronic Zine, which is a Zine that is published electronically, usually over the Internet.

The "QL Hacker's Journal" (QHJ) is both a Zine and an E-Zine. It is published in hard copy and electronically via e-mail on the Internet. It is also available on various QL BBS's around the world. The QHJ is created, edited, and published mostly using a QL. Some writing is done on a Z88. I've even written most of one article while flying at 25,000 feet.

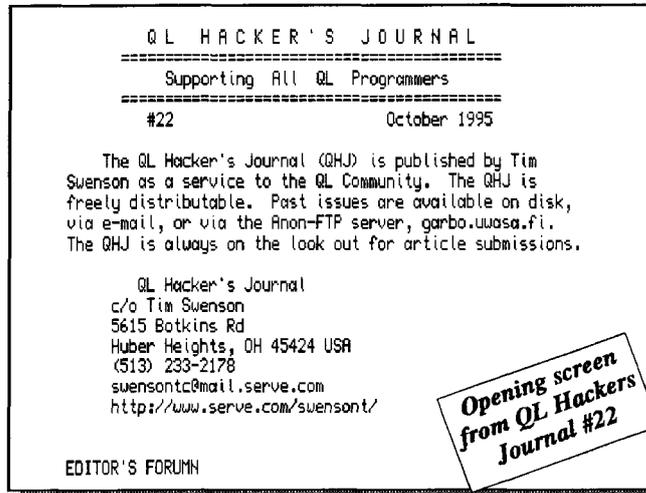
Why the QHJ Came About

The Editors' Forum in Issue #1 discusses why the QHJ came about, but I'll briefly cover it here. Back in 1990 I was the editor of the Capital Timex/Sinclair User Group newsletter. There were some articles I wanted to write, but I felt that the audience of the CATS newsletter were more QL users than QL programmers. Since there was no QL magazine or newsletter dedicated to QL programmers, I decided to create one. The QHJ is distributed for free, relieving me of the worry of owing anybody money if I ever call it quits.

Idea Generation

The first step in creating the QHJ is idea generation - getting ideas for articles. The QHJ is about 90% my writing, with an occasional article sent in by a reader. This means that I have to come up with a number of ideas for each issue.

To help spawn new ideas, I read a number of programming magazines (Dr. Dobb's Journal, C Users Journal, etc.) and scan through a number of programming books on my bookshelf. Sometimes these plant the seeds of an idea, other times it comes at random. Some of my best programming ideas come to me while I am washing the dishes (no kidding!). I think the labour of washing dishes allows my brain to wander freely.



Opening screen
from QL Hackers
Journal #22

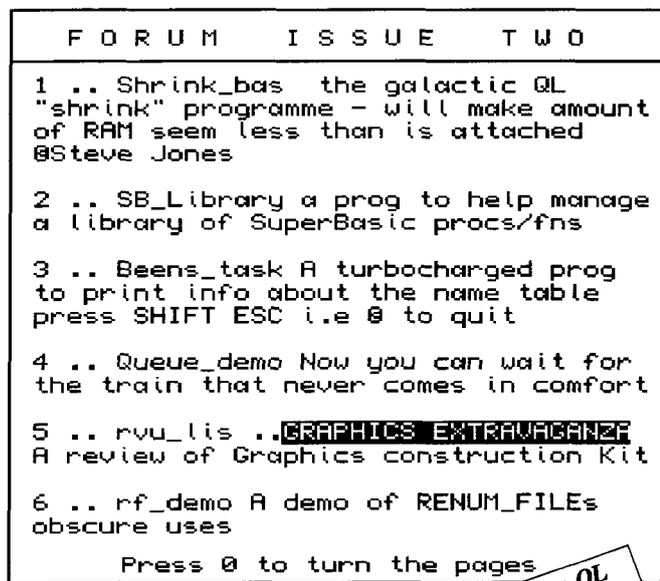
Article Creation

Once I have an idea for an article, I then have to sit down and work it out. If the article will be based on programming code, I first have to write the code. I mostly use MicroEmacs for writing the QHJ. Back before I had a Gold Card, Quill was real slow if you had more than 10 pages of text.

The reason I chose MicroEmacs is that I wanted an editor that would do word wrap and be portable across platforms. I already had a copy of MicroEmacs for the QL and knew I could get it for MS-DOS, and it did support word wrap. The only problem with MicroEmacs on the QL was that it was compiled with a 64K data file size. The source code support a

predefined data size, give the size at runtime, or changing the code to take up all memory. I did not want to have to enter the memory size each time I ran it, so I recompiled it to take up all available memory. This also meant that when I exit MicroEmacs it causes the QL to crash. Oh, well, I could live with that.

If an article or source code comes to me via e-mail, I have to transfer the file to the QL via MS-DOS disks, which means



Opening screen from QL
Forum (by Ian Bruntlett),
one of the first e-zines for the
QL back in the mid-1980s

that there is a CR at the end of each line that I need to get rid of. I use Metacomco's ED and an ED macro that I've written that will delete all CR's from the file. I have yet to figure out how to do this in MicroEmacs. Since it works in ED, I'm in no rush to get it working in MicroEmacs.

Once I've completed an issue, I use a print filter to print the text on my DJ520 and add bolding and underlining. In the most recent issue, I've used Quill because I can see where the page breaks will be and make sure that article titles are kept with the article and not split across pages. I don't think that I'll come back completely to Quill, as MicroEmacs gives me the format I want for the electronic version.

Distribution

Once I have the final print of an issue, I shrink it down with a copier and then cut and paste it to the format size that I use (roughly A5). This then gets sent to a printer to have them run off about 50 issues. To save some money, I fold and staple each issue myself.

My wife, because she knows Archive better than I, keeps my list of hard copy readers. She prints out the labels that go on the hard copy issues. I'm learning Archive, so I keep the e-mail list myself. I print out the current list of addresses right before I am ready to mail off the issue. The e-mail version gets a text head banner, where the hard copy gets a large print one.

From my Unix account at work I e-mail the issue to everyone on the e-mail list. Since MausNet QLers use dial-up links to the Internet, I've been asked to mail only one issue to a single MausNetter and he would then post it to the QL forum on MausNet.

Once I've mailed an issue out, I then post it to my Web Page where it is available 24 hours. I've noticed that the QHJ section of my web page is the more popular section. In May 1996, there were 129 requests for issue #23.

Format Decision

The initial idea behind the design of the QHJ was to focus on content and not print style. With over 120 e-mail readers versus 40 hard copy readers, keeping the format simple and not pretty (like using Line Design) is still the way to go. Plus, I have no idea of how many other QLers get the QHJ via BBS's. This is why I don't use any of the Desktop Publishing programs for the QL.

Conclusion

I've never felt limited by the QL when publishing the QHJ. Yes, I would like to do some nice graphics, but since most of my readers are receiving the issue via e-mail (pure ASCII), it would not work.

If you are thinking about using the QL to do a newsletter (be it for family, hobby, or business) both IQLR and QL Today demonstrate what can be done with the QL graphically. But before you put all your work into making your work look good, focus on content. People want newsletters of substance not ones that just look pretty. *[We'll do our best! -Editor]* ■

The SBASIC PE Kit

John Miller

As the QL world has evolved, more people have moved to use the Pointer Environment (PE), taking advantage of its relatively consistent and easy to use interface with the screen, keyboard and mouse. Many will have tried to develop programs using Superbasic or SBASIC with QDOS or SMSQ/E. Some will have found this a bit tricky, as I did.

Albin Hessler's Easyptr 3 makes it much easier to design sprites and menus. Norman Dunbar's tutorials, originally published in the QUANTA Newsletter, supplement the manuals, and are a great way to learn how to get to grips with Easyptr 3 (parts 1 and 2).

After writing a few programs using these facilities, I realised that I was writing code to do the same sort of thing several times over. I therefore created the "Pointer Environment Kit", which is now available in the Public Domain.

The PE Kit is intended to provide some base procedures, functions etc. to build an SBASIC application, avoiding repetition and simplifying application building. It is also intended to minimise the coding required for some of the 'standard' functions. It includes some sprites, menus, and example programs, but the main useful bits are some procedures and functions, and the text describing how to use them. The procedures and functions handle buttons and other menu objects which you want your users to handle.

The source code can be modified, and extended. The PE Kit has been placed deliberately in the Public Domain, in the hope that others will be keen to improve and add to it. I will be glad to receive such additions, and will continue to

maintain an up-to-date version for distribution.

The best way to develop a new PE application in SBASIC using the PE Kit is in the following stages.

- 1) Outline design and plan of the menu relationships
- 2) Design any new sprites required
- 3) Design any new menus required, incorporating the sprites
- 4) Build an appendix, incorporating the menus
- 5) Start a new program by copying the PE Kit SBASIC code
- 6) Define your buttons etc. in the PE Kit procedures
- 7) Add your own application code where necessary
- 8) Test and document your new PE program

The PE Kit refers to a number of menu items using names which are hopefully helpful. The text file which is included with the Kit gives a full description, but the following will give you some idea of what it is about.

Buttons in the PE Kit are defined as loose items in menus. You can click on a button, and select or unselect it. It can be unavailable (i.e. no action is currently possible). When you click on it, you may want something to happen. Some examples are:

- 1) To change it from selected to unselected or vice versa. Your program probably needs a variable to reflect which state it is in. I generally refer to such variables as 'flags'.
- 2) To cause some code to happen. This is likely to be a procedure to do a certain job, such as saving a record.
- 3) To invoke another menu. The PE Kit is recursive, so you can cause a button on the first menu to fire up another menu. If the user then chooses a button on the second menu, this might start a third menu. With the PE Kit, the code is reused, and your program shouldn't get lost (I hope!).
- 4) To unset another button or buttons. These are often called 'radio buttons', because when you push one, the others become unselected. The PE Kit handles groups of two or three radio buttons, but you could easily extend the code if you needed more. Each button has a flag to reflect its status.
- 5) To quit the menu.

All the above capabilities are handled in the PE Kit. In addition, buttons may be initialised to selected, unselected or unavailable states.

If you want the user to type data into a PE menu, you have to define an application window in the menu. PE Kit is designed to handle smallish data input, such as a number or a short description.

When you click on it, you probably want one or

more of the following.

- 1) To put the cursor in the window
- 2) To show the existing data and allow it to be changed
- 3) To validate the input, and show an error if faulty
- 4) To return to the original value if ESC is pressed
- 5) To accept the input if ENTER is pressed
- 6) To accept the input and move on to another data input field if up or down cursor keys are pressed.

The PE Kit handles all of these, plus initialising the data in the window, and displaying it in a defined format.

Sometimes you want the user to choose from some options, and see a description of the currently chosen option. An example might be where you want them to choose between 'Metric' or 'Imperial'. There might be several choices, but essentially a fixed list. The PE Kit handles these also as an application window, and refers to them as switches. Clicking on the window should display the next choice in a list, or the first one, if cycling round the list.

Another case is also where you want the user to choose an option, but the actual number of possible choices isn't fixed, or it might be valid to choose more than one option. The usual way to handle these cases is with a scrolling list of options. Easyptr can cope with quite complex scrolling, but the PE Kit only handles one option (although it is a very useful one!). It looks like a simple, vertical scrolling list of items, from which the user can choose.

You may often need to send a message to the user, such as 'File not found'. The only action required from the user is to say 'OK'. The PE Kit contains a message procedure and a pre-defined message menu to make this straightforward.

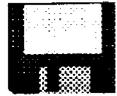
Other cases may arise where you want to pass a message to the user, but want some response from them, normally 'yes', 'no' or 'ESC'. The PE Kit has a pre-defined menu and function for this as well, which is very similar to the messages menu above.

The PE Kit contains some worked examples, showing how to use it in some detail. A knowledge of SBASIC or SuperBasic is required, and you need to own Easyptr 3. The Public Domain package contains all the items described above, and more. It also includes the Norman Dunbar articles on Easyptr 3 (courtesy of Norman and QUANTA) which provide an excellent description of its use.

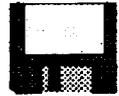
Given these, the PE Kit can help to build Pointer Environment applications quite quickly, with some help in avoiding errors. If it helps to encourage others to build applications, it will have done its job. On with the PE Kit! ■

QUBBESoft P/D

Purveyors of QL Hardware & Software



Public Domain & Shareware Software



Over 70 Disk's of Public Domain & Shareware Software including Psion Xchange V3.90L, C68 'C' Compiler V4.20, QL Emulator (AMIGA) V3.23, Molecular Graphics V5.12, LineDesign 2 Demo, MicroEmacs V3.11, Page Designer 3 Demo, Climes, QFax, Ghost Script V2.61, QTPI, Hackers Handbook, etc.,

£1.00p per Disk + Postage & Packing

Stamped addressed A5 Envelope or International Reply Coupon for current Catalogue

LineDesign EZ Clip-Art Collection 1

Over 100mb of Clip-Art files on a SyQuest EZDrive Cartridge. All files will load directly into LineDesign 2. Various themes are catered for such as Food, Backgrounds, Cartoons, Music, etc Also Includes 263 PFF Fonts and Thumbnail Catalogue for ease of selection.



£25.00p inc Cart & Cat

Inc P&P (UK) +5% (Euro) +10% (ROW)



PD EZ Collection

Over 100mb of QL PD & Shareware Software on a SyQuest EZDrive 135 Cart. Includes all the Software in the current QUBBESoft P/D Catalogue as follows:-

Disks PD 1 to PD 14, Disks SPECIAL 1 to SPECIAL 54, C68 V4.20a 8 Disk set, Ghostscript V2.61 6 Disk set, MicroEmacs V3.11, GNU Utils, XLisp V2.1d, Star Trek Reference, Adv '93, COWO Screens, QItaly Disks 14 to 29, LD2 Clip-Art Food 1 to Cartoons 4.

£25.00p inc Cartridge

Inclusive of P&P (UK) +5% (Europe) +10% (ROW)

SyQuest EZDrive 135 IDE Removable Medium Hard Drive

- Affordable unlimited storage
- Speed & reliability of a fixed drive
- Compact with 3.5in cartridges
- Super fast back-up & retrieval
- Let family & friends use their own Cart
- Stop using endless floppy disk libraries
- Store all your Clip-Art and Fonts
- Have quicker access to files than disks
- Compatible with QUBIDE
- Can be used as Master or Slave
- Faster than a CD ROM drive
- Cartridge holds 128mb

Cartridges

£17.00p each

EZDrive 135 Internal IDE Kit £135.00p

Inclusive of P&P (UK) +5% (Europe) +10% (ROW)

Cartridges

£17.00p each

QUBIDE AT/IDE Interface

Allows you to connect modern AT/IDE Hard Drives, including the EZDrive 135, to your QL. A massive amount of storage space can now be made available for your programs and files. Compatible with Super Gold Card, Gold Card, Trump Card and most memory expansion systems for the QL, also Minerva and SMSQ/E.

Now only £55.00p

Inclusive of P&P (UK) +5% (Euro) +10% (ROW)

QPLANE Powered Back Plane

QPlane is a powered back plane for the QL, primarily designed to be used with a PC Mini Tower Case and a PC PSU. QPlane has two power connectors, known as P8 & P9, which marry up with the special power connectors of the PC PSU to supply all the power requirements for your QL and expansion units. QPlane has 3 expansion slots.

£25.00p

Inclusive of P&P (UK) +5% (Europe) +10% (ROW)

Products in Pipeline

The Aurora: The QL's Graphics Card, is a replacement for the QL's Motherboard. Introduces new Graphics allowing higher resolutions varying from 512x256 upto 1024x768 depending on type of monitor connected. Can be connected to VGA, SVGA and Multisynch Monitors as well as the normal QL Monitor. Qimi compatible mouse interface also included.

The Super Duper Gold Card: This will be the successor to the Super Gold Card. Uses Motorola Cold Fire processor which is equivalent to 68040. Different memory configurations upto 64mb using industry standard 72 pin SIMM. Bi-directional Parallel Port. On board switch mode power supply unit which will detect input voltage and adjust output voltage accordingly.

**38, Brunwin Road, Rayne, Braintree,
Essex. CM7 5BU. UK.**

**Tel: +44 (0)1376 347852 Fax: +44 (0)1376 331267
Email: QUBBESoft@aol.com**

BUGS 'N' FIXES

England - Peter Tyler

Following the note regarding the Line Design Demo disk in the April/May QLToday, Peter Tyler has commented that the fix suggested might be a little difficult for some QLers to implement. The 'secret' fix mentioned in the original QUANTA contribution was simply to rename the fonts on the disk so that they would be instantly recognised by the PROFORMA search procedure. For those with Toolkit 2, just use:

```
WREN flp1_pf_fnt_,flp1_
```

For those who don't have Toolkit 2, then it is necessary to copy each font file in turn back onto flp1_ with the two prefixes deleted, and then delete each of the old files in turn because there is not enough space on the supplied DD disk to hold both sets of font files. Thus:

```
COPY flp1_pf_fnt_name_pff TO flp1_name_pff:  
DELETE flp1_pf_fnt_name_pff
```

Then the desired font can be loaded for use on the main screen without any further problems.

MORE BUGS 'N' FIXES

Bangor, Wales - Dilwyn Jones

Recently I had a problem loading a text file into a text editor. The file had been converted from a PC, and seemed OK when viewed using the QPAC2 files menu, but just would not load, giving the error message 'end of file' every time I tried. Eventually it dawned on me that all that was needed was to add a linefeed to the end of the file. So I wrote this short BASIC program to fix the problem. Make a backup copy of the file (in case something goes wrong) then run this program. Toolkit 2 is required.

```
100 INPUT 'Filename ? ';f$  
110 OPEN #3,f$ : f1 = FLEN(#3)  
120 BGET #3\fl-1,char : REMark check  
last character in file  
130 IF char <> 10 THEN BPUT #3,10  
140 CLOSE #3
```

The program simply reads the last character in the file, then adds a linefeed (CHR\$(10)) unless the last character was already a linefeed. The problem was that the converted PC text file had a carriage return or a formfeed character at the end of the file, thus causing problems for QL input.

Dynamic Buttons

Sweden - Per Erik Forssén

When I read the article in the May/June issue of QL Today about the Button Frame and BASIC I got an idea; The combination of SBASIC and QPAC2 buttons is great, so why not let the buttons be controlled by small SBASIC programs?! In this way you could have buttons with a behaviour altering as the system- defaults altered.

QPAC2 lets you create buttons in many different ways. However, once set up the behaviour of the buttons is fixed. Imagine for example that you want to pass the DATAD\$ to a program started via a hotkey. This is not possible from the QPAC2 HOT_xxxx commands, as the value of DATAD\$ at the time of execution of the HOT_xxxx command is passed to the program instead of the value at the time you execute the hotkey. To alter the behaviour of a button you need to remove the button, alter the action of the associated hotkey, and finally re-create the button.

With the help of SBASIC this is no longer necessary. My solution for dynamic buttons is something like this:

The dynamic button starts an SBASIC clone that runs an SBASIC program.

The SBASIC program then in turn performs the real action of the button.

The SBASIC program's dynamic behaviour is controlled by some kind of global system variable. This can be for example the environment variables present in the C68 compilation system (or the DIY Toolkit SET command). Personally I use the EEPROM in SuperHermes as it preserves the settings after power off.

Now a cookbook instruction for creating a dynamic button:

1) Create a hotkey for the dynamic button in your QPAC2 startup file. The hotkey should start the SBASIC Thing with a DO or LRUN command as argument. For example:

```
ERT HOT_THING('m','SBASIC';'DO win1_But  
tons_Make_btn') BT_HOTKEY 'm','Make'
```

These commands create a button which builds the current programming project on my system. The HOT_THING command will associate the execution of the SBASIC Thing with the <ALT>+<m> keypress. The argument to the SBASIC Thing will tell SBASIC to 'DO' the file 'win1_Buttons_Make_btn'. Finally the BT_HOTKEY command will create a button with the text

'Make' that will execute the hotkey <ALT>+<m> when you press the right mouse button, or Enter over it.

2) Create the SBASIC script files to be run when the hotkey is executed. For the hotkey above I have this script file:

```
REM
REM SBASIC script for make button;fil
ename 'win1_Buttons_Make_btn'
REM myself$='Make script'
JOB_NAME myself$
fnm$='win1_C68_make'
IF FOPEN(#3,fnm$) THEN BEEP 4000,30:ELS
E:CLOSE#3:EX fnm$;CONTRACT$(IPCRDP$(0))
:END IF
RJOB myself$
```

This rather cryptic program requires a lot of explanations. The lines

```
myself$='Make script'
JOB_NAME myself$
RJOB myself$
```

are only there to remove the SBASIC clone after make has been started.

The FOPEN command is used to test if the file to be executed is really there, if not the SBASIC will just BEEP 4000,30 and comit suicide. If the file really is there it is executed with the argument CONTRACT\$(IPCRDP\$(0))

The command IPCRDP\$ reads a page in the SuperHermes EEPROM (in this case page 0). I use this page to remember the name of the current programming project of mine. As this page is always 16 characters long, I have filled the end of it with spaces, and these spaces need to be removed somehow.

The solution is the command CONTRACT\$, which strips whitespace from a string. CONTRACT\$ is not a standard SBASIC command, but in this example it is equivalent to:

```
DEF FN CONTRACT$(str$)
RET str$(1 TO ' ' INSTR str$-1)
END DEF
```

In this way I have created a set of useful buttons that aid me a lot when I develop programs. Isn't SMSQ/E a great thing?!

I apologise if this article is technical, but if it is, this is not altogether my fault. First QPAC2 is a

WE SUPPORT SINCLAIR

QBOX USA

COMPUTER BULLETIN BOARD SERVICE
(810)254-9878

We support *all* SINCLAIR COMPUTERS
(QL, SPECTRUM, ZX81, Z88, Thor, QXL)

- Now in our 3rd year on-line round the clock since October, 1993
- Full message area and File Download areas
- We carry all popular SINCLAIR message areas from Europe
- Calls from 14.4k--300 baud are welcome
- QBOX - USA runs on a SINCLAIR QL with Super Gold Card, Hermes, QUBIDE, 200MB drive, USR sportster 14.4 modem

→ NO FEES CALL US

Being as old as they are, Spy cannot be loaded and passed a filename to open. You cannot pass any parameters to this editor, unlike some of the latter day programs, QED for example. On EXECing, you are asked to supply a filename to open or create. This must be done before you can proceed, and there is no facility to get a device directory so you must CTRL C back to SuperBasic, do a DIR and then switch back into Spy to type the name in - if you are as forgetful of filenames as I am. This was par for the course in the old days.

Of course, there is another foible. If you use a monitor, then you EXEC one version of the program. If you use a TV, then you have to EXEC another. Very strange as it is possible to determine the start up configuration and select a different console size.

After asking for a file name, Spy works out a suitable default workspace size and asks you to confirm that it will be enough - for normal editing this will usually suffice, but if you are going to be cutting and pasting between a number of open files, or adding great chunks of text, it might be better to add on a bit more.

If the file does not exist, the Spy offers the configured minimum workspace for you to accept or alter. The default is 20K, but this can be changed.

This is one bit of working with editors that I am not fond of. I don't know how much space I am going to need in advance. I prefer my editor to just load the file and let me get on with editing it. Having said that, I know how much of a processing overhead this would add to the program and thus reduce response times. It is also more memory efficient to allocate a single large chunk for the file than to have to allocate lots of smaller chunks. More chunks mean more overhead and a greater chance of fragmenting the heap.

Once a file is loaded, the program actually indexes it line by line. This is carried out once only and allows for very quick navigation within the file. This adds an overhead of 4 bytes per line of text in the file. The screen then appears and is split into an upper work area and a lower status area. Commands are also input through the status area.

Working with my normal 'test file' which is 2000 lines of assembler source code and approximately 130K, both editors performed very quickly and adequately.

There is a slight difference in the way that these two versions of the same program handle files that are open. Both can be configured to allow any number of files 'on the stack', but Master allows different 'views' of each file that is open.

Spy, for example, lets you open 10 files at once, memory permitting. You can edit each of these files at one specific place only. You only have a single view of each file.

Master, on the other hand, allows up to 5 views to be opened on each file so that you may have 10 files open and be editing in 5 different places in each - this could be confusing !

You can configure the number of views allowed in Master, but the limit of 5 per file may not be changed. By default, there are 15 views allowed. This means that you can open 15 different files with a single view each, or have 3 files open with 5 views each. On the other hand, you can have some other combination of files & views, subject to the above mentioned limits.

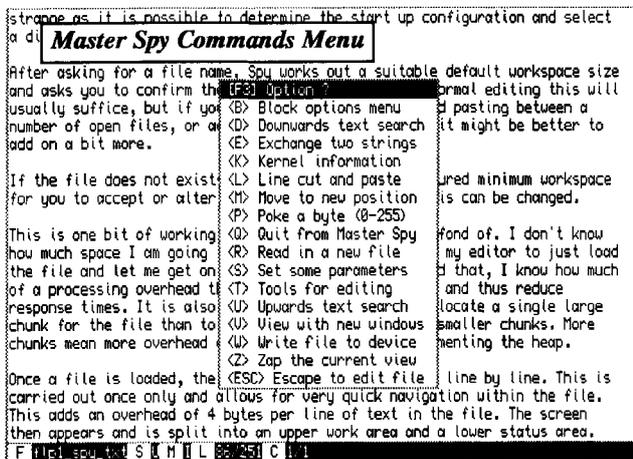
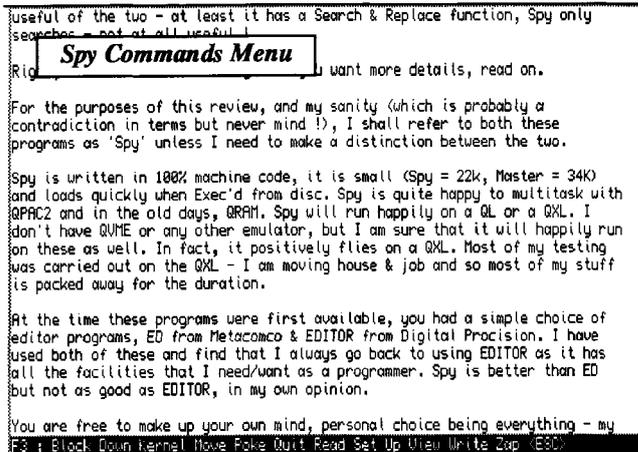
I tend to work with 2 views per file maximum as this reduces the stress on my brain when trying to figure

out where I am and what I am supposed to be doing. It is very useful when typing an assembler source program and you need to add a new EQUates, for example. Having one view at the top of the file with the EQU statements and another for 'proper' editing is a good idea. It saves wear and tear on your cursor move-

ment fingers when zooming back and forth between the two areas.

The 'stack' mentioned above is simply where Spy keeps a list of its files & views. Switching between each is a matter of CTRL and the up or down arrow keys.

What more can really be said about an editor ? Spy has its own set of commands for blocks, lines, and for the whole file. Search and replace is available (replace only in Master) and blocks of text can be cut from



one file and pasted into a number of others. Master allows a single line block to be cut in a single command - no need to mark the block first.

There are two different methods for accessing commands in Spy. Press F3 and you get a list of commands or a menu of commands in Master, press the initial letter of a particular command to either have the command carried out or for further choices. This is very good for beginners using Spy.

Once you have a bit of practice under your belt, you can cut out the menu and simply press CTRL and the appropriate letter to access the commands directly. Both methods are handy as the command letter is the initial letter of the command - CTRL W for Write (to file or printer), CTRL R to Read another file etc.

One 'odd' command is Kernel (F3 K or CTRL K), which displays a list of status information about the editor as a whole. You can see the current contents of the 'local' & 'global' cut & paste blocks, the list of views or open files, and so on. Pressing any key to exit from Kernel leaves the kernel window displayed on the screen and does not refresh the display - this can be confusing.

You need to force a refresh by moving the cursor up or down a line or you don't know where you are.

Having a number of files (or views) open at any one time requires that the window for each view needs to be able to be sized or positioned on the screen in some place where you can access it and possible see more than one open view at a time. Spy allows this.

Each view is created and then pressing F4 will bring up the Size Position sub menu. The window can then be sized and positioned on screen as required, but, the maximum size of a window is 512 by 256 and nothing will let it get bigger than this. Running on a QXL or some other emulator allows bigger areas of screen. Spy cannot be used in these areas - it is restricted to an 'ordinary' QL's screen area.

Master allows the screen to be sized and positioned in pixel dimensions or character dimensions. This is not in the manual, but can be found in the UPDATE_TXT file supplied on the disc. Spy only allows character sized adjustments to be made.

Spy does not come with a configuration program and certainly knows nothing about Tony Tebby's CONFIG blocks. Spy can, as mentioned above, be configured and a configuration sheet is supplied in the manual for your use.

The config sheet tells you what can be configured, what valid ranges are allowed and the offset into the

file where this value lives. The manual gives details of how to 'poke' values into the Spy program. Basically, this is what you do :

Exec the editor & load in its own code file. Set the ADVANCED status mode so that you can see the offset into the file displayed on the status line. Ensure overwrite mode is on.

Having done that, search for the start of the parameters area, CTRL D and PARS then enter. CTRL D = search Down the file from the current point. You are now at the start of the program's parameters.

Each parameter, or group of parameters has a 2 character flag for which you must now search. For example, SL is used to find the number of views/files on the stack, while MN is used for the number of lines, columns, the paper colour & ink colour for the main window. The flag is listed on the configuration sheet.

Having found the flag, the cursor is moved right until the offset in the config sheet is the same as that in the file. You can now 'poke' a single byte, word or long word into the subsequent 1, 2 or 4 bytes to change the startup configuration. Not at all easy, and fraught with danger. Remember

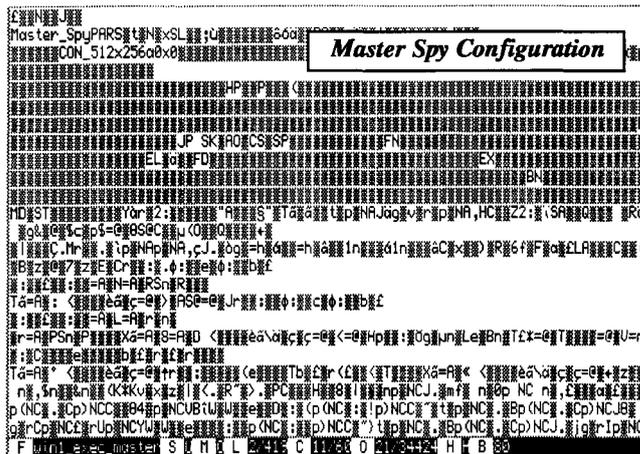
to save a spare copy of the original file BEFORE even thinking of attempting this sort of stuff.

Changing the default help file, for example, requires that you poke in the file name length, then type (in overwrite mode) the characters of the filename. Space filling to the end of the maximum 36 characters if a longer filename was previously used.

Having played around with this configuration, I have found a much easier method - CTRL M for MOVE and you are offered a choice of Top, Bottom, Offset, Mark or Line. Pick O for Offset. When requested, simply type in the offset for the parameter you want to change - it is on the config sheet. Press ENTER and there you are, ready to poke your new values.

A strange way to configure any program, if you ask me. Especially if you also have to work out the individual byte values when dealing with 2 byte parameters. Mmmmm!

Ok, so I don't like this and overall I would not switch to using Spy or Master Spy for my own programming work. I can recommend it if you want a cheap and easy editor to work with. It does have its faults, the configuration and the need to refresh the windows after the Kernel information has been shown. Master requires a refresh after F3 has been pressed or it leaves the commands menu superimposed on the text being edited - a quick page up then



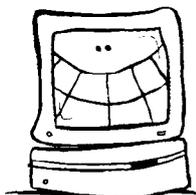
page down wipes it off.

The editors come with a fairly simple manual giving a brief overview of the program's development and its features. There is a more detailed explanation of the editors method of storing files in memory and a number of pages explaining the various commands. Missing from the manual is a proper explanation of what exactly you see when you call up the Kernel command. Having said that, however, the manual is quite adequate.

Do give it a try, there is an older version in the Quanta Library. It is on Disk MD01 (Misc Demo 01) and dates back to 1988 I suspect. If you like it, then get in touch with Roy Wood at QBranch for a copy.

My overall score is 75% for Master Spy and about 65% for Spy. ■

Letter-Box



First, the positive response:

... Dear Dilwyn, Congratulations on your new responsibilities on the QL scene. I liked the first edition of **QL Today**. It had a nice mixture of articles with something for everyone, whatever their level of QL use. ... (Geoff Wicks) ... last week I received a copy of **QL Today**. I like the magazine and I would like to support it, therefore I have written an article ... (Per-Erik Forssén) *That's excellent!* ... I also wish to express my appreciation for the efforts of all those who were involved with the first issue of **QL Today** ... (Ken Brickwood) .. I have just been reading the first issue of **QL Today** and am impressed with the contents and the speed with which the QL community has managed to replace IQLR with another professional magazine ... (Jeff M. Wass) ... I have now received two copies of the new **QL Today**, and would simply like to let you know that I think they are excellent ... (Mike Simpson) ... Keep up the good work! (M.J. Thornton, Darren Branagh, March R. Renick, F.W. Gregory, Stephen A. Hall, John Hall and many more) ... A vast improvement on IQLR (P.H. Thanner) ... I like the friendliness of the editors + their openness. Many thanks - it's excellent + well presented (Eleanor Patrick) ... Please continue in this manner (Kurt Miller) ... The best QL Magazine yet! Great (John Miller) ...

Plus a lot of very positive response in German, through the mailbox and by phone - thank you very much! This is very encouraging!

Peter Tyler, Aughton, England, writes:

First of all, let me add my voice to the welcome for **QL Today**. I perceive that the style is intended to emulate the extant IQLR, which I interpret as a semi-professional platform for the more expert amateurs and traders. Nothing wrong with that, except that it is challenging my ability to keep up with things! However, I do puzzle

over the Beginner's Basic series. Surely there are no candidates these days for this level of tuition?

What do you, the readers, think? Admittedly, issue 1 was put together in a great hurry. Write and let us know - are we pitching ourselves at too high a skill level, too low, or about right? We want to make QL Today what you the readers want it to be, and the survey has shown that we have a considerable amount of readers who liked Beginner's BASICS most [Editor].

R.G. Wall, Rixensart, Belgium, writes:

Your new magazine arrived today, on the tail-end of my IQLR subscription. I shall be renewing when the time comes. Can you twist someone's arm to review the current range of QDOS/SMSQ compatible printers. Things are changing so rapidly that backward compatibility is becoming a real issue. A friend who admittedly lives in the backwoods of Menorca, has just upgraded with an Epson inkjet and has had to throw out all the old PC programs (I have in fact the QL they used before the PC, which I use as a backup).

This is an important point - we do need to print articles about printers, but we don't have enough printers between us to do it ourselves. Any readers out there with a modern printer who feel that it would be of interest to other readers to include articles about these printers? Reviews, help articles, printer drivers, all welcome! [Dilwyn].

I think your friend has not told you the real truth. All PC programs (and, of course, all QL programs) support EPSON 9pin printers and 24pin printers. Unlike other printer manufacturers, even the most recent EPSON inkjet printers like the Stylus Color II or Stylus Color 500 still emulate 9 and 24pin printers. I went through all generations of EPSON printers and I never had any compatibility problems. If this issue does not fill up too quickly, I will add a review about my EPSON Stylus Color II, otherwise it will go into the next issue [Jochen].

Just to show that we haven't managed to get it all right and please everyone so far:

G. Caulwell, Manchester, England, writes:

I am not a subscriber, but have read a friend's copy of the first issue of **QL Today**. On the whole, not too bad, but a little room for improvement.

- 1) Too much of a German slant (no disrespect to the Germans, but the majority of QL users are here in England). I also disliked the idea of the German supplement, although it seems that the Germans will pay extra for it. How about a French and Dutch supplement too, as there must be as many QL users in France, Holland and Belgium as in Germany?
- 2) Not enough software reviews. Work forces me to have a PC, I won't be a QL user for long, so I won't subscribe, but I would have hoped you'd have reviewed a lot more than you did.
- 3) Some of your articles are basically free adverts (eg. for this QPC device).
- 4) Why was there no pictures in the show report in issue 1?
- 5) The article about the Epson Stylus printer was useful for those with one of them, but why didn't you print a full review?

Issue 1 was done in only a week by Jochen Merz, and so he had to gather articles at short notice from those close

to him, which meant people in Germany and fellow QL traders. My involvement in issue 1 was more as a contributor than as an editor (excuses, excuses...) [Dilwyn]

ad 1) As Dilwyn said, everything had to be done in a rush, because there was not a lot of time and the idea was to have the issue ready for the USA show. I think issue 2 was different. Why do you dislike the idea of a German supplement? It does not affect you at all - you do not pay for it. I can't see something wrong to support people who cannot read English very well or at all. We all liked the idea and the response is good - especially as the German club was virtually dead. Also, the German add-on covers three countries - Germany, Switzerland and Austria (and we also have Dutch readers of the German part). The Dutch and French clubs are alive (unlike the German club was), so there is no need to be competitive. [Jochen]

ad 2) We would love to publish more software reviews. Software suppliers and prospective reviewers, get in touch with us! Where is your contribution to help reducing the shortage of software reviews?

ad 3) The QPC articles could perhaps be seen as a free advert, but equally it is important to show that such an important product has had extensive testing before release. And also, some QL users are having to buy PCs for use with their work, so we can keep such people as part of the QL community if they have a QL emulator available on their PC to run their existing software.

ad 4) I was asked to write the show report (Quanta AGM, Tynemouth) some time after the event. Had I known at the time, I would have taken pictures of course. I agree, we should try to include a few pictures in future. Again, issue 2 is different. [Dilwyn]

ad 5) The printer itself will be reviewed in this issue. To be short, it is highly recommendable! [Jochen]

Darren D. Branagh, Ireland, writes:

... I am writing to you to say how much I enjoy your new magazine, QL Today. ... QL Today is the best thing to come along in a long time. I totally demolished the first and second issues in a matter of days. I would gladly pay a little extra to see a cover mounted disk on it each month - a bit like the glossy PC magazines, any chance this might happen? ... Thanks for a great mag and keep up the good work!!!

Thanks, Darren, this is really a good idea - and thanks also for offering help on the disk copying. I thought quite a while about your suggestion, and here is what I would suggest to you (all the readers, feedback please!): we can't raise the price now, it is a bit too late. Also, adding a disk every time would mean that somebody would have to do **a lot of copying** - the print run of issue 2 is already 800 (!) and we're hoping to be over a thousand very soon. The problem is that we cannot really justify charging for the disk, the copying, an extra bag which connects it to the magazine (otherwise the postage in Germany would be much higher) and for the extra postage just for having two or three files on the disk. I like the idea very much and I would suggest that we add a disk at the end of each volume (i.e. every year) containing the files of all the 6 issues. We could make this 10 or 12 pages thinner to make up (a bit) for the costs of the disk and the extra postage.

In the meantime, if you do not want to type listings in: you can get the files from the JMS-Mailboxes

(0203-502013 and 0203-502014) which should not cost you more for downloading than paying for an extra disk every magazine. I also think that these files will be quickly distributed to other bbs's. If you do not have modem facilities available, there's still the JMS PD-Service offer: you can get files out of my mailbox on a disk, which costs DM 1,- per disk, DM 1,- per program and DM 5,- for postage and package. A full list can be obtained if you send an International Reply Coupon. [Jochen] ■

80MHz QXL - Caution!

USA - James Hunkins

I was very concerned by the last issue's article on increasing the QXL's clock speed. Up to now the increases have been fairly 'minimal'. However, the speed discussed in that article will most likely reduce the life expectancy of the processors.

Unfortunately, I don't have any numbers on the problem, but it can be a real problem.

Increasing the speed of a processor beyond the manufacturer's recommendations may decrease the life expectancy of the processor even if it seems to function properly at the higher clock speed. The degree of the problem depends on the case temperature of the processor more than the clock speed.

By increasing the clock speed, the processor uses more energy and therefore generates more heat. By attaching a fan and supplying proper system circulation the case temperature can be held down. However, if this is not done properly or the speed is increased so much that the fan/cooling method can not compensate, the processor will definitely have a decreased life expectancy.

How far you can push your clock speed depends on the individual processor chip (variances in the manufacturing process affect this), the cooling methods you use, and just what you expect for a life expectancy. The clock ratings on a processor from a manufacturer are tested for a given case temperature and life expectancy.

With proper care, it should be feasible to push some chips from 20 to 25 MHz with probably negligible loss of chip life. However, going beyond that will most likely have an impact and should be avoided if possible. Instead, it is better to buy a higher clock rated processor (don't forget to also improve your cooling methods).

A final note is that other chips in your system that respond to the clock increase (such as the PAL) also have speed limits and can generate additional heat. However, the processor on the QXL will most likely be the biggest concern when pushing the clock rate.

[I spoke to Stuart Honeyball of Miracle Systems Ltd. about this article, and he also

expressed some concern about driving the QXL to such extreme speeds. He cited a case where an over-clocked QXL appeared to function normally one day, then refused to work at all the next, and the only difference was that the second day was warmer than the first! It seems clear that pushing the QXL to these extreme speeds is risky, and if the chips blow up, you'll have only yourself to blame, as will the author of the article if or when it happens to his QXL. Stuart agreed most QXLs would happily go up to 25MHz and is considering offering an upgrade once the QXL2 becomes available - Dilwyn]

Compatibility

France - Tony Tebby

Once again there is a subdued growling noise in the corner and a small white rectangle comes crawling out of the fax. For the third time, it is an SMSQ/E user requesting a small change to make SMSQ/E more compatible with QDOS on SBASIC KEYROW operations. Easy, you might

think, no more than a dozen lines of code and it's done. In this case there is, however, a good reason why there is a difference between SuperBASIC and SBASIC. The original definition was taken from the Spectrum to support freeware games that just take over the whole machine and so the horrific side effects of the SuperBASIC version of KEYROW were unimportant. Existing SMSQ/E users are not so likely to want to use this type of program, but they are likely to want to execute several programs simultaneously, so the SBASIC version is cleaner. Making the SBASIC version dirtier might well cause more complaints than it removes as I had already seen the day before

..... The day before, the small white rectangle that came out of the fax was a complaint that the CONSOLE IOB.ELIN operation in SMSQ/E was directly compatible with the QDOS version and did not, therefore, correspond to the documentation whereas the Minerva version was both logical and "correct". Earlier versions of SMSQ/E also had a logical and "correct" IOB.ELIN operation, but, unfortunately, the Turbo Toolkit INPUT\$ function exploited the QDOS bug, and so INPUT\$ did not work properly with SMSQ/E. No hope of correcting the Turbo Toolkit, so SMSQ/E was "downgraded". Only a few users of SMSQ/E need to use the

JUST WORDS!

Software for Writers and Word Lovers

You've asked for it!

Version 3.00 of QL-THESAURUS is a major revision of the program incorporating new features requested by users. These include:

- * Backwards scrolling of both word lists and word groups
- * The ability to "pick up" the chosen word for entry into a word processor
- * Simplified antonym searching
- * User configurable ink and border colours
- * A revised data base.

UPGRADE PRICE: £3 or 3 IRC's. Return master disk. Upgrade free by purchase of another JUST WORDS! program.

The Complete Range:

SOLVIT-PLUS 2 <i>Dictionary for word puzzles</i>	£15
QL-THESAURUS <i>23,000 words in 1,000 groups.</i>	£15
STYLE-CHECK <i>Improves your writing style.</i>	£15
WORDS PACKAGE <i>(Solvit-Plus 2 + Thesaurus)</i>	£25
WRITER'S PACK <i>(Thesaurus + Style-Check)</i>	£25
ALL 3 PROGRAMS	£35

Available direct from the author, from QUO VADIS or from Q BRANCH.

Pay by sterling cheque drawn on UK bank, Eurocheque in guilders or International Postal Order.

(£5 = 13 guilders)

Geoff Wicks, Bertrand Russellstraat 22, 1097 HL Amsterdam, Netherlands.
Tel: (31) (0)20 - 692 1521. Netherlands Bank: Postbank number 4111942

Turbo Toolkit but nearly all users would benefit from a correct IOB.ELIN. How do you trade off the small benefits for the mass against the major needs of a few users?

For Minerva, which is an "upgrade", there is no problem. The Turbo Toolkit is clearly wrong and it is ridiculous for all software developers in this market to have to work around problems which would simply disappear if Turbo Toolkit were corrected. Minerva is optional, and if you really want to use software from a supplier who is either unwilling or unable to correct even simple problems, you can always go back to JS.

SMSQ/E however is not optional. Some SMSQ/E users need to use badly written software, or software which exploits the "garbage in, garbage out" error handling of most QDOS facilities, or even worse, as in the case of INPUT\$, software which deliberately exploits a QDOS bug. SMSQ/E needs to provide a way of supporting such software.

Note that there are many different reasons for writing badly behaved software for QDOS.

* UNIX / AmigaDOS / MSDOS / Windows do it a different way and therefore QDOS is wrong. Some programmers must find a way of forcing QDOS to behave like UNIX / AmigaDOS / MSDOS / Windows by poking values into the system variables or linking in extensions that change the behaviour of the device drivers, scheduling or memory management.

* Using incorrect parameters for certain operating system calls can provide facilities whose existence was not previously suspected, even by the designers and writers. These features might save a few bytes of code or give a few percent of speed improvement.

* Finding ways of exploiting "undocumented features" makes the software writer feel that he is very clever and he hopes that lots of other people will say "isn't he CLEVER". (In my world, "clever programmer" is a term of abuse reserved only for those programmers for whom the terms incompetent hack, irresponsible tinkerer, bungling idiot are all too weak.)

* Coding errors are a constant source of problems. SMSQ/E calls are, in general, much cleaner than QDOS calls. This can cause problems where deliberately, or more often, accidentally, program code relies on certain not very well defined values being returned (for example, the top half of a register always being cleared, or an address register always pointing to a zero byte).

Some software has problems with timing. For example, QLiberator externals only ever worked because the initialisation within SuperBASIC

took more than 1 tick (20ms), on faster processors and with the much improved SBASIC the entire operation was completed in much less than a tick and so the "co-task" had finished before the calling routine thought it had started. The only cure is to modify the QLiberator code to make it safe, but QLiberator is not currently supported so a patch is necessary.

* QDOS did not provide all the facilities that one could ever wish for, and so "cheating with discretion" was sometimes necessary to provide a particular facility. This is, possibly, the only justifiable reason, and no software in this category has ever really caused a problem for SMSQ/E.

More than 25% of the design effort of SBASIC was in providing ways of emulating SuperBASIC "undocumented features". More than 50% of the support effort has been in finding ways of working round the problems created by badly behaved software. The SMSQ/E approach has been to try to provide compatibility with the old, while adding new facilities (for example SuperBASIC EOF never worked reliably on NET or SER: correcting the bug could cause problems so a new reliable EOFW function was introduced). Even so, the work to ensure compatibility has been enormous.

Unfortunately, the worst software is that which is used by other software writers: top of the nasty league, by far, is Turbo, followed by Turbo Toolkit with QLiberator and C68 3rd and 4th. So far however, apart for "C68", all the work of ensuring compatibility has been on one side, my side (and Laurences Reeves' side for Minerva).

This makes an interesting comparison with other computers. I am writing the document using Word V7. All of a sudden, POOF!

**"This program has performed an illegal operation and will be shut down.
If the problem persists, contact the program vendor."**

I contact the program vendor: "Contact Microsoft Support" is his reply.

I contact Microsoft Support "Are you sure that you do not have any software (device drivers for the graphics card or printer, for example) from another supplier installed on your PC?"

"Well, yes."

"Tough" (in fact, they are much more polite than that, they have special training in taking 5 minutes to say "Tough" while making you feel that it is all your fault.)

Ah! Another little white rectangle is crawling out of the fax! ■

SINCLAIR

Z88

OFFICE/FAX 01494-871319 (EEC) *W.N. Richardson & Co.*

MOBILE: 0850 597650

6 Ravensmead
Chalfont-St-Peter
Buckinghamshire SL9 0NB

THE IDEAL PORTABLE COMPANION FOR THE QL

THE CAMBRIDGE Z88 A4 NOTEBOOK

WITH BUILT-IN WORD PROCESSOR, SPREADSHEET, DATABASE, BASIC, CALCULATOR, CLOCK, ALARM, CALENDAR & VT52 TERMINAL.
USES 4XAA ALKALINE CELLS (ca. 20 HOURS) *



CAMBRIDGE Z88 A4 NOTEBOOK COMPUTER & ACCESSORIES

IT IS RECOMMENDED THAT ACCESSORIES ARE BOUGHT SOON AS THE PRESENT STOCK IS RUNNING LOW. NEW STOCK WILL HAVE TO BE REPRICED. ALL UK PRICES INCLUDE VAT.

DESCRIPTION	UK	USA
CAMBRIDGE Z88 COMPUTER	£99	\$130
32k RAMPACK	£18	\$25
128k RAMPACK	£38	\$50
512k RAMPACK	£80	\$100
1Meg RAMPACK	£160	\$200
32k EPROM PACK	£18	\$25
128k EPROM PACK	£24	\$33
256k EPROM PACK	£55	\$75
EPROM ERASER	£32	\$43
PARALLEL PRINTER LEAD	£39	\$50
SERIAL PRINTER LEAD	£12	\$16
* MAINS ADAPTOR (230vac; 6v,500ma)	£10	\$20
TOPPER (PROTECTIVE COPVER)	£12	\$16
CARRYING CASE (PLASTIC)	£10	\$12
'Z88 MAGIC' A BETTER USER HANDBOOK	£15	\$20
FILE TRANSFER TO OTHER COMPUTERS		
SPECIAL Z88-QL SERIAL LEAD	£10	\$15
COPY DISKS OF QUANTA PROGS IMP/EXP & ARCHVIE EXPORT	£2	\$3
PCLINK KIT (For PCs)	£25	\$35
Z88 TO MAC KIT	£25	\$35
Z88 TO BBC KIT	£15	\$20

UK PRICES INCLUDE VAT; USA PRICES DO NOT. FOR OTHER NON-EEC COUNTRIES DEDUCT 10% FROM THE UK PRICES.

POSTAGE UK £5. EEC £15. USA £20. OTHER COUNTRIES £30.

ALL THE STOCK IS WARRANTIED FOR 90 DAYS. IN THE EVENT OF REPLACEMENT BEING ARGREED, BUT THE ITEM BEING OUT OF STOCK AT THE TIME, A REFUND WILL BE MADE PROVIDED THE ITEM IS RECEIVED IN GOOD CONDITION.

QL & PC COMPUTER USERS WILL FIND THE CAMBRIDGE Z88 ESPECIALLY USEFUL FOR WORK AWAY FROM THE DESKTOP, WITH TRANSFER PROGS DATA CAN BE SAFELY EXCHANGED WITH THEIR DESKTOP SYSTEM.

W.N.RICHARDSON & CO CONTINUES TO PROVIDE FULL SPARES AND SERVICES FOR SINCLAIR COMPUTERS, QL & THE CAMBRIDGE Z88

The QL on the Internet

Berlin, Germany - E.C. Herrnsdorf

There is plenty of talk about the Internet, especially the World Wide Web these days. Apparently, however, it is mainly PC users and some Mac users who surf the Net. Therefore, most computer-related information on the Net is about these computers, but if you have a closer look you'll find web pages for other computers - including the QL. For instance, there is even an improved version of XChange (yes, you know, Quill, Abacus, Easel and Archive for the Thor) that can be downloaded and run on an expanded QL.

I realize that there are only relatively few users who have connected their QL to a modem, and even if you have a modem you'll still need appropriate browser software to look at World Wide Web pages. There are, however, no WWW-Browsers that I know of for the QL. ProWesS is able to read the format of WWW pages, HTML, but it is not a browser. If you have a modem, you'll probably be able to use other areas of the Internet using your communications software. You'll be able to send e-Mails and maybe even download software from ftp-sites. There was an article on how to do this in a recent issue of IQLR.

If you want access to the World Wide Web and the QL-related information on it, there is no other way than to use a PC or Mac (I don't know if there is a browser for STs), download the software to that system and then convert the downloaded files to the QL format using a conversion utility - I'll describe how to do that later in this article. If you don't have a PC or Mac yourself, you can ask friends or maybe use the one you have at work. If you live close to a university, there is a big chance that they have computer rooms for students, and these computers can often be used with a guest ID even if you are not a student.

If you finally get your hands on a PC, you'll need more than just a modem to surf the Net. First, you need an Internet access provider. There are many companies providing Internet access - just check your local computer magazines. You can also contact one of the major on-line services like CompuServe, America Online, MSN, T-Online, and Prodigy. All of these offer full Internet access in addition to their own online services. There are enormous differences in access fees, so contact more than just one provider before you choose

one. As for CompuServe, for instance, the monthly subscription fee is currently US\$ 9.95, which includes five hours of free Internet access per month, and US\$ 2.95 for each additional hour.

Next, you'll need a Web Browser. This is a piece of software that will enable you to load, read, print, and control web pages. The best known browser is Netscape, but Microsoft Explorer and Mosaic are also popular. Some browsers are shareware, others are commercial products. If you are a subscriber to CompuServe, you'll get a special version of the Mosaic browser free with the CompuServe Information Manager software.

Now, you can finally access the World Wide

Web. Start the browser and wait for the software to make the connection to the Net. Then, select the 'OPEN DOCUMENT' button or select 'OPEN' from the 'FILE'-menu and type in the WWW-address of the page you want to read. If you don't know an address you'll need some sort of starting point,

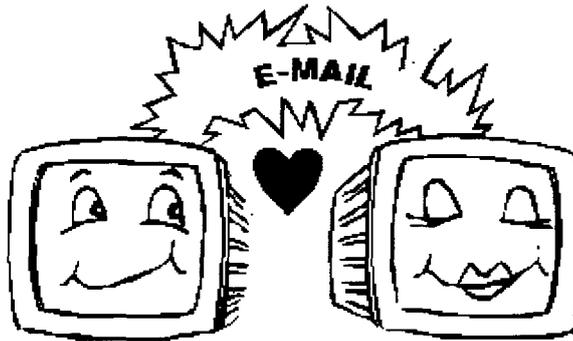
therefore you should first use a search engine. The best-known search engines on the Net include Yahoo, Lycos, and Webcrawler. Their addresses are:

<http://www.yahoo.com>

<http://www.lycos.com>

<http://www.webcrawler.com>

Thus, type in the address of one of these search engines and wait for the document to load. Then, enter a search string in the search field, i.e. 'QL' or 'Sinclair' or 'Sinclair QL' or even "'Sinclair' NOT 'Spectrum'". There is no space in a QL magazine like QL Today to give you a detailed description on how to use a web browser for a PC. Further, not all available browsers work the same way. Therefore, I'll just concentrate on the basics. WWW-pages are always opened using the OPEN command from the FILE menu. You must then type in the address of the page you want to open and press Enter. Many pages will also have links to other pages. These can be accessed by clicking on a so called Hypertext; most browsers display Hypertexts underlined and in blue. Downloading of software works almost the same - just click on the name of the program you want to download. The browser will, however, probably ask you where you want to save the downloaded files. Because access is faster on a hard-disk you should save all downloaded files on drive c:\ first. To do this, type c:\progname and press Enter, with progname being whatever this file is called. You can later copy all your down-



loaded files to a floppy. But again, all browsers are different: just experiment with your browser to find out how to save pages or how to download files. However, if you have any specific questions, please feel free to contact me by fax or e-mail - you'll find my address at the end of this article.

Let's assume you have successfully downloaded a file from the Internet and now want to convert it to the QL. To do this, you need a QL file conversion utility. The best one I could find is Digital Precision's XOVER by Stefan Schmidt. This tool allows you to import and export files from QDOS, MS-DOS, and TOS and to convert QDOS text files. (Converting files is quite simple. Actually, this text was initially written on a QL, then transferred to a PC running WordPerfect 6.1, and back again to the QL.) *[The easiest way is to use SMSQ/E - it allows you to write to and read from DOS disks directly, you don't need any conversion utility or density fiddling-Editor]*

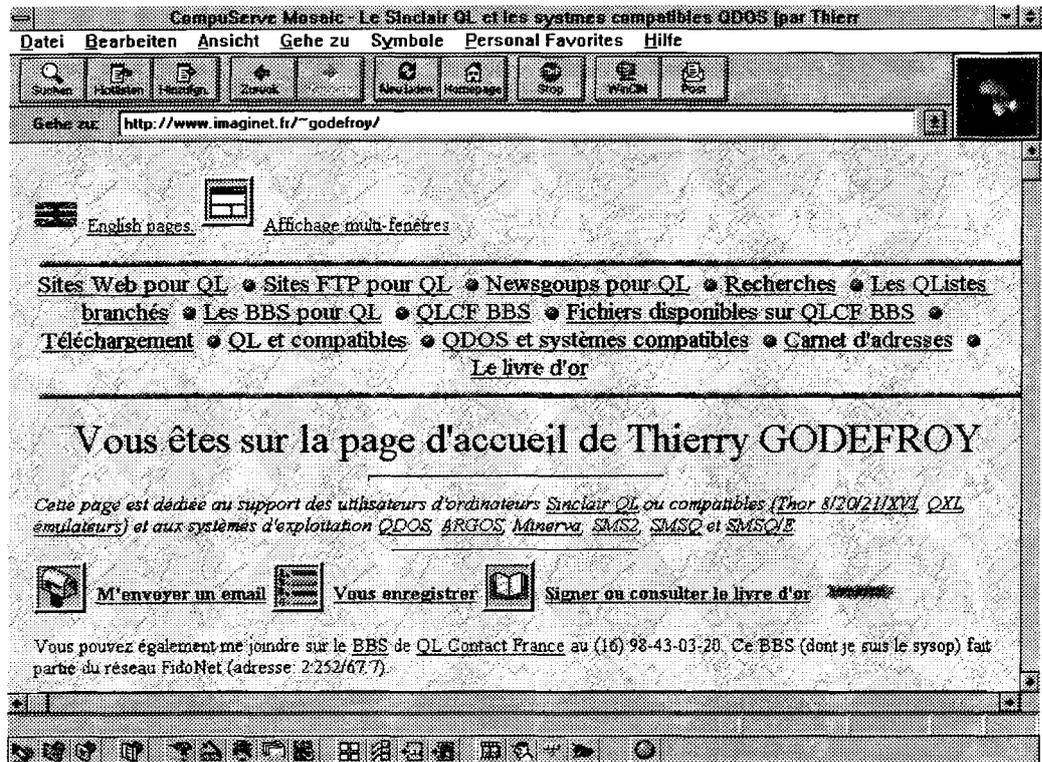
First, copy the downloaded files from the PC's hard disk to a floppy. This floppy must be formatted to 720K - NOT 1.44 MB! You can format a standard 3.5" disk to 720K by typing

```
format a: /F:720
```

at the DOS prompt on the PC. But be warned, ANY DATA ON THAT DISK WILL BE ERASED BY FORMATTING THE DISK (just like on the QL)! Further, this works only with MS-DOS 5.0 or higher. Formatting a 720K-disk with older versions of MS-DOS is a little more complicated. Just type format /? for help. You may also encounter some problems if you try to format a pre-formatted disk to 720K. Therefore, buy unformatted disks only. Then, copy all your downloaded files from the hard-disk to the floppy by typing copy c:\progrname.* a: (again, progrname is whatever you called the file). Besides, many programs downloaded from the Internet will be packed and you must unpack them before you use them. To do this, you'll need either PKUNZIP.EXE or LHARC.EXE (these are the

most popular 'zip'-programs on the PC).

If you have your files on the disk you can finally forget about the PC and turn it off. Then, turn on your QL and start XOVER (this must be EXECed) or whatever conversion utility you use. Press CTRL C to activate the program and insert the PC disk in flp1_. Select the 'IMPORT FILES' option and copy the file(s) from the MS-DOS disk to a QL device other than flp1_, e.g. flp2_, ram1_ or even mdv1_. If you have downloaded a text file, you should then select 'CONVERT QDOS TEXT



FILES', else there will be problems with some special characters, especially umlauts. You can only convert QDOS text files on a QDOS device, not on a MS-DOS device. You MUST NOT use the 'convert QDOS text files' option with executable files! Quit XOVER, load Quill, text87 or whatever you use to view text files and read your WWW pages on the QL. If you have downloaded a program, EXEC or LRUN it.

OK, finally, here's a list of WWW and ftp sites related to the QL or Sinclair computers in general. These include articles, a complete index of 'QL WORLD' (nvg), programs and pictures:

ftp - sites:

- ftp://ftp.nvg.unit.no/pub/sinclair
(This is one of the best sites for all Sinclair computers)
- ftp://garbo.uwasa.fi/ql/
- ftp://maya.dei.unipd.it/pub/sinclair_QL/
- ftp://ftp.gui.uva.es/sinclair/QL

WWW-sites:

- http://www.imagnet.fr/~godefroy/
(This is a very good web sites for the QL and compa-

tible systems. It is available in both English and French. It includes links to FTP-sites, other web-sites, and even a list of wired QLers'.)

<http://www.dk-online.dk/users/erlingj/xchange/index.htm>
(This is the unofficial - official' site for PSION Xchange. You can download the latest version of Xchange from this page.)

<http://www.di-ren.co.uk/>

<http://www.uni.mainz.de/~roklein/ql/index/html>

<http://www.serve.com/swensont>

<http://www.nvg.unit.no/sinclair/ql-contents.html>

<http://www.wuarchive.wustl.edu/systems/sinclair>

<http://www.xs4all.nl/~wij2/sinclair.html>

<http://ourworld.compuserve.com/homepages/peta>

Besides, it may be a good idea to support these web sites by actually uploading any information, articles or pictures you have. As with QL Today, these sites can only survive if QLers support them. If you have any questions, just send me a fax to +49-(0)-30-7929461 or e-Mail me (100410.675@compuserve.com or europro@ccmailer.wiwiss.fu-berlin.de).

Hints for the QXL-Card with MS-DOS

Munich, Germany - Franz Krojer

1. In one of the last issues of IQLR the configuration of the QXL operating systems (SMSQ, SMSQ/E) was explained. But there are also situations when the QXL card is started directly from MS-DOS. Often (and I do it too) the command "BREAK ON" is executed in the "AUTOEXEC.BAT" file on the PC. When SMSQ or SMSQ/E are then running, it can happen, that after typing CTRL/C the MS-DOS operating system gets control over SMSQ irregularly: the "QL" is crashing. Therefore, a "BREAK OFF" should always be executed before SMSQ or SMSQ/E are started. I have written a batch file which does exactly this:

```
@echo off
break off
c:
cd \qxl
smsqe %1
break on
```

You can name this batch file as you want. I named it "qxl.bat". The directory of this file should be part of "PATH". Then I start with "qxl" or "qxl /".

2. Assuming you have two MSDOS partitions "C:" and "D:". You have created two "QDOS" file systems, for example you have said "format win1_30" and "format win2_40". Now you have 70 MB for your QXL card available.

Assuming that on your MSDOS disk drive "D:" there is enough free space and you want to create a third QDOS partition "win3_" on it. You could split the MSDOS partition "D:" into "D:" and "E:", but this would be a lot of work: save all data of "D:", delete partition "D:", create the two partitions "D:" and "E:", restore the data.

But there is a simpler way: use the "SUBST" command. First create a MS-DOS directory on Your drive "D:", for example "MKDIR D:\QXLWIN3". Then execute the command "SUBST E: D:\QXLWIN3" (do this in your "AUTOEXEC.BAT"). Now you have created a virtual disk drive "E:" which points to "D:\QXLWIN3". Now when executing a "DIR E:" a message like "no files found" should appear but not one like "drive E: does not exist". You can start your QXL operating system now, saying "format win3_40" for example.

Of course, if you have a CD-ROM or more partitions, the name of your new virtual disk drive could not be "H:" or "I:" instead, and depending on your DOS drive your QXL drive will also change, it would be "win6_" or "win7_".

"SUBST" should be executed before MS-Windows is started. It cannot be used, when the MS-Windows "32 bit file access" is enabled (but the default is, as far as I know, "16 bit file access"). I don't know if Windows-95 or Windows-NT have a similar command for creating such a virtual disk drive.

3. A QXL file system looks like a normal file in MS-DOS or MS-Windows with the name "qxl.win". Therefore it can easily happen, that this file could be deleted accidentally, for example when executing the command "DEL *.*".

To hide the file "qxl.win" for such an accident, simply execute the MS-DOS command "ATTRIB +H qxl.win" for each of the "qxl.win" files. After that, executing a "DIR", for example, will not display your hidden QXL file system in MS-DOS. To see it again, you must execute the command "ATTRIB -H qxl.win". ■

EPSON Stylus Color II

Jochen Merz

Mr. Wall (see "Letter Box") asked in his letter for printer reviews. I also had a number of enquiries from other customers about printer recommendation, and I can highly recommend the EPSON printers.

First, the specification: the printer is not very big (43 cm x 25 cm if not in use, otherwise it is 43 x 58 as the paper feeder points outwards); it should fit onto every desk if you want to replace an old printer. The printer has two connectors: the common Centronics parallel port which can be used to connect it to a SuperGoldCard, any serial/parallel converter or directly to the parallel port of an ATARI or PC (with

TF SERVICES

MINERVA

The ORIGINAL system upgrade

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

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

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

MKI..£41 (£40) [£43] RTC(MKII)..£66 (£63) [£76]

GOLD CARD (incl. SUPER) compatible

QL SPARES

Keyboard membrane	£12.50 (£12) [£13.50]		
Circuit diagrams	£3.50 (£3) [£4]	68008 cpu	£8.50 (£8) [£11]
8049 IPC	£8.50 (£7.50) [£9]	Serial lead	£11.50 (£11) [£12]
1377 PAL	£3.50 (£3) [£4]		
Power Supply (by sea)	£17 (£16) [£21]		
8301, 8302 or JM rom set	£10.50 (£10) [£11]		

Other components (sockets etc.) please phone

QL REPAIRS

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

£27 including 6 month guarantee (UK)

QBBS - UKs first QL scrolling Bulletin Board

Megabytes of files. Messages to/from UK/Belgium/Holland/USA/Italy/Germany for a UK phonecall. TANDATA callers add SIX zeros (000000) or wait for 3 seconds of modem tone if dialling manually.

01344-890987 (up to V32bis)

TF Service now on the Internet!

E-mail: tony@firshman.demon.co.uk

<http://www.firshman.demon.co.uk>

superHermes

A major hardware upgrade for the QL now ready and shipping

- All Hermes features (see below for list) PLUS full 19200 throughput not affected by sound
- IBM AT keyboard interface (for UK/US/Germany)
- HIGH SPEED RS232 two-way serial port, ie 5000cps throughput (supergoldcard/qtpi zmodem) at 57600bps
- THREE low speed RS232 inputs (1200 to 30bps). Driver for SERIAL MOUSE supplied. Other uses - RTTY/graphics tablet etc.
- THREE spare i/o lines (TTL) with GND/+5v
- Capslock /scrolllock LED connector
- Turbo/keylock connectors
- 1.5k user data storeable in EEPROM

All this on a professional board about twice the size of the 8049 coprocessor it replaces

Cost (including manual/software)	£92 (£87) [£90]
IBM AT Keyboard (acer)	£24 (£23) [£27]
Serial mouse	£13 (£12) [£14]
Capslock/scrolllock led (+wire&plug)	£1.50 (£1) [£1.50]
Keyboard or mouse lead	£3.50 (£3) [£3.50]
High speed serial (ser3) lead	£4.50 (£4) [£4.50]

Specify PANEL MOUNTING or IN-LINE

I²C Interfaces

Connects to Minerva MKII or any Philips I²C bus

Parallel Interface gives 16 input/output lines. Can be used for logic level output eg model train controllers. Input direct to motor drivers (eg L293/L298) **£26 (£24) [£27]**

Analogue Interface Each gives 8 analogue to digital inputs, and 2 digital/analogue outputs. For temp measurement, sound sampling etc. **£31.50 [£29] [£32]**

Data sheets (analogue/parallel)

Control software/manual (Superbasic extns) **£2.50 (£2) [£3]**

(First interface purchase includes free 15D/9D lead)

COMPUTER CLEANER

High Spec AC mains filters and spike removers

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

2-way adaptor ... **£14** 3-way adaptor ... **£18**
4-way trailing socket **£24 (all UK prices)**

Prices incl. AIRMAIL postage & packing. Prices are EC except UK (Europe outside EC) [outside Europe] unless stated otherwise. Ring for UK prices or see Quanta. Payment by Mastercard/Visa/Access/Eurocard/cheque/postal order or CASH. MAIL ORDER ONLY - no callers without ringing first. Send IRC for full list and details.

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

TEL. 01344-890986 FAX & BBS: 01344-890987

QXL or QPC, of course!). It has got a Mac-compatible serial port running at 57600 baud, but it is RS422 instead of RS232 - anyone knowing a cheap solution to convert RS232 19200 baud to this rate? You can connect two printers at the same time and the printer decides automatically which one is the input. If there is a longer pause on one port and the other port is accepting data, then it switches from one port to the other.

The printer has 5 buttons and a very easy to handle on/off switch. The buttons are multi-functional: LOAD/EJECT, paper select (ordinary, 720dpi, glossy & transparent film). You can also reduce the amount of ink which is used, select condensed print, RESET the printer, clear the heads. There are various other modes possible when you turn the printer on, e.g. hex dump, demo page, calibrate the colours, setup mode etc. - this is very important! You don't need a PC to configure your printer or calibrate the heads.

Compatibility: this is where EPSON always did very well! It uses an extended ESC/P2 command set but still emulates all previous EPSON 9pin, 24pin and 48pin dot matrix or ink printers. This means, you will not have any software problems, you don't need any new drivers (of course, if you want to benefit from the new features which, of course, were not available at the time of 9pin printers) you may need a new driver or configure your existing one.

As the Stylus Color II not only supports the standard fonts in the common sizes (Roman, Sans Serif, Courier, Prestige, Script in 10cpi, 12cpi, 15cpi, condensed etc.) but also supports truly scaleable fonts, it would be a good idea if you get a dedicated driver for your program(s) in order to use the fonts. The ESC/P2 driver for text87, for example, provides the use of these fonts between 8 and 32 point. It also supports nice features like, Outline, Shadow etc.

The printout is quite good on ordinary paper, the ink is extremely black. EPSON has improved the ink even over the ink of the Stylus Color (1), it dries quicker and is darker.

The output is excellent on 720dpi paper, and is the best I've seen on glossy paper (extremely expensive, but looks like a photograph).

There are two ink cartridges: one for black, one for the three colours. I don't want to discuss the advantages/disadvantages of this scheme, it's just the way it is. You can leave the printer unused for 3 or 4 weeks (and probably much longer) and it does not dry in or block your print heads etc. - a major advantage.

The Stylus Color II is fast (up to 400cps high-quality print) and very, very quiet.

All I can say as a conclusion is: it produces excellent output, EPSON printers are very, very compatible so you have next to no driver/software problem and it is therefore highly recommended.

The price is in the usual range for colour printers. If

you want to save money, don't go for the IIs, for many reasons - just don't do it. If you don't need the colour, get the Stylus 820 which is much cheaper (but louder and slower).

I just noticed ('cause I had two orders) that the Color II is not made anymore and you might not be able to get it when you read these lines. The new model is called Stylus 500, looks pretty much the same but has only three buttons instead of 5 (don't think this is an advantage, so if you can get a II, get it!). Otherwise, specifications seem to be the same.

Finally, quite an important consideration for QL users: they seem to be some of the very few people who like to know how to program their printers. The German manual (haven't seen the English) still gives a detailed list of all available printer control codes (ESC/P2) with all their parameters, so that you don't have to buy an expensive programmers' manual. Two or three years ago the EPSON manuals contained only a short summary which they expanded again - well done!

My price for the Stylus 500 is DM 679,- and the ink costs 39,90 (black) and 69,90 (colour). ■

ProWesS SBasic Interface

Wolfgang Lenerz

You may be aware of Prowess, the new window manager. Prowess is still being tested and debugged (by the way, if you have a prerelease version, please do send you comments to Joachim Van Der Auwera - he needs them to make the software even better!). Until recently, the only way to program Prowess was with the 'C' language (shudder). Now you can also use a decent language, i.e. SBasic (NOT SuperBasic). Hence, you need SMSQ (/E) to be able to use the Prowess Sbasic Interface. It will refuse to load on machines other than SMSQ/E machines (there are good reasons for this, explained in the manual). With the interface, you can write SBasic programs that make use of this window manager (here again, since I wrote the SBasic Interface, my comments are probably biased...).

Programming Prowess is actually quite easy. As I say in the manual for the interface, programming Prowess is not complicated, but complex - there are many things to learn at first, but they are not difficult to understand.

The interface should be in the next (pre?)release version of Prowess. It comes with many examples and a good manual (ahem). Any SBasic programmer should be able to use this within minutes! No experience of QPTR or Easyptr is necessary.

Whether this all makes sense, necessarily depends on the success of Prowess. Only the future will tell. ■

FWD Computing

(formerly Mechanical Affinity)

P.O. Box 17

Mexico, IN 46958 USA

317-473-8031 Tuesday thru Saturday only, 6 to 9 P.M

FAX 317-472-0783 7 P.M. thru 11 A.M.

Internet E-Mail address: fdavis@walnut.holl.com

Cash, checks, money orders, or COD. Payable to F. Davis.

Add 10% for foreign currency exchange.

C.O. D. Fee will be added to C.O.D. orders.

Postage for North America is included in price.

We do not accept credit cards; \$10 charge for Electronic Wire Transfers.

PRODUCTS FOR QL

SUPER HERMES - All of the features of HERMES, plus IBM AT compatible Keyboard Interface, high speed RS232 (SER3) from 1200 upto 38400, three low speed RS232 inputs (serial mouse, RTTY, etc), and 1.5K user data storable in EEPROM, and a lot more. This with full cable set for only \$160.

SMSQ/E - The newest and most extensive operating system ever for the QL. Has Pointer Environment, SBASIC, and Tool Kit 2 built in. Three versions available: QL, QXL, Atari-QL Emulator. Price \$147 each.

QPLANE - The Powered Back Plane for the QL is instock. It utilizes a PC Power Supply Unit to help you place your QL motherboard, drive interface, Qubide, etc. inside a PC tower case or full sized desk top case. Add a Super Hermes, Falkenberg Keyboard Interface, or one of our new Di-Ren Keyboard Interfaces plus an IBM style keyboard and it is set to go. Qplane price \$52.

SPECIAL COMBO of QUBIDE and QPLANE - This includes the Qubide IDE/AT hard drive interface and the Qplane for only \$160. Give your QL an update and power as a personal computer!

DI-REN QL KEYBOARD INTERFACES - This will allow you to use a 101 or 102 key AT keyboard (name brand is recommended) with your QL. This is a very small size board and is easily fitted. It translates most keys to QL format and offers keyboard record/playback facilities. The price is \$55.

AMADEUS QL CONTROLLER - Designed to link the Sinclair QL to the Amadeus system. This device connects to the QL's ROM port thus enabling high speed communications. Comes with a through port allowing other devices using this to continue to function. The price is \$70.

AMADEUS AMA-SOUND - Record and play back sounds via your computer. This device employs 12 bit sampling and gives the high quality audio of the ADPCM algorithm. Recorded files may be stored, edited and replayed. Includes all hardware and software. Sample data is in 4 bit packages. All data can be transferred between different types of computers. 3 bit sampling may also be employed. The price for this great innovation is \$84.

QL KEYBOARD MEMBRANES - Replacement membranes for \$18.

QL POWER SUPPLIES - Get a backup or replacement for \$16 while they are still available. These are 110 volt. The supply is limited.

MECHANICAL AFFINITY CLIPART SET - The QL worlds largest clipart set, compressed on ten 720K or five 1.44 disks for only \$38.

PUZZLER - Win QL WRITERS PACK

Here's a challenge for you. The following Superbasic program has 3 deliberate errors in it; can you spot them?

```
10 INPUT text$:REMark The input text
20 :
30 wordcount=0:REMark No words yet
40 checksum=0:REMark Initialise checksum
45 DIM chars%(255):REMark Array for character count
50 FOR char=1 TO LEN(text$)-1:REMark Do whole string but not the Linefeed
60 IF text$(char)=' ':wordcount=wordcount+1:REMark Another word starts soon
70 checksum=checksum+CODE(text$(char)):REMark Add character value to checksum
80 chars%(text$(char))=chars%(CODE(text$(char)))+1
90 END FOR char
95 wordcount=wordcount+2:REMark For first and last word
100 :
110 PRINT 'Number of words: ';wordcount
120 PRINT 'Checksum:          ';checksum
130 FOR char=0 TO 255
140 IF chars%(char):PRINT CHR$(char)!chars%(char)
150 END FOR char
```

The program starts off by requesting a string to be INPUTted which is then loaded into the variable text\$. The string text\$ is then analysed and the results PRINTed to the screen. The analysis gives the number of words counted (words are separated from each other by a single space character), a checksum which is simply the sum of all the character codes of the string, and the number of instances of each character in the string. For example, if you gave the program the string "QL Today is great" then it should tell you that there are 4 words, the checksum is 1517 and then list each character with its instances e.g. spaces 3, Q 1, L 1, a 2, etc.. Note that capital 'T' is different from lower case 't'. The problem is that some of the analysis is incorrect. Study the program and send a printout of your corrected version to Jochen Merz Software. We must have your entry by 15 October 1996. The first correct entry to be pulled out of the hat will receive a copy of Geoff Wicks' QL Writers Pack. Good luck!

SUPER GOLD CARDS from QUANTA

To clear up any confusion that may have arisen, the SUPER GOLD CARDS that are available from Quanta were manufactured especially for Quanta by Miracle Systems who ceased selling them themselves because of their interpretation of the EU's EMC Directive. They are identical except for the colour - the Quanta ones are green whereas the Miracle ones were red. The SUPER GOLD CARD is really easy to fit (it just plugs into the QL's expansion port) and makes programs run a lot faster bringing your QL up to its maximum possible speed. Amongst other benefits it offers a battery backed clock, 4M bytes of RAM, Toolkit II, access to DD/HD/ED disk drives, and a Centronics parallel printer port. To get hold of one contact John Taylor (Quanta Treasurer) at 145 Barrowby Road, Grantham, NG31 8AJ, UK.

QL Returns to Portishead!!

BRISTOL QL WORKSHOP
Sunday 17th November 1996
10am to 4pm

at

SOMERSET HALL
Wyndham Way, Portishead, North Somerset
offering

MAJOR QL TRADERS STANDS
Bring and Buy Sale
Help and Advice Clinic
Talks and Demonstrations
Ample free parking
Refreshments and Bar

For further information, directions and
accommodation availability please contact
MikeAshford Tel. 01934 415416

Last Minute News

This time we reserved the last page for last minute news and it seems we had no difficulties in filling it. Here they are, added a couple of minutes before this issue goes to the printer:

Q-emulator

Daniele Terdina emailed us to tell us about the latest versions of his QL emulator for the Apple Mac. There are two versions, V1.01 is for the 680x0 chip version of the Macs, and costs 65,000 Lire (about £25), while V2.01 is for the PowerPC version of the Macs, and costs 90,000 Lire (about £38). Prices include postage and packing, and a 16 page manual. Payment in Italian Lire currency when ordering from Daniele. Owners of V1.x can upgrade to V2.01 for 60,000 Lire.

SJPD have a demo version of this emulator available on their disk PCPD16.

QL Today hopes to carry a review of this product in the next issue.

Daniele Terdina, Via dei Navali 16/1, 34143 Trieste, Italy.

E-mail: sistest@ictp.trieste.it

SJPD releases...

SJPD have announced what I believe is the first encyclopaedia for the QL, on a 3 disk set in their Classic Books section, called CB136. More details in the next issue of QL Today. SJPD's new disk SJS173 contains a 9 part text file by Bill Cable about using modems and getting online with the QL. SJS172 comprises a disk of utilities by QL Today contributor, Mark Knight.

QLiberator...

Ian Stewart of Liberated Software will be updating QLiberator this autumn. Please send us any bug details or ideas for new facilities, and we'll pass them on to Ian. Or contact Ian at: Liberated Software, 8 Boswell Court, Clifton Road, Kingston Upon Thames, KT2 6PP, England. (Ian may not reply to letters in person at this stage).

Aurora...

Some more details about the Aurora graphics card and replacement QL motherboard from Qubbesoft P/D. We have read about the graphics facilities in earlier issues, but here are some details of the hardware. There will be only 1 joystick

socket, the non-standard serial port sockets currently fitted to British QLs will be replaced by 10 way header connectors, making it easier to add standard 9-pin D connectors with a short lead from the board. Similarly, the keyboard membrane will be replaced by a 20 way header. And the ROM port will be a 34 way header, with some additional lines. The ZX8302 and IPC chips will remain, allowing the serial ports and network to remain the same, and existing add-ons such as Di-Ren's keyboard interface and the SuperHermes from TF Services will be useable. It seems likely that the TV modulator will be dropped since monitors are much cheaper nowadays and few people use the QL with a TV set now. The microdrives will also go. There will be some provision for users who will want to build the Aurora into a Tower Case - the board may be powered by 5volts via Qubbesoft's QPLANE bus extender device, for example.

OPDs for collectors...

Tony Firshman (TF Services) has acquired a few ICL OPD units. These were a development of the original QL made by ICL many years ago. They use the same microdrive cartridges (formatted differently), and the system software bears some resemblance to the QL. Contact TF Services for further details.

EZ-drives from Qubbesoft P/D...

The price of the blank 135MB cartridges is now down from £18.00 to £17.00

7th Italian QL Meeting

Reggio Emilia Italy
c/o Sala Congressi Circostrazione 2
Via Fratelli Cervi 70 - Pieve Modolena

Sunday, 20th of October

Internet WEB page with info about the meeting:
<http://www.geocities.com/SiliconValley/Park/6533>

For info about trains, accomodation etcetera contact:
Davide Santachiara, Via Emilio De Marchi 2,
42100 Reggio Emilia, Italy

Tel. +39 522 300409 (fax/answerphone 10 - 19 CET)
BBS/Fax +39 522 300509 (21 - 3 CET)

The availability of the place will be confirmed in the first days of September. In the very unlikely case that the room will not be available the meeting will be held at the same day in another place.

Beginner's Basics 3

Data Types

Yates, Bristol - Stuart Honeyball

A computer program can be viewed as something that takes in data and manipulates it to derive its output. Data can either be given to the program at the time it is written by the programmer in the form of constants or it can be input from the outside world during run time. It is important to decide what form the data should take prior to writing the program. In Superbasic there are 3 types available: string, integer and floating point.

Strings are used to store a collection of characters. String variable names end in the '\$' character and string constant values are always given within either double or single quotes, e.g.

```
Zed = 'Z' MagazineTitle$ = "QL Today"
```

is a Superbasic statement that will put the value "QL Today" into the variable MagazineTitle\$. The quotes around the value are not stored but are used by the interpreter to indicate where the string starts and finishes. What happens if there is a quotes character actually in the string as in "Oh, blow" he cried."? Simple, you use single quotes instead i.e. "Oh, good" he sighed.'. Likewise, the following statement

```
DoubleQuotes$='''
```

sets the variable to the single character shown.

You can not have both single and double quotes in the same constant value but have to resort to an expression like:

```
SingleAndDoubleQuotes$ = ''' & '''
```

which sets up a 2 character string.

Integer names end in '%'. The values are numeric and lie in the range -32768 to 32767. They are used for things like counting where there is no concept of fractional part e.g. the number of characters in a string:

```
TitleLength%=LEN(MagazineTitle$)
```

will set the integer variable TitleLength% to 8 since this is the number of characters in "QL Today". (Note that the space is a character.)

Floating point numbers are used to represent real numbers like 1.5, PI, etc.. Their names are distinguished by not ending in '\$' or '%'. The values of Superbasic floating point numbers range from about -1.6×10^{616} (-1.6 times 10 to the power of 616) to $+1.6 \times 10^{616}$. The smallest usable

magnitude greater than zero is about 1×10^{-620} . The constant values are written like integers e.g. "96", or have a decimal point e.g. "96.0", or in a condensed scientific notation e.g. "9.6E1". The "9.6" here is referred to as the mantissa and the "1" as the exponent.

All integers can be expressed as floating point values so why have the integer type? Firstly, the program makes more sense and is therefore more readable if quantities that are naturally integer are expressed as such, and secondly, integers take up less memory space than floating points. Integers use just 16 bits (i.e. 2 bytes) whereas floating points require 48 bits comprising 32 bits for the mantissa and 16 bits for the exponent.

You can freely use integers and floating points together as in:

```
Circumference = 2 * PI * Radius
```

The 2 is an integer. The PI constant is the nearest approximation the floating point system can get to the actual irrational number. The two variables are of the floating point type.

Cercion is the process of converting one type to another for them to be combined. In the above example the integer 2 is actually coerced into a floating point type before being used in the multiplication. This is temporary and causes no permanent change but allows the multiplication to be done. Superbasic can coerce any type into any other type: e.g.

```
Num$=123
```

sets the string variable Num\$ to "123"

```
Num=Num$
```

sets the floating point variable Num to 123

```
Num%=Num
```

sets the integer variable Num% to 123.

There is another type called boolean which can have only 1 of 2 values: True or False. It is not native to Superbasic and so is represented by integers 1 and 0 instead. The following program

```
10 Condition%=RND<.5
20 IF Condition%
30 PRINT "Heads"
40 ELSE
50 PRINT "Tails"
60 END IF
```

will get a value from RND and ask whether it is less than .5 putting the result into the integer variable Condition% which is standing in for a boolean. Note that booleans can be represented by integers and therefore by floating points and therefore by strings. We could have used Condition or Condition\$ instead.

Coercion can lead to some strange results. What do you make of the following?

```
10 a$="123.0"  
20 b=a$  
30 c$=b  
40 PRINT a$,b,c$,a$=c$
```

Try it and see. (Don't be confused by the '=' operator; in lines 10 to 30 it is used for assignment whereas in the PRINT statement it is for comparison.) Is coercion useful or just downright dangerous?

There is an even more confusing use of yet another type of coercion. In Superbasic, whenever you introduce a new name, whether it be a variable name like the a\$, b, or c\$ above or a structure name of a loop, procedure or function, it gets added to the Name Table so that future references to it can be made. Now, the device or file name parameters of LOAD, DIR, OPEN, COPY, etc. are strings yet you can get away without putting the quotes around them. So, for example, instead of typing in:

```
LOAD "Flp1_Prog_bas"
```

you need only type:

```
LOAD Flp1_Prog_bas
```

Extra code has been written in Superbasic to take the parameter that looks like a name destined for the Name Table and turn it into a string value. Incidentally it does actually get added to the Name Table as well! Daft!

Also, parameter names of procedures and functions do not need to conform to the right type e.g. a parameter named Para% could hold a string. The following program clears the screen then prints a directory on it. What is the directory of?

```
10 DEFine PROCedure ClsDir(Ram1_)  
20 CLS  
30 DIR Ram1_  
40 END DEFine ClsDir  
50 :  
60 ClsDir "Ram2_"
```

Change line 30 to

```
30 DIR "Ram1_"
```

and try it again.

Always be aware of coercion - it needs to be handled with care. Also, put the quotes around string constants so that there is no confusion. Finally, wherever possible, use the type that naturally fits the bill. ■

The German QL Show



27th of October, Solms, Taunushalle

If you come from the West (A45 from Dortmund), leave the A45 at cross Wetzlar and move via A480 and B277a to the B49 (E44) direction Weilburg/Limburg. You

cross the river "Lahn" 4km after "Oberbiel" when you head for Solms-Burgsolms. When you are in Solms follow the main road direction Braunfels and you will see the Taunushalle lefthand.

If you come from the North go onto the A45 direction Dortmund and leave exit Wetzlar Ost directly onto the B49 (E44), direction Weilburg/Limburg. Pass Wetzlar and after Oberbiel as above.

From south or East as above.

