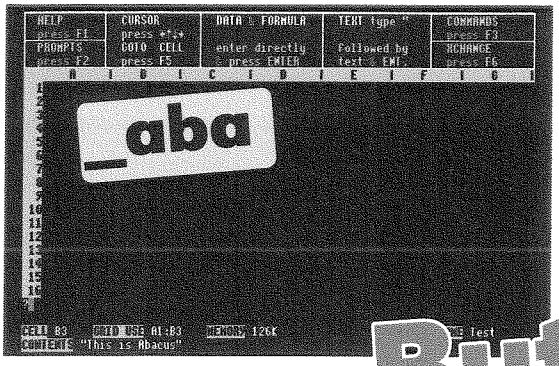
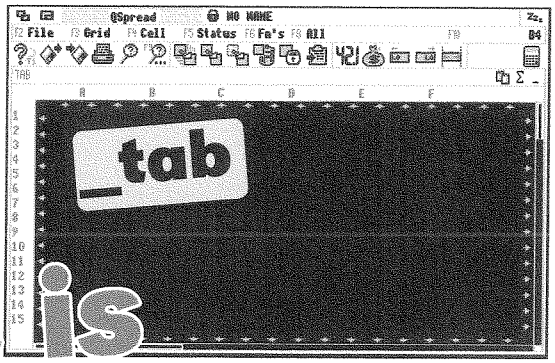
QLtoday

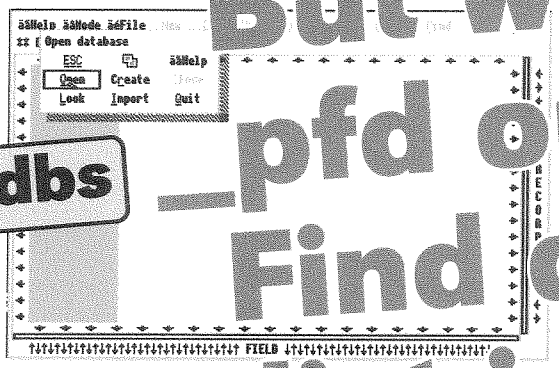
Volume 6
Issue 3
Sept./Oktober
2001

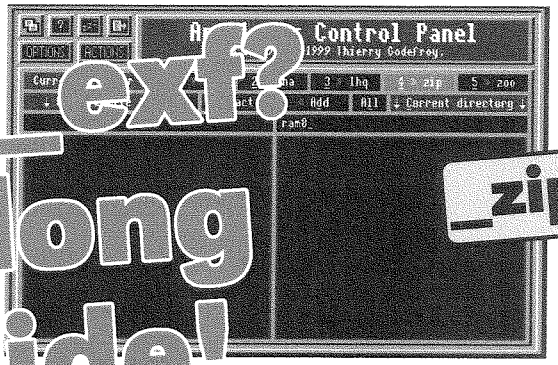
ISSN 1432-5454

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

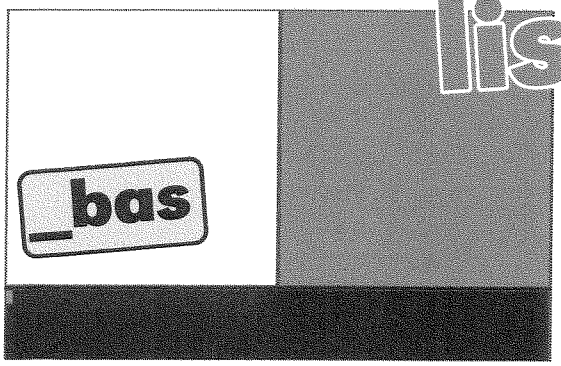


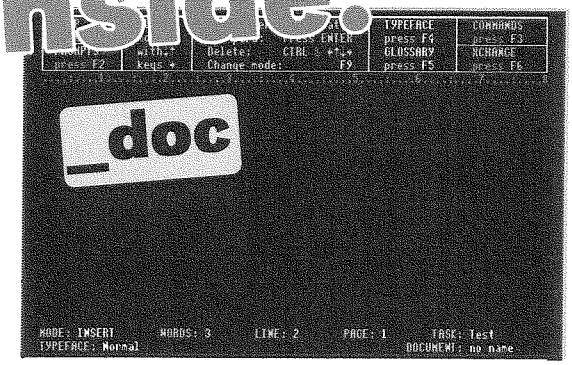






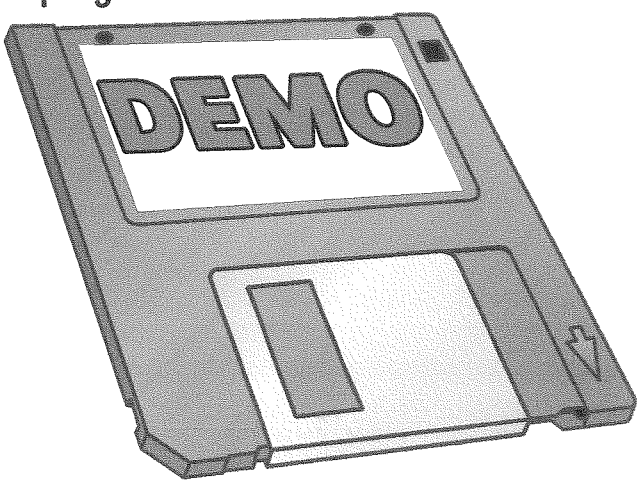
But what is _pfd or _exf? Find a long list inside!





New QL website with 80 MegaBytes of QL programs! Details inside!

Many demo versions of commerical programs on this issues' cover disk!



Contents

Advertisers

in alphabetical order

QL Today

ISSN 1432-5454

German office & Publisher:

Jochen Merz Software
Im stillen Winkel 12
47169 Duisburg
Germany

Tel. +49 203 502011
Fax +49 203 502012
Box1 +49 203 502013
Box2 +49 203 502014
Mobile +49 1701 222277
email: JMerz@J-M-S.com
email: jochen@QLToday.com

English office:

Q Branch
20 Locks Hill
Portslade
BN41 2LB
United Kingdom

Tel. +44 1273 386030
Mobile +44 7836 745501
Fax +44 1273 381577
email: qbranch@qbranch.demon.co.uk

Editor:

Dilwyn Jones
41 Bro Emrys
Tal-Y-Bont, Bangor
Gwynedd
United Kingdom LL57 3YT

Tel. +44 1248 354023
Fax +44 1248 354023
email: dilwyn.jones@dj.softnet.co.uk
email: dilwyn@QLToday.com

Co-Editor:

Bruce Nicholls
38 Derham Gardens
Upminster
Essex RM14 3HA
United Kingdom

Tel +44 1708 510764
Fax +44 870 0568755
email: qltoday@q-v-d.demon.co.uk
email: bruce@QLToday.com

Please send comments and articles to
editor@QLToday.com

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

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

Article and Advertising deadlines are as follows:

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

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

Editorial

The big news this month is the announcement by a group of Italian QLers including Giorgio Garibello (who set up the Quantum Ring group of QL-related websites) of a website dedicated to freely downloadable QL software. Those connected to the internet will be able to download about 80MB of freeware, shareware and demo versions of QL programs.

The internet is already rich with QL related websites and software to download, but this nonprofit making site (available in both Italian and English language versions) will give the QL a one stop source of free software rather like the repositories available to users of other computers. In addition to free downloads, users and authors will be able to offer software for upload, giving QL software writers a one-stop facility to offer their programs to the QL using public. This site, in conjunction with the established QL PD libraries, should encourage and promote the wealth of free QL software available (Jon Dent – another hint to complete SOQL!). The owners of the site are to be congratulated too on their clear and firm stance against including illegal and offensive material (including the pornographic and other files which seem ever prevalent on most of the internet but thankfully not a problem so far on the QL scene).

Partly inspired by the above announcement, and partly due to encouragement from a few other QLers, I have now decided to set up a PD software disk copying service. It is increasingly becoming easier to obtain QL software free from the internet, but those QLers without internet access still need a source of PD software on disk, so I have decided to start my own service – see my announcement in the news pages, and please note that it is my own venture, unconnected with QL Today or JMS.

With Peter Graf's replies to the Q40/Q60/emulators debate in this issue we now hope to draw a close on this saga. Opinions varied, debate was open and frank and at times heated, but at least it's been given a good airing and hopefully cleared the air a bit. Peter and Claus Graf can be confident that many readers back them in their efforts to produce quality QL hardware and software. It seems apparent that most people believe that Q40, Q60 and Goldfire are the (essential) future of native QL hardware, while the emulators have a role to play as well.

With the awful events in the USA in September still in our minds, we at QL Today (and I'm sure we speak for the whole QL community in Europe) would like to express our deepest, if somewhat belated, condolences to our friends in the USA. You will always be in our thoughts.

The DEMO VERSIONS Cover Disk

With Volume 6 Issue 3 of QL Today, we bring you a disk with demonstration versions of seven commercially available QL software packages. Most of these programs are fully working versions of these programs, but with some restrictions on their use (e.g. saving or printing may be disabled, or the programs may only work with restricted file sizes or numbers of files). If you like these demo versions, please consider buying the full versions of the programs from the QL software traders. The more they sell, the more they are likely to produce!

Many of the programs require pointer environment (built into SMSQE of course, and the files ptr_gen, wman and hot_rext for QDOS), Menu Extension (MENU_REXT supplied with many commercial QL programs) and Toolkit 2 which is built into most modern QL systems.

The seven demo programs are all supplied in zipped file format. This means that they have been packed and compressed using a program called ZIP to allow more programs to be supplied on this disk. You need a copy of the Unzip program to decode these files, so (thoughtfully!) I have provided a copy of that program along with a short SuperBASIC program called BOOT which you should LRUN to start decoding the seven program zip files.

You will need a separate blank, formatted disk for all 7 programs (or as many of them as you plan to try out). Put the cover disk (or preferably a backup copy of it) in FLP1_ and the blank, formatted disk in FLP2_. If you have only a single floppy drive system as I do, you will need to unzip the files to a ramdisk or hard disk first, then transfer them back to a floppy disk.

The program asks you to select one of the seven programs by entering the number shown on screen (type in a number from 1 to 7 and press ENTER). Then it asks you to enter the name of the drive holding the zipped file (e.g. FLP1_), then the name of the drive to unzip it to (e.g. RAM1_ or FLP2_) and finally the name of the drive holding the UNZIP program itself (e.g. FLP1_). Unzip will load and attempt to decompress the files. This may take some time, depending on the speed of your QL system.

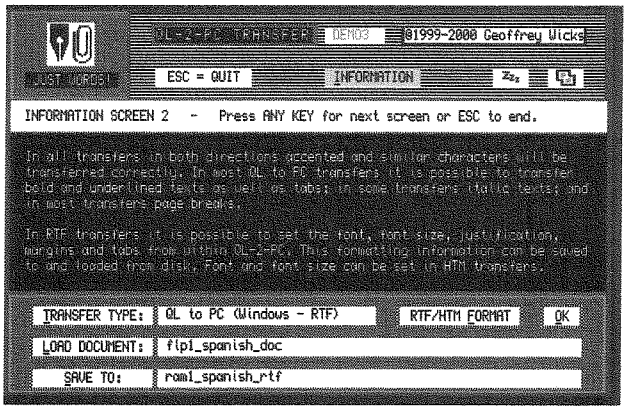
QL Unzip is maintained by Jonathan Hudson and full documentation, up to date file sets etc can be obtained from his website on **www.daria.co.uk**

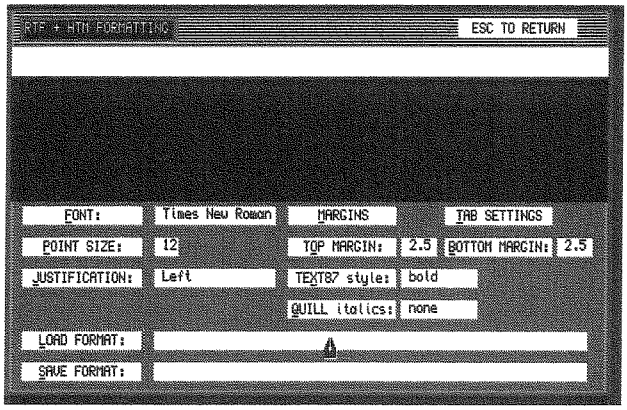
QL-2-PC DEMO VERSION

Author: Geoff Wicks, of Just Words
Price: £10.00 from Just Words or QBranch
Documentation: QL2PCman_doc (Quill doc file which includes tutorial)
Requirements: Expanded memory, pointer environment and Toolkit 2

QL-2-PC is a program for transferring word processing files between QL and PC word processors and PC HTML editors. The QL files to be transferred can be in Quill, Text 87, Perfection or ASCII format. The PC files can only be in ASCII format. The program also contains experimental routines for extracting the text message from e-mail files, and for tidying up text that is OCR read, or has been downloaded from the internet.

The demo version is a fully working version of the program with just one restriction. You can load a document of any length into the demo version, but it will only transfer the first 3,000 bytes, about 500 to 600 words. The transfer can create a pseudo WordPerfect 4.2 file for DOS, or Rich Text Format (RTF) and HyperText Markup Language (HTML) for Windows.

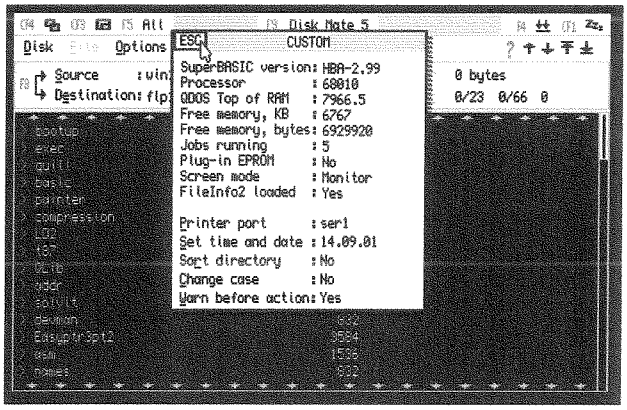


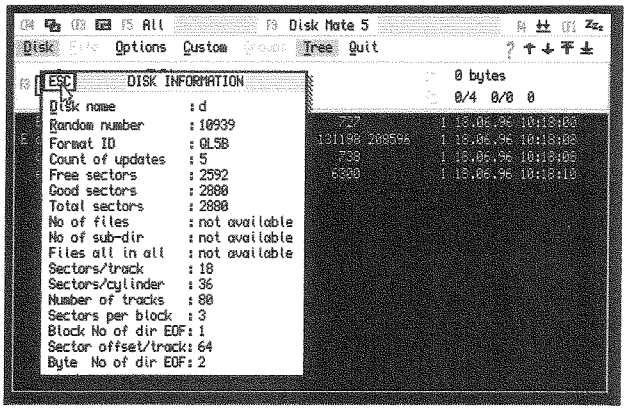


DISKMATE 5 DEMO VERSION

Author: Pal Monstad of PM Data
Price: £16.50 from QBranch
Documentation: dm5_demo_txt (plain text file)
Requirements: Pointer environment, Menu extension and Toolkit 2

Diskmate 5 is a simple to use disk and file handling program, geared to handling directories, and giving you disk and system information. The demo version is limited to letting you work with no more than 10 files. To start the program you need to LRESPR the extensions_cde file as well as ensuring that the pointer environment and menu extensions are available., then execute the dm5_demo_obj program. Place a disk (e.g. the DiskMate 5 demo disk!) in a drive and explore the program's commands and capabilities. It's quite easy and intuitive to learn!





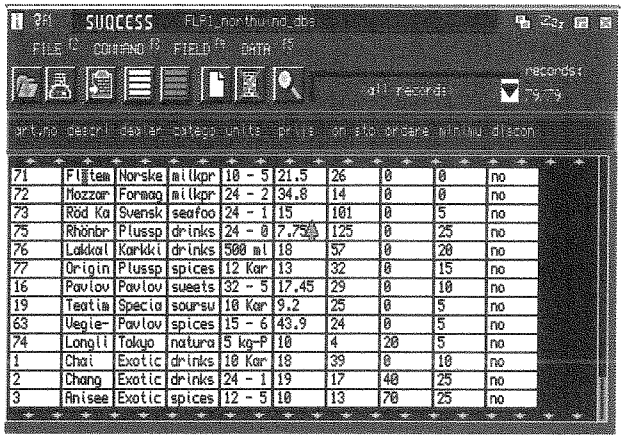
SUQCESS DEMO VERSION

Author: Wolfgang Uhlig
Price: £25.00 from QBranch or JMS
Documentation: READMEFIRST_txt (plain text file)
Requirements: DBAS system (supplied), Pointer environment, Menu Extension v7.50 or later.

This is a demonstration version of a database application by Wolfgang Uhlig, based on the DBAS database engine system from D.W.Howells. It has the limitation that changes are not written out to the database, but most other facilities in it work. Before attempting to execute the SuqcessTrial_obj program, you should LRESPR the DBAS_SYS file and configure the SuqcessTrial_obj program with a copy of the Config or MenuConfig programs, to tell the program the directory where it will find its SUQCESS_INF file. This is also where the help file SUQhelp_dat should be located in order to allow the program to give online help.

An example database file called "northwind_dbs" is supplied to let you play with the program to find its capabilities.

For more information about Suqcess, see Tim Swenson's review in Volume 5 Issue 1 of QL Today.



QROUTE DEMO VERSION

Author: Brian Henderson (ST version), Rich Mellor (QL version)
Price: £25.00 from QBranch or RWAP Software
Documentation: README_txt, UPDATES_txt, PDmanual_txt (plain text)
Requirements: At least 896K memory, Toolkit 2, Pointer environment, Menu extensions

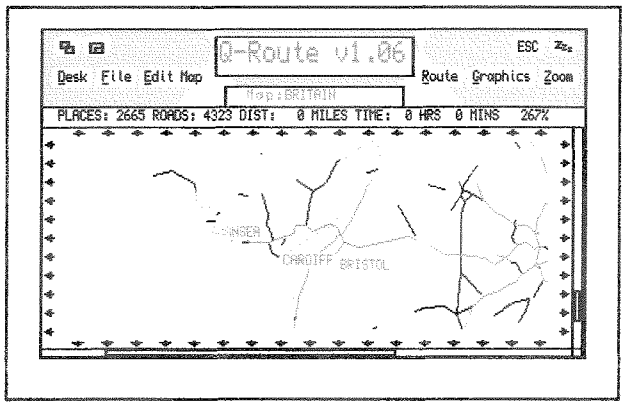
Important: after unzipping the program, you MUST rename four files to have full stop filename separators instead of QL style underscores (this is for compatibility with files from the Atari version). The files concerned are: BRITAIN_SEA, BRITAIN_MAP, FINDER21_INF and MAIN_MEN and they should be renamed to BRITAIN.SEA, BRITAIN.MAP, FINDER21.INF and MAIN.MEN respectively.

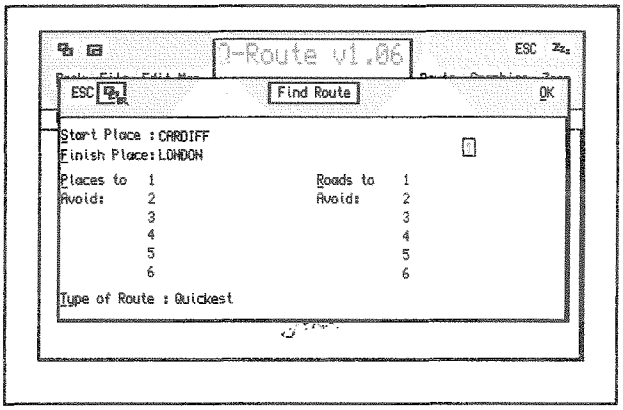
QRoute is a route finder program based on Route Finder v2.2, an Atari ST program by Brian Henderson, converted to the QL by Rich Mellor of RWAP Software.

At its simplest level, this sort of program will tell you how to get from A to B, allowing you to specify quickest route or shortest route. On a more advanced level, you can specify places to avoid (don't tempt me!) or places to go by, with the route details shown graphically as a map or as a text table.

This demo version cannot save maps, print routes or find routes of more than 100 miles, and there is a delay built into the Find Route routine.

A single map of Britain is supplied with this version.





SPY DEMO VERSION

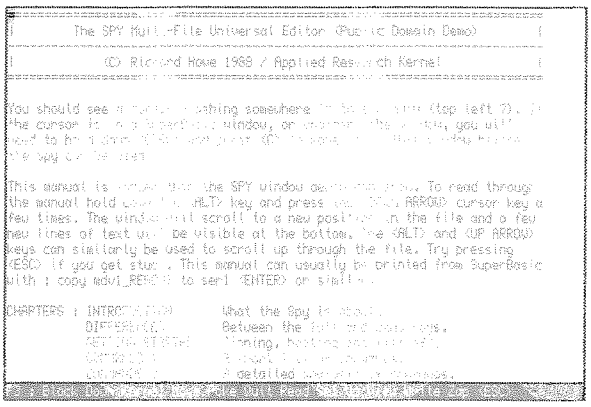
Author: Richard Howe
Price: Standard version £15.00, Master Spy £30.00 from QBranch
Documentation: README (plain text file)
Requirements: None specified

An updated version of an old favourite from ARK. Spy is a text editor, compact, fast and simple to use. It is not pointer driven but very useful nonetheless. The demo comes in two versions, one for use with TV display mode, the other for monitor mode.

This demo version is basically the full version of Spy but without the File and Block 'Write' commands (in other words, you can't save anything). Once in the editor, press F1 for on-screen help. First, read the 'README' text file for instructions.

Spy is primarily a programmer's editor and does not pretend to be a word processor, although like most editors it can be used to type simple letters and so on. Spy can load text or binary files, can multitask with other programs and can read and stack as many files as your hardware can cope with.

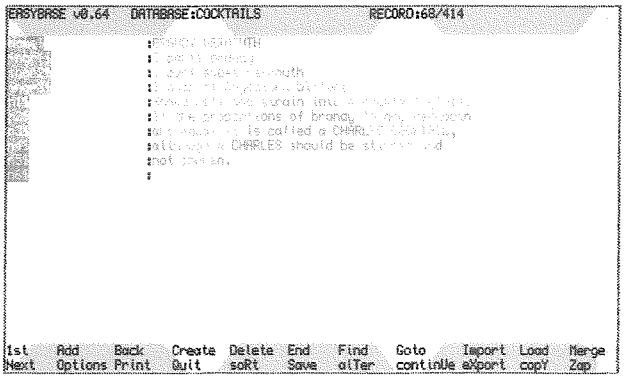
Note how scrolling works - when you first use the program, you may be a little confused when you try to scroll the cursor off the visible screen and only one line changes. This is for faster scrolling - let go of the cursor keys and the whole screen of text is then redrawn



EASYBASE DATABASE PROGRAM DEMO

Author: Dilwyn Jones
Price: £15.00 from Q-Celt Computing in Ireland
Documentation: EasyBas1_doc, EasyBas2_doc, EasyBas3_doc
Requirements: Expanded memory, Menu extension (Menu_rext only required if copying entries to Scrap)

A simple to use non-pointer driven database system with built in printing facilities for labelling etc. The Save, Export and Print commands are disabled in this demo version, but everything else works and a full manual and example databases are supplied for you to experiment with the program. The full version of EasyBase offers comprehensive printing, importing, exporting, password protection and many other features, as well as the usual database facilities. It is designed to be simple to use, with just the right features and facilities and nothing more. Handles up to 100 fields (or lines) of information per record, with files based in memory for speed and less likelihood of disasters with open disk based files! Specify your own database structure. Although the program is not pointer driven it can use higher resolution screens by changing its own window height to let you view more of databases with large numbers of fields.



CUESHELL DEMO

Author: Albin Hessler
Price: £20.00 from QBranch
Documentation: README (plain text file)
Requirements: Toolkit 2, pointer environment, menu extension

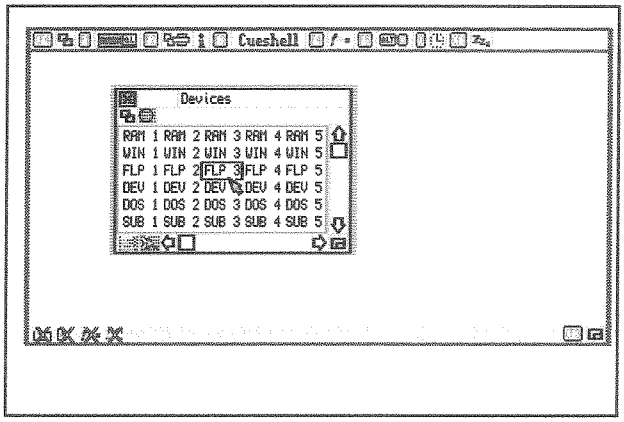
A comprehensive and very impressive pointer driven file handling, job handling and system setting utility. CueShell is mostly icon-controlled with the row of icons across the top of the program's display.

To access the devices list, right click (DO) on the name of the drive in the deviuce names ("DEVICES") box in the main window. Another window will pop up with a list of files in it. Click on the icon next to the filename (the icon identifies the type of the file) e.g. the README file on the CueShell demo disk, then click on the magnifying glass symbol under it to view the file. Click on the X symbol to exit the files list.

The program has a limited form of drag and drop capability for copying whole disks or directories. Left click on a drive or directory name to indicate where to copy files from, then right click or DO on a ramdisk name for example to indicate the destination.

Incidentally, it's not very obvious - to quit from the program, click on the 'CueShell' program title at the top of the screen, then select the Quit command.

This demo version cannot copy and delete files, but all other facilities work. There is so much in this program that it will take you a while to figure it all out given the lack of instructions, but believe me CueShell is well worth the effort.



NEWS

NEW PD Software Library

Your friendly editor has decided to set up his very own PD software library service for the QL (albeit with a bit of prompting from certain other notables on the QL scene).

While preparing the QL Emulators CD I acquired hundreds of items of freeware, shareware, PD, charityware, cardware and demo versions of commercial QL software, so what with the announcement by Giorgio Garabello of a massive repository of free QL software available via the web (see his news item elsewhere in this issue) it occurred to me that while those of us fortunate enough to have internet access can now obtain QL software free from the web, those less fortunate do not have such ready access. So the Dilwyn Jones PD Software Library is born. Please note that this is unconnected with QL Today or JMS.

Disks can be copied for you at a charge of £1.00 per disk if I supply the disk, or 75pence if you supply the disk. Catalogues are available either on disk (send me a blank formatted disk), or by email on request, or download it from my website **www.soft.net.uk/dj/index.html**

Library disks will be supplied on HD disks unless DD disks are specifically requested. DD disks are harder to obtain at the moment, so I'll keep those for QLers unable to use HD disks!

I'll also be able to supply some of the QL CD-ROMs I prepared for Q-Celt Computing - namely the QL Emulators CD, QL Religion CD, QL Line Design Clipart CD plus Gerhard Plavec's QL PD-CD.

Dilwyn Jones,
41 Bro Emrys,
Tal-y-bont,
Bangor,
Gwynedd, LL57 3YT,
U.K.
Email: **dilwyn.jones@dj.softnet.co.uk**
Website: **http://www.soft.net.uk/dj/index.html**

Q-CELT Computing News

With the release of the latest version of the uQLx emulator, the QL Emulators CD has now reached version 1.15.

Release 1.5 of the DJ Bargain Bundle CD is now out. Version 1.1 updated the EasyBase database system to version 0.65, version 1.2 added the Doc2Html program for creating simple web pages, while version 1.3 added several programs such as Dictionary, Today, Translation and a couple of gambling programs, then version 1.4 added some new pointer driven utility programs like Metric Conversions, Unzip Librarian and the SystemSet utility. Finally version 1.5 added a simple temperature unit conversion utility. Upgrades are available from any version to the current 1.5 as usual.

The Spectrum Emulator CD by Darren Branagh, which contains the ZeXcel Spectrum Emulator for the QL by Ergon Development (now Freeware) and several thousand Spectrum games, Demos and Utilities, is now out and fully available - it will be launched at the Byfleet Show on the 23rd Of September. Also Phoebus Dokos's adventure game CD containing hundreds of InfoCom ZIP text adventure games for use with the QL version of the ZIP interpreter (included on the CD too) is now ready. This has a very elaborate full colour CD sleeve designed by Phoebus (he is a graphic artist by trade) It will also include a 37Mb (yes, 37Mb!) Animation sequence at the start of BOOT on the CD (worth seeing!) The first release will also be for sale at Byfleet for the special introductory price of just £8, with 50% of the profits going to childrens charities.

Consideration is being given to producing a ZX81 Emulator CD as well (would you like to turn your QL into a Spectrum or ZX81???) This is in the production stages, and work is being done on gleening several hundred ZX81 programs from the Internet, so expect this to surface in the next couple of months.

Q-Celt are also looking into producing a large PD QL Encyclopedia on CD. There are several Public domain PC encyclopedias out there, which would need converting to QL format, so if you would be interested in one of these please contact us and we will look into producing it. We would also welcome any ideas for future CD's and/or alternative products from our customers. If you would like to see us produce something, then please contact us.

News from RWAP SOFTWARE

Rich Mellor

We are looking to expand our range of software and would welcome any submissions for possible commercial publication, or even ideas for future products.

We have one or two ideas in mind, in particular a new word game (pointer driven) and converting Q-Route to compile under the latest version of Turbo (using its PTR interface).

If anyone is interested in helping out with the programming in return for a few royalties and the ever grateful thanks of QL users, please let us know.

Replies/submissions should be sent by snail-mail (sorry, no real email access) to:

RWAP Software
7 Common Road
Kinsley
Pontefract
West Yorkshire
WF9 5JR
TEL: +44 (0) 1977 614299

NESQLUG Articles

Al Boehm

In the July Virtual Meeting conducted by email, NESQLUG voted to upload all articles as soon as they were edited onto their website:

www.geocities.com/nesqlug1/

These articles will be available in an open section where anyone, member or not, can browse them. Members will be notified by email of the title when a new article is uploaded. A quarterly journal will be published containing all the technical articles for members who want a paper copy or do not have web access. Also Quanta has the option of publishing those articles with wider interest.

News from GEORGE GWILT

George Gwilt has written to let me know of updates to his range of QL software. Most of the programs are available from

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

GWASS v1.14

GWASS v1.14 allows filenames used in batch mode to contain all characters except forward slash "/".

MOVING PROGRAMS

New versions of the following programs have been produced so that they can now be moved around the screen when F9 (SHIFT F4) is pressed.

1. **NET_PEEK v3.31**
   Displays ram
2. **DISP3 v3.9**
   Displays up to 9 directories
3. **GWDISS v2.23**
   Disassembles all 68xxx instructions
4. **SYM1_BIN v2.9**
   Decodes the symbol file produced by GWASS

TurboPTR
A new version of TurboPTR caters for buttons. The procedure Do_Sleep has been added to TPTR_BAS v3.5 which automatically sets a program to a button and returns it when the button is hit. The button will be a valid entry in the button frame if that exists, otherwise it will appear at the top left of the screen.
The extensions loaded by TPTR v2.8 now include the functions:
`MODE%` returns the mode (4, 8, 33 etc)
`PNAME$` returns the current job name
`BFRAME` returns a position in the button frame
and the procedure:
`BFREE` frees the position in the button frame.

SETF v3.2, which enables windows to be set up:
a) produces a second repeated section for the first main window. (This is needed to produce the button.)
b) can now be aborted at any stage by pressing ESC.
Part of the TurboPTR package is a sleep sprite. This now contains a mode 8 version.

SEEWIN1 v2.2 displays all repeated sections of all windows. Thus it will display the button window produced by SETF.

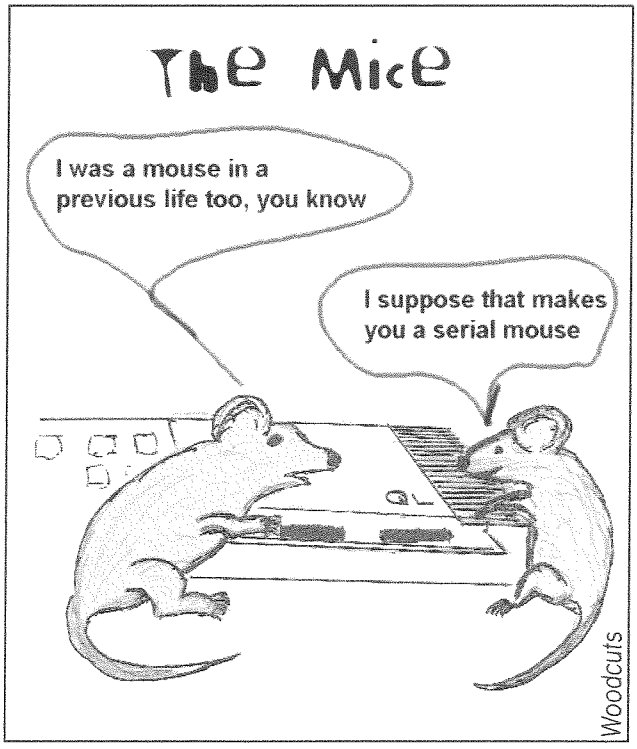
ALTER v 3.1 now caters for repeated sections of main windows.

DIGIVIEW
DIGIVIEW is a program to view pictures from and to manipulate Kodak digital cameras. At the moment it caters for DC200, DC210 and DC215. It is in the course of testing and being extended to other marks of Kodak digital camera. The program is based on Simon Goodwin's DIGICAM_BAS. The testing version number is now 1.3.

TURBO V4.10
Version 4 Release 10 of Turbo Compiler is now available from the Other Software Page on the Dilwyn Jones website. George Gwilt lists the following changes to previous versions:
1. The BAUD command can now take more than 1 parameter, as required to work with the independent baud rates allowed by SMSQ/E.
2. CONTINUE is now accepted. Previously it was rejected with the cryptic error message "End Of Statement Expected"
3. END_WHEN is now accepted as an alternative to END WHEN (useful for QLers using pre-JS QLs)
4. Trailing windows have been eliminated
The initial release managed to include the wrong version in its config block, so make sure you get the fixed version which includes a program to patch the details.
Further details in the UPDATE_TEXT file with the compiler on my website.
http://www.soft.net.uk/dj/software/other/other.html

J-M-S News

And once again I would like to invite everybody to join the QL News List. Here you will be informed about News and Updates, QL Shows etc. You won't receive masses of Email, an average of 1 to 3 per month.
If you would like to join, send an email to **smsq@j-m-s.com**, subject "subscribe" and a text like "yes, I'd like to joing the QL News list".
It is maintained manually by me.
Please note: if you change your email address and news bounce, I have to remove the address from the list. I can't trace changed addresses, so please notify!



Cartoon

Different viewpoints

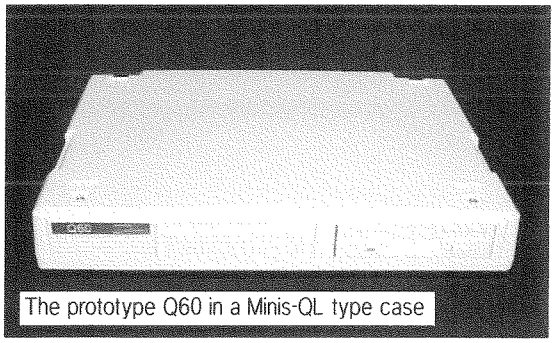
Peter Graf

In QL Today I found my latest articles accompanied by highlighted frames telling the readers in bold printed text, that my words were harsh, that I had attacked others, and so on. That would not be too much of a problem, if the author of these accusations had published his negative judgements about me as his personal view, telling his own name. But this way it is a official for QL Today. It shows that my contributions are merely tolerated, and officially condemned, while articles with insulting words against QL hardware users are welcome. This combination is not acceptable, and I consider not to write any further contributions for QL Today. Remember, I never used words like "slap in the face", "arrogant" and "unbelievably stupid" for others. I never claimed to have the only answer for the future of the QL. I never attacked other's favorite systems with wrong statements. And I didn't start the argument. The authors responsible for all that, could freely do so, and I did not find such official comments by QL Today for their articles. Shouldn't it be up to the readers to decide, whose words were harsh, and who has attacked?

In the given situation, I think it is better not to answer the articles of the QPC dealers, who are at the same time QL Today officers. I am thankful that they contribute so much work to create a QL magazine. Therefore I think their time is better spend with that work, instead of answering all the points where I must disagree.

Wolfgang Uhlig has nothing to offer, but to call me a member of a "clan" where one writes "ridiculous nonsense".

So I start with Jim Hunkins: Nobody said or thought that *all* major software development was done on native hardware. So we don't disagree. If an emulator helps you, this is, of course, fine. Portable computing is no natural advantage of emulators, though. The low power consumption of a Q60 shows, that native hardware would be very well suited for a portable QL compatible computer. It could be faster and consume less power than a PC laptop with emulator. An interesting project, but not affordable and impossible to produce, if there are only a handful of people who want it, and the rest prefer emulators. The less folks go for QL hardware, the less we can do.



The prototype Q60 in a Minis-QL type case

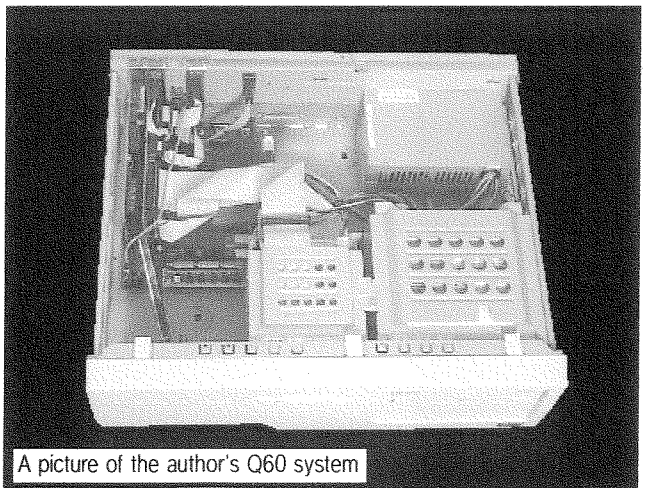
Steve Hall: User perception is indeed what matters. But general statements like "system A is appreciably faster than system B" can not be based on the perception of one single user (who runs system B in a slowed-down mode). If we want to compare, we must compare based on facts - or better not compare at all.

You say that QPC2 on an up-to-date PC gives you "hardware expansion only limited by what is currently available", while you call the Q40 "limited". Here you are completely wrong. If PC hardware extensions run under Windows, that does not mean they run under QPC. Neither QPC nor SMSQ support all those PC stuff. You would need hundreds of new drivers. And even if you want to write all those drivers yourself, you can hardly do that under QPC2, because you don't have direct control over the hardware from within QPC2. You would need to change QPC2 itself, but it is not open-source-software, so you can hardly do.

If I look at the current situation, I do not see that Q40 and Q60 have disadvantages concerning supported extensions under SMSQ/E. I see the opposite. For example, the CDROM drivers don't run under QPC, but on Q40 and Q60. The sampled sound drivers don't run under QPC, but on Q40 and Q60. I don't write this to criticize QPC, just to correct the negative perspective toward the Q40.

You call QPC+Windows+PC "a fast QL". I have my doubts if I would call a QL that needs minutes to boot SMSQ/E "fast". However, there is a difference between "fast" and "faster". If somebody wants to be "faster" he will probably go for a Q40 or Q60. Even if you sell your car to afford the most expensive Windows PC available today, you have no chance of reaching the Q60's computing speed under QPC. There are advantages of QL hardware. Not only for purists and "Wintelphobes", as you wrote, but also for realistic QL users and developers.

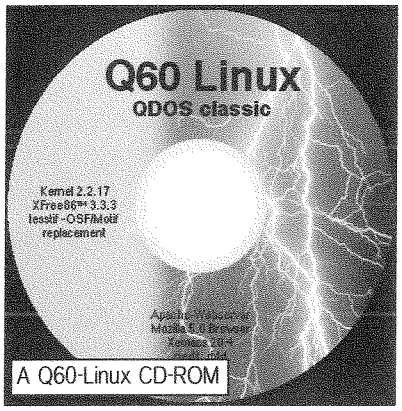
Marcel Kilgus: Obviously nobody said, that QPC users were not part of the QL community. To deny that a Windows PC is a QL (just because it can run an emulator), offends nobody. A PC can emulate a gameboy as well as a QL. Do the Gameboy users have to apologize, if they think a PC is not a gameboy? Like Claus has to apologize, if he thinks a PC is not a QL?
You say that a Windows PC with QPC is a QL system. So it is surprising that you claim QPC does not emulate "any hardware on the planet", i.e. not even the QL! Maybe lets look at a simple example. What happens, if I use the assembler instruction "MOVE.W #$FFFF, $20000" under QPC? I thought, it first emulates the 68000 processor hardware in order to execute the instruction, it then emulates the QL video hardware in order to translate the data into PC screen data, it then sends a call to the Windows operating system, which then calls a driver software. (If I had my driver working correctly) the PC graphics card then sends a small white line to the screen. I thought it worked this way, and would call it emulation of the missing hardware. But according to what you say, I must be wrong. QPC must really be a miracle beyond my imagination.


A picture of the author's Q60 system

You say that a Q40 is not hardware compatible to the QL. So what happens, if I use "MOVE.W #$FFFF, $20000" on a Q40 or Q60? The CPU executes the instruction directly, and the video hardware directly shows a small white line on the screen. The same happens on a QL or GC or SGC system. Like the Q40, GC and SGC also have adapted operating systems and new features, so "hardware compatible" can never mean "exactly equal". But compared to Ataris and Amigas, let alone PC's, shouldn't we be allowed to call QL style hardware "hardware compatible"?
It seems like a Windows PC that can run an emulator must be called "a QL" (otherwise one has to apologize), while QL hardware is not even allowed to be called "hardware compatible to the QL". Sorry, this becomes too complicated for me. I thought computers were computers and emulators were emulators.
I can not confirm your benchmark figures, latest SMSQ/E versions. Test909-graphics gives me factor 2.5 for Q40, not for QPC. Test909-Math gives advantage for QPC indeed. Little wonder, since you added Intel FPU code to SMSQ/E, but not 680x0 FPU code. By the way, instead of only publishing two sub-results of this benchmark, how about the overall result of Test909: I got factor 4 for Q40 there, compared to a 1200 MHz Athlon/Rage-Pro. Forgive me that I switched the copyback cache on, for both targets, and not only for the PC.
Regarding audio input: Fine that my remarks "made your day", but I can not find anything ridiculous about using a soundcard for LINE/MIC-In and the IDE bus for CD-In. Nobody wrote SMSQ/E software for this, but software was not the issue. The Q60 mainboard was said to be the problem, and a new mainboard was proposed. Who would complain that a PC has no sound input, just because it uses extension cards for it? By the way, I do not generally reject LINE/MIC-In as an idea for a new mainboard. It might simplify writing SMSQ/E software. I just don't accept, that the existing Q60 *hardware* had no means to allow audio-input.

Wolfgang Lenerz: Fully agreed. By the way, it was Claus, not me, who adressed the problem that moving to emulators can lead away from the origininal OS. You point out that there is also software development under emulators. I can only answer, that in my opinion both aspects exist at the same time. And you are right, many people have a PC and Windows because they want email, internet surfing, playing games, and so on. But if the only answer to our lack of QL software is to use Windows software instead, we will never fill the worst gaps of QL software! We need as much motivation for native software development as possible, and in my opinion there can be no better motivation than native computers.
Look at other small computer scenes. Many of them have internet access, for example. Recently I talked to a ZX Spectrum magazine maker, and was surprised to hear that most of those folks are users of native

hardware and they have internet access. It must be possible for us as well!



David Denham: I agree to all of what you say. Except your idea, the Q40 might lead to Linux like QPC2 to Windows. The Q40 is a QL compatible machine especially designed for QDOS and SMSQ/E and it directly runs both operating systems. Nobody needs to deal with Linux on a Q40 at all. If using a modern QL is all you want, just switch on and there you are. Opposed to the PC which primarily runs Windows, and QPC is just a program, depending on (and directly related to) Windows.
The fact that you can not afford a Q40, makes me feel a bit depressed, because Q40 and Q60 were designed for dedicated QL users like you! I did my very best to make the Q40 as affordable as possible, but I also felt the need for so many improvements. If I had to throw away some of the features, in order to make it cheaper, I would not know which! The best thing that can make QL hardware cheaper, is to produce more boards.

Conclusion:

Imagine we all sit behind Windows screens. Is this really the future of the QL? The kind of "QL community" we have in mind? I don't think so. But in order to keep interesting QL hardware alive, we desperately need more support for hardware. A software emulator can easily be given away for free, if the author wants. But each single piece of hardware needs to be produced, which is (besides being a lot of work) only feasible and affordable, if we reach higher quantities, with the help of fair public attention.

***The pictures accompanying this article are from the Q40 and Q60 website, which readers can access at* www.q40.de**
A limited supply of Q40 and Q60 systems are available direct from Peter Graf - contact him by email on* pgraf@q40.de *to discuss your requirements. Despite the ill feeling which has clearly been caused, QL Today would very much like to wish the Claus brothers well with the Q40 and Q60. Hopefully, this debate is now closed.

QL Filename Extensions

by Dilwyn Jones and Jochen Merz

In response to a request received from a reader, the following list of filename extensions has been compiled. No doubt there are others and we will gladly publish any additions to this list.
The use of filename extensions is not as 'formal' or rigid in QDOS/SMSQ as in some other operating system. We use these for convenience, in order to recognise certain file types, e.g. files with filenames ending in _doc are generally document files, usually Quill files. But this is not the end of the story. Many QL programs will recognise files with different extensions if the files themselves are in the correct format. For example, if you save a file from Quill with the filename FLP1_EXAMPLE_doc or FLP1_EXAMPLE_PER, it will manage to reload either one. It is generally wise to stick with the 'standard' extensions for a particular program's files.
One example of the use of these extensions is the FileInfo program used in conjunction with QPAC2. Properly installed and with the correct definitions entered, you can hit a filename in QPAC2 files menu, then use the EXEC command to execute that file. But since a text file, for example, is not an executable program, FileInfo ensures that instead of trying to execute the text file, what gets executed is the program associated with the filename extension. So instead of executing EXAMPLE_TXT as a text file, you can execute your favourite text file editor and tell it to load that file automatically. FileInfo is available from Thierry Godefroy's website and from the usual sources of free QL software.

_aba	Abacus spreadsheet
_a	A C compiler file
_a87	Text 87 font list files
_alt	Altkey definition file
_app	Application menu or (in Easyptr) an Appendix Manager file
_arc	ARC compressed file
_asm	Assembler files
_bak	Backup file
_bas	SuperBASIC or SBASIC BASIC program
_bin	Binary file, machine code output from assemblers, or BASIC extensions
_blb	Pointer environment Blob file
_bmp	Bitmapped graphics files
_boot	Usually a BASIC or command file, used

for startup purposes

_box List of MIDI files to be played in MidiPlayer 3

_byt Bytes file, or basic extension or pure code file

_c87 Text 87 configuration file

_c91 Text 87 configuration file

_cde Code files, e.g. machine code or extensions files

_cdk Cuedark screen saver file

_cfg A configuration file, used by several programs

_cmd A command file, used by several programs

_code as _cde

_com A compressed file - usually graphics

_crf Assembler cross reference files

_ctl Control files for JMS Thing And EPROM Manager

_cue Cuedark and Cueshell files

_cut Page Designer 2 and 3

_dbf Archive database

_def Definition files used as configuration files for example.

_dic Dictionary files used by Solvit Plus 2 and 3

_dmo Demonstration file

_doc Quill or Perfection document file

_dat Many uses - used for printer data files by the Psion programs, and general data files by many other programs

_ddf Data Design database file

_ddi Data Design Index file

_ddm Data Design

_dev Device application file

_dm5 Disk Mate 5 file

_dmp Screen dump file

_dtp Used for some Digital Precision Desktop Publisher files

_ebf Easybase database files

_eng Used for help files etc to denote English language files

_exp Export file, used by many programs, generally either comma separated exported data files or exported in text format

_err Error report file

_exc As _exec

_exec Executable programs for the QL

_exf QSpread export filter

_ext SuperBASIC or SBASIC extensions

_fax Fax page file

_f87 Text 87 font file

_fmt Format information file, used by several programs

_fnt A screen font file, used by many graphics programs

_font A screen font file, used by many graphics programs

_gen Code generation files from compilers, or PTR_GEN pointer environment file

_grf Easel graphic file

_gif Graphic Interchange Format

_h A C compiler file

_hdf High Definition Font file, used by Professional Publisher and Page Designer 3

_hlp A help file

_hob Help Object file, used by Psion programs such as Quill

_hrq HRQ compressed file

_idx Index file

_imf QSpread import filter

_img Image or graphics file

_job Job file (executable program)

_jpeg JPG compressed graphics files

_jpg JPG compressed graphics files

_ldp Line Design page, or LDump (JMS) file

_ldr Loader file

_lhq An archiver file format

_lhz An archiver file format

_lib Library file (used by C compilers and assemblers)

_lift Page Designer 1

_lis Listing files

_ljs Lonely Joker game save file

_log Log file, used by BBS systems to log callers etc

_lzh LZH compressed file

_m91 Text 87 Macros file

_man Manual files, or Unix Man pages

_map Map files, used by adventure games, midi, graphics etc programs

_men Menu file used by EasyPtr

_mgr Manager file, used by various device manager programs

_mid Standard MIDI File used by Al Boehm's QL MidiPlayer

_midi As _mid

_mrg A file to be merged with another file

_msg Message files from some BBS systems

_obj QLiberator compiled BASIC program

_ovl QLiberator overlay files

_pad Notepad file

_page Page Designer 1, 2 and 3 page file

_p87 Text 87 printer driver

_pas A Pascal file

_pat Pattern files, used by Page Designer 2 and 3 and The Painter

_pcx a graphics file format widely used on PCs

_pfd Proforma Driver - a printer file used in Proforma from PROGS

_pff Profroma fonts (e.g. used by Line Design and Prowess)

_pic	A picture or pointer environment area save file
_pic0	4 colour _pic file (Mode 0)
_pic4	4 colour _pic file
_pic8	8 colour _pic file
_pkt	Packet files for some BBS systems
_png	Portable Network Graphics (internet format)
_prl	Perl script files
_prt	Print files or printer driver files
_pws	Prowess file
_qed	QED editor file
_qlq	QLQ rext file (JMS) font file (NLQ printing program)
_qpp	A Quick Posters page file
_rc	A configuration file used by programs such as Qascade
_rel	Relocatable file (e.g. used for S-ROFF type files)
_rext	Runtime extensions or BASIC extensions
_rom	ROM image file
_rpm	Resident Procedure Managerfile, for the QEPIII Eprom programmer
_rtm	(1) a runtimes file (compiler extensions) (2) a QLiberator compiled program with compiler runtimes attached to the program
_run	Runtimes, e.g. used by compilers
_saf	Safety files - backups created by some programs
_sav	BASIC program saved in tokenised format, used by SBASIC and by Liberation Software's QLOAD utilities, and by QLiberator
_scn	A screen graphics file
_scr	A screen graphics file
_seq	Sequence data file for Vision Mixer program
_snl	Vision Mixer screen names list
_spr	Sprite file, used by Easyptr and QPTR
_t87	Text 87 text files
_t91	More recent Text 87
_tab	QSpread table file
_tar	Tape Archive file
_task	Turbo compiled task (a compiled BASIC program)
_tex	From the TEX DTP system
_tmp	Temporary files or work files created by some programs (e.g. Quill)
_txt	Plain text files
_win	Hard disk file, typically used in QXL.WIN files for SMSQ and SMSQ/E
_wrk	Work file, used by QLiberator
_xrf	Cross Reference files, used by BASIC syntax checkers etc
_zip	Zipped (compressed/archived file) used by the Zip program
_zoo	ZOO compressed file

During our search to identify these files, we came across some extensions we were not familiar with. Can anyone supply meanings and uses for these filename extensions for us? We'll publish your contributions, along with any other types not covered here, in a later issue.

_abt	_fmt	_out
_acc	_frn	_pgn
_app	_hpr	_prc
_bb8	_ivg	_swc
_cpt	_in	_syn
_css	_inp	_ttl
_dia	_inv	_typ
_ep2	_mod	_weg
_epl	_ok2	

The Ultimate Sin?

Geoff Wicks

I am about to commit what many people will regard as being the ultimate QL sin. No, I am not going to admit to owning a signed portrait of Bill Gates, but I am going to criticise the father of the QL. Was the QL's greatest enemy Clive Sinclair?

This puts me in an unusual situation. I feel like the adult looking back on an unhappy childhood and his parents' constant quarrelling. His life would have been a lot better if his parents had never met, but then if they had never met he would not have been born.

Many QL-ers, including me, are grateful to Clive Sinclair. Most of what we know about computers we learnt from the ZX81, the Spectrum and the QL. I even learnt the fundamentals of MS-DOS on a QL. Two of my uncles retired early because of the computerisation of their work, but Sinclair prepared me for the modern world. In some ways we QL old-stagers know more about computing than today's "whiz-kid", who spends his time surfing the net using the latest technology.

But far from being a Quantum Leap, the QL Clive Sinclair gave us was a shoddy product that deserved to fail. That the

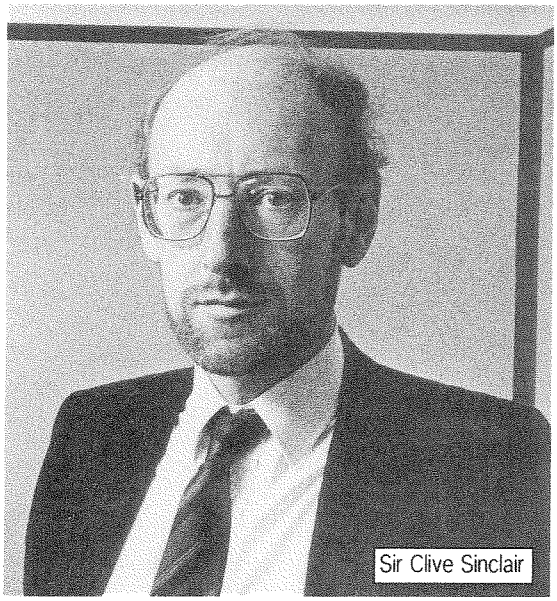
QL is still alive today is not because of, but in spite of, Clive Sinclair.

At about the time the QL was conceived, Sinclair Research was a profitable company. One set of figures I have seen was that in 1982 500,000 home computers were sold in the UK. Of these 220,000 were ZX81s and 75,000 were Spectrums. In 1982 - 83 Sinclair Research had profits of £14M on a turnover of £54M. The following year the profit remained the same although the turnover had risen to £77M.

The question in 1983 was not whether there would be a successor to the Spectrum, but what its specifications would be. The details leaked to the computer press was that the ZX83 would be a battery powered machine using microdrives for storage, a flat screen for viewing, have a built in modem and come packaged with business software.

If Sinclair had succeeded in building a ZX83, it would have been a major technological breakthrough. In 1983 a portable computer was a heavy machine the size of a small suitcase. The ZX83 would probably have been lighter than today's laptops.
Two problems proved insurmountable. The ZX83 consumed too much power to be battery driven, and the flat screen technology never developed beyond a two inch monochrome television.

Once the idea of portability was dropped, no clear vision seems to have emerged as to what the QL should be or to which market it should be aimed. Officially it was to be not a home, but a business computer. The final product was an unhappy hybrid that was neither one nor the other.
Several features of a computer determine whether it is more suitable for home or business use. These include the processor, the memory, the keyboard, the storage, the display and the operating system. When we look at these things one by one, we see inconsistencies in the QL design and realise how muddled the thinking was in Sinclair Research.


Sir Clive Sinclair

Take something simple like the display. For business programs such as spreadsheets and word processing you need an 80 column display and thus a monitor. This was not supplied with the QL, which meant a business purchaser had to pay not £400, but about £650 for his computer. As a compromise Sinclair added a television interface, but you cannot not use mode 4 comfortably on a television set. Then take a look at the chip that controlled most of the in/out operations, the 8049. This does not permit different baud rates for the two serial ports. As the computer did not have a parallel interface, it was difficult, if not impossible, to run both a modem and a printer at the same time. Hardly specifications for a business computer.

Early Sinclair computers were notorious for their strange keyboards, which were, however, acceptable when the aim was to develop a cheap home model. By the time the QL was produced, most of Sinclair's rivals used a proper keyboard, but Sinclair wasted time, energy and money on improving their own keyboard rather than using an alternative. The QL's keyboard was a vast improvement on that of the Spectrum, but totally unsuitable for intensive business use.

Most people, if asked to name just one feature of the QL that ensured its failure, would have little hesitation in naming the microdrives. Looking back we see them as quaint devices, but they were a remarkable technical achievement, certainly when compared to disk drives on a storage capacity/weight basis. They may have had some potential in a portable business computer, but they were too unreliable and too limited in storage capacity for company use.

At about the time I bought my first QL, I was attempting to find a publisher for a book I had written. You do not trust 80,000 precious words, the result of several years' research, to microdrive cartridges. If the QL had had a tape cassette interface it would have given bulk, cheap and safe storage, particularly for the serious home user, but that would have

demeaned a business computer.

There is some dispute about how easy and cheap it would have been to have put disk drives in the QL, but, in practice, QL users were using disk drives within a year of the QL's release. To complicate matters still further microdrive cartridges were in short supply during the first year of the QL's life. A cassette interface would have made things simpler for the software houses, who would probably have produced more QL programs, thus improving its survival chances.

The original QL had 128K memory, which sounded a lot at the time, but 32K was immediately consumed by the display and more by slave blocks and system variables. If you typed as little as 1,000 words in Quill you had a constantly whirring microdrive as it updated the _tmp file. I started to use Quill seriously only after I bought memory expansion, and discovered that temp files were not a normal Quill feature. For serious home use the QL's memory barely insufficient, for company use it was totally inadequate.

Perhaps the most fascinating sign of muddled thinking in the QL design was the choice of processor. The safe choice would have been to have kept the Z80. It was a well used chip that Sinclair's engineers knew how to exploit to the full. Instead the choice fell on the Motorola 68000 series, indicating that Sinclair was looking at technological innovation. But then they chose the 68008, the most limited chip in the series, whose 8 bit bus slowed down and limited the potential performance of the machine. My technical knowledge fails me here, but I believe it was the choice of this chip that led to a restriction on the maximum possible memory of the QL. For years everyone thought this was about 1Mb. It was only after Miracle had thought of making a "Mini QL" on a Gold Card with its own processor that took over from the 68008 that today's high speed, high memory QLs became possible. Because of a lack of speed and memory, the QL failed to keep up with other computers in the development of graphics, vector fonts and communications.

It is hard to believe that when the QL was first released, it had one of the most advanced graphics capabilities of any computer. It was soon beaten by other computers, and it has taken us over 15 years to catch up with the PC. If, 10 years ago, we had had 8 or 16 colours in mode 4 resolution, there would have been more software produced and the QL's survival more assured. It is probably no co-incidence that Macs, which use the 68000 chip, are still seen as superior to PCs in the graphics and printing industries. The final uncertainty in the design of the original QL was the operating system. The original intention was to have a minimal system. Just enough to load and run the Psion suite. This would probably have been satisfactory for business use, but not for the Sinclair fans who wanted to do their own programming. So QDOS and SuperBasic were added, but initially that was only possible by putting a "dongle" or "kludge" into the back. From day one this made the QL an object of derision in both the computer and non-computer press. Few experts took it seriously from then on.

The QL satisfied neither the demands of the business nor of the home market. It was not Sinclair that made the QL what it is today, but the peripherals industry, which supplied the memory, the disk drives, the keyboard interfaces and the operating system that turned it into a serious computer.

The tragedy was the once the peripherals industry had done its work, there were numerous practical examples of the QL's suitability for serious home or small business use. These include a school for teaching computer studies, a magistrates court and a security firm. There are also numerous examples of academics using the QL for writing theses, of authors writing books and of people running clubs and societies. If Sinclair had got the QL specifications right, the market would have been there. The QL probably could never have rivalled the PC, but it could have rivalled the Mac.

We QL-ers bear a grudge against Amstrad for "killing" the QL after they took over Sinclair, but if we are honest Amstrad's decision was commercially sensible and sound. It was not Amstrad that "killed" the QL, but Sinclair.

But let us indulge our fantasies and assume that it was Sinclair and not Microsoft that had conquered the world. What would we QL-ers be doing? I suspect many of us belong to the awkward squad. Perhaps we would be tinkering with a minor system called MS-DOS and arguing among ourselves if Windows was a necessary improvement to the purity of the original system.

TF Services

Compswitch

A UK 4-way trailing socket designed to switch off computer peripherals automatically when the computer is switched off, or (in the case of an ATX computer) when it auto-powers down. *Compswitch* has one control socket, and three switched sockets.

Cost..... £24

***************NEW***************

superHermes

A major hardware upgrade for the QL

All Hermes features (working ser1/2 at 19200, independent baud rates/de-bounced keyboard/ keyclick) IBM AT kbd I/f // HIGH SPEED RS232 at 57600// serial mouse port and 2 other RS232 inputs// 3 I/O lines // EEPROM

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

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

SuperHermes LITE: All Hermes features (see above) + an IBM AT keyboard interface only.
Cost (incl keyboard lead) .....**£53** (£54/£55)

QL REPAIRS (UK only)

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

£27 incl 6 month guarantee

Minerva

The ORIGINAL system operating system upgrade

OTHER FEATURES COMMON TO ALL VERSIONS

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

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

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

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

QL RomDisq

Up to 8 mbyte of flash memory for the QL

A small plug in circuit for the QL's ROM port (or Aurora) giving 2, 4 or 8 mbytes of permanent storage - it can be thought of as a portable hard disk on a card, and reads at some 2 mbytes per second.

Think of it - you could fully boot an expanded QL, including all drivers/SMSQ etc off **RomDisq** at hard disk speed with only a memory expansion needed.

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

MPLANE

A low profile powered backplane with ROM port

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

Cost ..... £34 (£35/£36)

I2C INTERFACES

Connects to Minerva MKII and any Philips I²C bus

Power Driver Interface 16 I/O lines with 12 of these used to control 8 current carrying outputs (source and sink capable)
2 amp (for 8 relays, small motors) ..... £40 (£43/£44)
4 amp total (for motors etc) ..... **£45** (£48/£50)
Relays (8 3a 12v 2-way mains relays (needs 2a power driver).....**£25** (£28/£29)
Parallel Interface Gives 16 input/output lines. Can be used wherever logic signals are required.....**£25** (£27/£28)
Analogue Interface Gives eight 8 bit analogue to digital inputs (ADC) and two 8 bit digital to analogue outputs (DAC). Used for temp measurements, sound sampling (to 5 KHz), x/y plotting .....**£30** (£31/£32)
Temp probe (-40°C to +125°C).....**£10** (£10.50/£11)
Connector for four temp probes.....**£10** (£10.50/£11)
Data sheets.....**£2** (£2.50/£3)
Control software & manual (for all I/F).....**£2** (£2.50/£3)

QL SPARES

Keyboard membrane ..... **no longer on sale**
1377 PAL ..... **£3** (£3.50/£4)
Circuit diagrams ..... **£3** (£3.50/£4)
68008 cpu or 8049 IPC..... **£8** (£8.50/£9)
8301/8302 or JM ROM or serial lead.....**£10** (£10.50/£11)
Power supply (sea mail overseas)..... **£12** (£17/£21)

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

10 May 01

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

Returning from a DO

Al Boehm

As an American (English?) idiom, the title could mean "Returning from a party"! Sorry if I got you party animals all excited anticipating reading about the hijinks I got into trying to get home from a wild party, but it is merely about the Keyword DO. On the other hand, you programmers may find this article quite interesting.

The TK2 Keyword DO lets you run unnumbered SuperBasic code that is stored in a file. For example, DO win1_QDtxt could start the editor QD with txt files as the default load suffix. There also could be a win1_QDbas to start QD with the bas suffix. There can be one for full screen QDs, part screen QDs, and so on.

I also use ALTKEY for nearly the same purposes, but I ran out of single letters that had any meaning for me. Thus, I have a large directory of Boots to start programs with various options. Besides booting programs I also DO other SuperBasic utility programs to set printers, change windows, dump the screen a certain way, etc.

Most of the time I use DO from the command line. However, it can be used in a SuperBasic program. The trouble has been once DO completes its task, it stops. Thus, it has only been useful in a program for such things as ending tasks such as closing files or printing debug.

I queried Simon Goodwin to see if there was a way to return from a DO. He did not have a simple method, but something he wrote triggered a brainstorm. The problem is that DO does not know where to go back to. What do you do when a program stops and you want it start where it stopped? You use CONTINUE which restarts the program at the line after the one where it stopped. Would this work with DO?

It does! Try the following test program:

```
100 FOR I = 1 to 10
110 DO ram1_DOtest
120 END FOR I
```

with the file ram1_DOtest consisting of:

```
FOR J=1 TO 5:PRINT I,J:END FOR J:CONTINUE
```

This worked fine with the JSU rom and Minerva both with and without the Pointer System. It also worked with SMSQ and SMSQE except when the SBasic program ended, the DO line started and kept repeating itself until Ctrl/Space was pressed. The problem as I see it, is not that the DO keeps calling itself with CONTINUE after the program has stopped, but rather why is called at all after a STOP. A simple workaround is to use a second DO without a CONTINUE, for example in the above program for SMSQE add:

```
130 DO ram1_DOStop
```

where the file DOStop is simply:

```
STOP
```

A limitation in the JSU rom and Minerva is that the DO will not run nor return if it is called from a FUNction. It will work OK from a PROCedure. But with SMSQE, it also works fine from a FUNCtion.

Some years ago, it was very important for me to evaluate complicated algebraic equations. I mean these dudes were sometimes several pages long! Once I even exceeded the 32K length for string variables! No, I didn't type in 32K of algebra; simple equations were recursively expanded by substituting expressions such as polynomials for single variables.

On the Spectrum and TS2068 there was an VAL function that would calculate the result of an algebraic statement written in Basic. For example,

```
A$='12+3.2*COS(1.3*PI)'
PRINT VAL(A$)
```

would result in 10.11909. Coercion in the QL: A=A$, only gets the first number, A=12. Several attempts were made to develop a VAL for the QL. The mini-compiler QL_VAL developed by M. Garcia, Quanta disk 15, is particularly noteworthy. Also there is Simon Goodwin's QUEUE% function (DIY vol Q) which stuffs an input buffer and allows a string to be evaluated. But for my work they all had shortcomings.

Now (when I don't need it anymore) it is possible to write an expression to ram disk, evaluate it, and return the answer:

```
100 REMark Test EVAL
110 a$='12+3.2*COS(1.3*PI)'
120 EVAL a$
```

```
125 PRINT result
130 STOP
140 :
500 DEFine PROCedure EVAL(a$)
510 REMark Compute numeric result of
    expression a$
515 REMark answer is returned as
    variable "result"
520 REMark Freeware. 28 Aug 2001 by A.
    Boehm
530 LOCal F
540 F=FOP_OVER('ram1_EVAL_str')
550 PRINT#F,'result=' & a$ & ':CONTINUE'
560 CLOSE#F
570 DO ram1_EVAL_str
590 END DEFine EVAL
```

This test should print 10.11909 when run. I have an FVAL which is similar except it is a FUNction for use in SMSQE only.

Aside from evaluating an expression, this use of a returning DO could be used for some limited on-the-fly code manipulation. That is, the SuperBasic program could change a$ into series of commands which could then be executed.

DO requires TK2. MERGE seems to work just as well from the command line. However, MERGE does not work within SuperBasic. Plus I don't think DO would know where to return in a compiled program.

Programming in Assembler - Part 9

Norman Dunbar

First a list of corrections that I have to make to the last article which covered the QDOS screen memory etc.

Page 30, example code - EXTOP code missing. The following code should have been in the code snippet for the example shown:

```
extop   move.l  $32,(a0),a1
        move.w  $64(a0),d1
        moveq   #0,d0
        rts
```

Page 30, paragraph 2 (below the code example) - wrong register used in text. The sentence '... but coming out, D1.W holds ... and A2.L holds ....' should of course read 'A1.L holds ....' and not A2 as I originally wrote.

That's all for now - I'm sure there are more bugs in there somewhere - but either nobody has spotted them (or nobody read the article).

Subroutines & A Useful Library

Here we are in part 9 of the series on assembly language for the QL (not including the QLTdis episodes which are 'the project') and what we will look at today are subroutines.
A subroutine is simply a piece of code that you call lots of times within your program. Because it is called so many times, you extract the working code, move it somewhere safe and add an RTS at the end. This is your subroutine - in its draft form!

Where the code used to be in the main source, now simply has a 'BSR sub_routine' in its place. The more times a routine is called, the bigger the saving in your typing and memory usage in the final program. Another major advantage of using subroutines is that you only need to change or correct them once - of course, if you make a mistake then every call to that subroutine is flawed as well!

For example, in a program you have written, you might find that you write the same piece of code numerous times to clear the screen, something like the following:

```
start  blah blah blah
:
:
       move.l  channel_id,a0        ; First channel id
       moveq   #sd_clear,d0         ; CLS
       moveq   #infinite,d3         ; Infinite timeout
       trap    #3                   ; CLS title window
:
:
       move.l  other_channel_id,a0 ; Another channel id
       moveq   #sd_clear,d0         ; CLS
       moveq   #infinite,d3         ; Infinite timeout
       trap    #3                   ; CLS title window
:
:
       move.l  another_id,a0        ; And another
                                    ; channel id
       moveq   #sd_clear,d0         ; CLS
       moveq   #infinite,d3         ; Infinite timeout
       trap    #3                   ; CLS title window
:
:
       rts             ; All done - back to
                       ; wherever I came from
```

and so on. The above code looks duplicated and where you have duplication, you can usually - but not always - extract the duplicate code to a subroutine. We can now rewrite the code above as follows:

```
start blah blah blah
:
:
      move.l  channel_id,a0     ; First channel id
      bsr     cls
:
:
      move.l  other_channel_id,a0 ; Another channel id
      bsr     cls
:
:
      move.l  another_id,a0     ; And another channel id
      bsr     cls
:
:
      rts     ; All done - back to wherever I came from

*-------------------------------------------------------
* Subroutine to clear the SCR or CON channel whose ID
* is held in A0.
*-------------------------------------------------------
cls   moveq   #sd_clear,d0        ; CLS
      moveq   #-1,d3              ; Infinite timeout
      trap    #3                  ; CLS title window
      rts
```

The code that does the setting up of the various parameters for the system call to clear a channel has been extracted and placed at the end all by itself. An RTS instrcution has been added to allow us to go back to where we came from. The second piece of code is easier (?) to read and will be smaller when finished.

So that is all there is to it. If you remember back to the boring part of this series (what do you mean 'which boring part?') where I discussed the inner workings of the BSR instruction, you will remember that BSR stacks the address of the instruction that will be exected next (after the BSR), jumps to the address given and continues executing from there until it finds an RTS instruction.

The RTS instruction stops the program in its tracks, sets the PC (2 points if you can remember what PC stands for ...) to the address that was stacked and proceeds to execute from there again. Those of you who are ahead of me at this point will realise that the RTS instruction takes the top 4 bytes off of the stack REGARDLESS of what they are. If they are a valid return address then fine, no problems. If, on the other hand, they are some data, then who knows what will happen when the RTS is executed.

For this reason, it is very important that your stack should be exactly the same on the way out of a subroutine as it was on the way in. Don't do this, for example:

```
start blah blah blah
:
:
      move.l  channel_id,a0       ; First channel id
      bsr     cls
:
:
      rts

*-------------------------------------------------------
* BROKEN subroutine to clear the SCR or CON channel
* whose ID is held in A0.
*-------------------------------------------------------
cls   move.l  d0,-(a7)            ; Preserve D0 until later
      moveq   #sd_clear,d0        ; CLS
      moveq   #-1,d3              ; Infinite timeout
      trap    #3                  ; CLS title window
      rts                         ; Program explodes here!
```

In this example, the old value of D0.L is on the stack on top of the return address. When the RTS instruction is executed it doesn't know (or care) about what is on the stack, it just grabs the top 4 bytes and sets the PC to that 'address'. (You get 2 points if you remembered PC = Program Counter!)

Building A Library

As you progress with assembly language programming, you may find that you build up a lot of subroutines in your programs. What to do with them all?

Why not build yourself a library of routines that you can include in every program that needs them. This way, you have a full set of tried and tested bits of code - which you should document somewhere - that can be reused over and over again. The rest of this article will help you on your way by building a number of useful (well, I have found them to be useful over the years) subroutines that you can use.

Documentation

As with all good things, documentation is a must. If you have a large number of useful routines then they should be documented somewhere. This will allow you to look for a routine in your library and from that, find out its input & output parameters and which file it lives in.

A suitable template that you could use for each subroutine is as follows:

```
*-------------------------------------------------------
* NAME
* DEPENDENCY (1)
* DEPENDENCY (2)
* PURPOSE
* INPUTS
* OUTPUTS
*-------------------------------------------------------
```

The above looks very like comments in a source file – this implies that we could add the documentation to the source file and then run a utility program to extract the details and store them in a text file – which you can edit and/or print as desired.

Having a standard header above each subroutine also implies that you could write a utility program to scan the entire library and ask you which ones you want to include in your output file – which will be you source file for your next masterpiece – before extracting them all and writing them to this file.

As for the subroutines themselves, I mentioned above that they exist in a draft form when you simply extract the code from the 'wordy' source and add an RTS to the end. This is fine, but it could be that you need to preserve certain registers so that the code calling the subroutine doesn't need to keep saving and restoring them. The updates required are:

- Check which registers will be used by the code explicitly – save them before and restore them after the main part of the subroutine code.
- Check which system calls are made by the subroutine and look up the QDOSMSQ documentation to see which registers are trashed by the system call. Add these registers to the save and restore routines.
- Save the registers as the first line of code in the subroutine and restore them as the line immediately before the RTS (or as near to the RTS as possible).
- Always have the subroutine return an error code and/or the flags set to signal if an error occurred or not.

An actual example follows:

```
*-------------------------------------------------------
* NAME         CLS
*-------------------------------------------------------
* DEPENDENCY   None
* PURPOSE      To clear a screen/console channel.
* DESCRIPTION  Clears the screen channel whose ID is
*              supplied in A0.
* INPUTS :
*              A0.L = channel ID
* OUTPUTS :
*              D0 = Error code
*              Z flag set if no errors, unset otherwise
*-------------------------------------------------------
```

```
cls   move.l   d1/d3/a1,-(a7) ; These will be corrupted
                              ; by SD_CLEAR
      moveq    #sd_clear,d0   ; CLS (SD_CLEAR defined
                              ; in GWASL)
      moveq    #-1,d3         ; Infinite timeout
      trap     #3             ; CLS the window
      move.l   (a7)+,d1/d3/a1 ; Restore corrupted
                              ; registers
      tst.l    d0             ; Set Z flag if all ok,
                              ; else not set
      rts
```

In the above example, I have extended the 'cls' code from our original subroutine as follows:

- I have added a documentation header comment.
- I have preserved D3 because I use it in the code myself.
- I have preserved D1 and A1 because the QDOSMSQ documentation states that these two registers are 'undefined' on return from the system call SD_CLEAR.
- I have restored all 3 of these registers before the RTS.
- I have added a TST.L D0 instruction to set the Z flags according to whether an error was detected or not.

Note that although D0 is used by the code and by the system call, I have not preserved it. This is quite simply because I use D0 to return any error codes back to the caller. As I have documented its corruption in the header, I assume that the user of the subroutine will read this and know all about it!
Bullet proofing the code like this helps to reduce unexpected bugs in your programs when you forget to save a register and after a subroutine call, assume it still has the same value as before. I know, I have been there. Of course, there is not much you can do to prevent the documentation you use from being wrong (been there too) but at least you did your best !

Onwards with the code for my (useful) subroutines.

```
*-------------------------------------------------------------------------------
* NAME          STR_COPY
*-------------------------------------------------------------------------------
* DEPENDENCY    None
* PURPOSE       Copy the string at (A2) to the string at (A1) overwriting.
* DESCRIPTION   Copies the string whose address is passed in A2 over the
*               string whose address is passed in A1 thus overwriting the
*               old contents of the receiving string.
* INPUTS :
*               A1.L = Address of the receiving string
*               A2.L = Address of the sending string
* OUTPUTS :
*               A1.L = Address of the receiving string (preserved)
*               A2.L = Address of the sending string (preserved)
*-------------------------------------------------------------------------------
str_copy    movem.l d0/a1-a2,-(a7)  ; Preserve working register
            move.w  (a2)+,d0        ; Get size of 'from' string
            move.w  d0,(a1)+        ; Set new size of 'to' string
            bra.s   sc_next         ; Skip the dbra stuff first time round
sc_moveb    move.b  (a2)+,(a1)+     ; Move a single byte
sc_next     dbra    d0,sc_moveb     ; And the rest
            movem.l (a7)+,d0/a1-a2  ; Restore working registers
            rts                     ; Exit
```

```
*-------------------------------------------------------------------------------
* NAME          STR_APPEND
*-------------------------------------------------------------------------------
* DEPENDENCY    STR_COPY
* PURPOSE       Append the string at (A2) to the end of the string at (A1).
* DESCRIPTION   Copies the string whose address is passed in A2 to the end of
*               the string whose address is passed in A1. The old contents of
*               both strings will be preserved - except A1 which will be
*               extended of course !
* INPUTS :
*               A1.L = Address of the receiving string
*               A2.L = Address of the sending string
* OUTPUTS :
*               A1.L = Address of the receiving string (preserved)
*               A2.L = Address of the sending string (Preserved)
*-------------------------------------------------------------------------------
```

```
str_append  movem.l  d0/a1-a2,-(a7) ; Save the working register
            move.w   (a2)+,d0       ; Size of 'from' string
            move.w   (a1),d1        ; Size of 'to' string
            add.w    d0,(a1)+       ; New size of 'to' string
            adda.w   d1,a1          ; New 'to' string end position
            bra.s    sc_next        ; Copy all bytes over using STR_COPY
                                    ; D0 will be restored after STR_COPY exits
                                    ; STR_APPEND exits via STR_COPY.
```

```
*-------------------------------------------------------------------------
* NAME          STR_REVERSE
*-------------------------------------------------------------------------
* DEPENDENCY    None
* PURPOSE       Reverse the bytes in the string at (A1).
* DESCRIPTION   Reverses the bytes in the string whose address is passed in A1.
* INPUTS :
*               A1.L = Address of the string to be reversed
* OUTPUTS :
*               A1.L = Address of the string to be reversed (Preserved)
*-------------------------------------------------------------------------
str_reverse move.l  d0-d1/a1-a2,-(a7)   ; Save working registers
            move.l  a1,a2               ; Copy start address
            move.w  (a1)+,d0            ; Fetch length word
            beq.s   sr_quit             ; Nothing to do
            adda.w  d0,a2               ; Near the end of the string
            addq.l  #1,a2               ; The last character in the string
            lsl.w   #1,d0               ; D0 = INT(D0/2) OK for odd and even !
            bra.s   sr_next             ; Skip the first one for DBRA
sr_loop     move.b  (a2),d1             ; Fetch the 'last' character
            move.b  (a1),(a2)           ; Move the 'first' byte to 'last'
            move.b  d1,(a1)+            ; Move the 'last' byte to 'first'
            subq.l  #1,a2               ; And adjust 'last'
sr_next     dbra    d0,sr_loop          ; And do the rest
sr_quit     movem.l (a7)+,d0-d1/a1-a2   ; Restore the working registers
            rts
```

```
*-------------------------------------------------------------------------
* NAME          STR_INSERT
*-------------------------------------------------------------------------
* DEPENDENCY    STR_APPEND
* PURPOSE       Insert the string at (A2) into the string at (A1) at pos D0.
* DESCRIPTION   Inserts the string whose address is passed in A2 into the string
*               whose address is passed in A1 at the position passed in D0.W so
*               the first character in the inserted string will be character D0
*               after the insertion. (Zero is the very first character !)
*               If D0 >= length (A1) then call STR_APPEND to do the work.
* INPUTS :
*               A1.L = Address of the receiving string
*               A2.L = Address of the string to be inserted
*               D0.W = Position (starting at 0) where to insert before
* OUTPUTS :
*               D0 = Error code
*               A1.L = Address of the receiving string (preserved)
*               A2.L = Address of the string to be inserted (preserved)
*               Z flag set if no errors, unset otherwise.
*-------------------------------------------------------------------------
str_insert  cmp.w   d0,(a1)             ; Are we appending perhaps ?
            bge     str_append          ; Yes, easy case to deal with !
            tst.w   d0                  ; Is there anything in D0 ?
            bge.s   si_ok               ; Yes, negatives are bad !
            moveq   #-15,d0             ; Bad parameter
            rts                         ; Z is unset, D0 = error code

si_ok       movem.l d1/a1-a4,-(a7)      ; Save those workers
            move.l  a1,a3               ; A3 = Address of A1 string
            adda.w  (a1),a3             ; Plus the size ...
            addq.l  #2,a3               ; A3 = final char of A1 string plus one.
            move.l  a3,a4               ; A4 to be the new last char after insert
```

```
            adda.w  (a2),a4             ; Add the extra length to the last character
            addq.l  #2,a4               ; And now we are there (plus one)
            move.w  (a2),d1             ; Size of inserted string
            bra.s   si_dnext            ; Skip dbra
si_dmove    move.b  -(a3),-(a4)         ; Move a byte
si_dnext    dbra    d1,si_dmove         ; Do the rest
            move.w  (a2),d1             ; Fetch the inserted length again
            adda.w  d1,a2               ; A2 now nearly at the last character
            addq.w  #2,a2               ; One past the last character
            bra.s   si_inext            ; Skip dbra stuff
si_imove    move.b  -(a3),-(a4)         ; Insert a byte
si_inext    dbra    d1,si_imove         ; Insert the rest
            movem.l (a7)+,d1/a1-a4      ; Restore those workers
            clr.l   d0                  ; No errors
            rts
```

```
*-------------------------------------------------------------------------------
* NAME          STR_COMP
*-------------------------------------------------------------------------------
* DEPENDENCY    None
* PURPOSE       To compare two strings for exact equality
* DESCRIPTION   Comprare the strings at (A1) and (A2) for exact equality.
*               Numbers in the string are considered as well as letters etc.
*               Equivalent to 'IF (A1$ = A2$)'
* INPUTS :
*               A1.L = First string
*               A2.L = Second string
* OUTPUTS :
*               D0 = Result of comparison.
*                    -1 = A1 string is ‹ A2 string
*                     0 = A1 string = A2 string
*                    +1 = A1 string › A2 string
*               A1.L = First string (preserved)
*               A2.L = Second string (preserved)
*-------------------------------------------------------------------------------
str_comp    movem.l a0-a2,-(a7)         ; Must preserve workers
            moveq   #2,d0               ; Include numbers, case is significant
sc_params   move.l  a1,a0               ; Uses different registers
            move.l  a2,a1               ; So swap them over
            move.w  UT_CSTR,a2          ; Fetch the vector address
            jsr     (a2)                ; Compare strings using ROM routine
            movem.l (a7)+,a0-a2         ; Restore working registers
            tst.l   d0                  ; Make sure Z is set/unset
            rts
```

```
*-------------------------------------------------------------------------------
* NAME          STR_COMPI
*-------------------------------------------------------------------------------
* DEPENDENCY    STR_COMP
* PURPOSE       To compare two strings for approximate equality
* DESCRIPTION   Comprare the strings at (A1) and (A2) for approximate equality.
*               Numbers in the string are considered and letter case is ignored.
*               Equivalent to 'IF (A1$ == A2$)'
* INPUTS :
*               A1.L = First string
*               A2.L = Second string
* OUTPUTS :
*               D0 = Result of comparison.
*                    -1 = A1 string is ‹ A2 string
*                     0 = A1 string == A2 string
*                    +1 = A1 string › A2 string
*               A1.L = First string (preserved)
*               A2.L = Second string (preserved)
*-------------------------------------------------------------------------------
str_compare movem.l a0-a2,-(a7)         ; Must preserve workers
            moveq   #3,d0               ; Include numbers, case is significant
            bra     sc_params           ; Jump into STR_COMP to do it all
```

CAMBRIDGE Z88

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

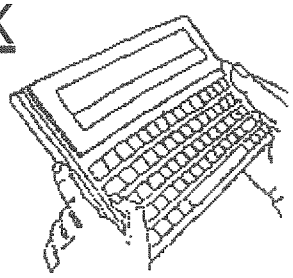
MOBILE: 07808 576118

E-MAIL: WNR@COMPUSERVE.COM

6 Ravensmead
Chalfont-St-Peter
Buckinghamshire SL9 0NB

THE IDEAL PORTABLE COMPANION FOR THE QL, PC AND MAC.

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) *

WE DO REPAIRS, PART EXCHANGES, AND BUY Z88's & PARTS

NEW! CAMBRIDGE Z88 WITH OZ4+512k Internal Ram £150

ALSO AVAILABLE, 512k RAMPACK £40

DESCRIPTION	UK	USA
CAMBRIDGE Z88 ISSUE4-128K	**£120**	**$180**
CAMBRIDGE Z88 COMPUTER	**£99**	**$130**
☆**RECONDITIONED Z88**	**£60**	**$90**
32k RAMPACK	£18	$25
128k RAMPACK	£38	$50
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**	
TOPPER (PROTECTIVE COVER)	£12	$16
CARRYING CASE (PLASTIC)	£10	$12

1/2 PRICE STARTER PACK
1 - Z88 ☆ PLUS
TRANSFER KIT
+ 32K RAM
+ 32K EPROM
£72 ONLY!!

FILE TRANSFER TO OTHER COMPUTERS		
SPECIAL Z88-QL SERIAL LEAD	£10	$15
COPY DISKS OF QUANTA PROGS IMP/EXP & ARCHIVE EXPORT	£2	$3

PCLINK KIT (For PCs)	ALL WITH INSTRUCTION BOOKS, CABLES AND EPROMS AS REQUIRED!	£25	$35
Z88 TO MAC KIT		£25	$35
Z88 TO BBC KIT		£25	$35

POSTAGE UK UP TO £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

```
*-------------------------------------------------------------------------------
* NAME          FILE_CLOSE
*-------------------------------------------------------------------------------
* DEPENDENCY    None
* PURPOSE       Close the channel passed in A0
* DESCRIPTION   Close the file channel whose QDOS ID is in A0. To prevent
*               any original QL systems from serious problems, checks for #0
*               being closed and ignores it.
* INPUTS :
*               A0.L = Channel ID to be closed
* OUTPUTS :
*               D0 is preserved as IO_CLOSE does not return errors except NOT
*               OPEN and we ignore these here ! The Z flag is indeterminate
*               after this subroutine.
*               A0.L is returned 'undefined' to avoid using the channel again !
*-------------------------------------------------------------------------------
file_close  cmpa.l  #0,a0               ; Test for SuperBasic #0
            beq.s   fc_exit             ; Ignore it
            move.l  d0,-(a7)            ; Preserve the worker
            moveq   #io_close,d0        ; Prepare to close it
            trap    #2                  ; Close it
            move.l  (a7)+,d0            ; Restore the worker
fc_exit     rts
```

```
*-------------------------------------------------------------------------------
* NAME          FILE_OPEN
*-------------------------------------------------------------------------------
* DEPENDENCY    None
* PURPOSE       To open a file like 'OPEN #3,filename'
* DESCRIPTION   Opens a file in mode 0 (old exclusive device) The filename is
*               passed in A0 (a pointer to the name). The current job assumes
*               ownership of the channel. May need a TRAP #4 before calling if
*               the filename is relative A6 when called. (SuperBasic).
* INPUTS :
*               A0.L = Pointer to filename
* OUTPUTS :
*               A0.L = Channel id.
*               D0 = Error code
*               Z flag set if no errors, unset otherwise.
*-------------------------------------------------------------------------------
file_open       movem.l d1-d3,-(a7)     ; Those workers need saving
                moveq   #0,d3           ; Old exclusing device mode
fo_params       moveq   #IO_OPEN,d0     ; Trap code
                moveq   -1,d1           ; Current job owns the channel
                trap #2                 ; Open it
                movem.l (a7)+,d1-d3     ; Restore workers
                tst.l   d0              ; Make sure Z is set/unset
                rts
```

```
*-------------------------------------------------------------------------------
* NAME          FILE_OPENIN
*-------------------------------------------------------------------------------
* DEPENDENCY    FILE_OPEN
* PURPOSE       To open a file like 'OPEN_IN #3,filename'
* DESCRIPTION   Opens a file in mode 1 (old shared device) The filename is
*               passed in A0 (a pointer to the name). The current job assumes
*               ownership of the channel. May need a TRAP #4 before calling if
*               the filename is relative A6 when called. (SuperBasic).
* INPUTS :
*               A0.L = Pointer to filename
* OUTPUTS :
*               A0.L = Channel id.
*               D0 = Error code
*               Z flag set if no errors, unset otherwise.
*-------------------------------------------------------------------------------
file_openin     movem.l d1-d3,-(a7)     ; Those workers need saving
                moveq   #1,d3           ; Old shared device mode
                bra     fo_params       ; Do the rest via FILE_OPEN
```

```
*------------------------------------------------------------------------------
* NAME          FILE_OPENNEW
*------------------------------------------------------------------------------
* DEPENDENCY    FILE_OPEN
* PURPOSE       To open a file like 'OPEN_NEW #3,filename'
* DESCRIPTION   Opens a file in mode 2 (new exclusive device) The filename is
*               passed in A0 (a pointer to the name). The current job assumes
*               ownership of the channel. May need a TRAP #4 before calling if
*               the filename is relative A6 when called. (SuperBasic).
* INPUTS :
*               A0.L = Pointer to filename
* OUTPUTS :
*               A0.L = Channel id.
*               D0 = Error code
*               Z flag set if no errors, unset otherwise.
*------------------------------------------------------------------------------
file_opennew    movem.l d1-d3,-(a7)     ; Those workers need saving
                moveq   #2,d3           ; New exclusive device mode
                bra     fo_params       ; Do the rest via FILE_OPEN
```

```
*------------------------------------------------------------------------------
* NAME          FILE_OPENOVER
*------------------------------------------------------------------------------
* DEPENDENCY    FILE_OPEN
* PURPOSE       To open a file like 'OPEN_OVER #3,filename'
* DESCRIPTION   Opens a file in mode 3 (new overwrite device) The filename is
*               passed in A0 (a pointer to the name). The current job assumes
*               ownership of the channel. May need a TRAP #4 before calling if
*               the filename is relative A6 when called. (SuperBasic).
* INPUTS :
*               A0.L = Pointer to filename
* OUTPUTS :
*               A0.L = Channel id.
*               D0 = Error code
*               Z flag set if no errors, unset otherwise.
*------------------------------------------------------------------------------
file_openover   movem.l d1-d3,-(a7)     ; Those workers need saving
                moveq   #3,d3           ; New overwrite device mode
                bra     fo_params       ; Do the rest via FILE_OPEN
```

```
*------------------------------------------------------------------------------
* NAME          FILE_OPENDIR
*------------------------------------------------------------------------------
* DEPENDENCY    FILE_OPEN
* PURPOSE       To open a file like 'OPEN_DIR #3,devicename'
* DESCRIPTION   Opens a file in mode 4 (directory) The filename is
*               passed in A0 (a pointer to the name). The current job assumes
*               ownership of the channel. May need a TRAP #4 before calling if
*               the filename is relative A6 when called. (SuperBasic).
* INPUTS :
*               A0.L = Pointer to filename
* OUTPUTS :
*               A0.L = Channel id.
*               D0 = Error code
*               Z flag set if no errors, unset otherwise.
*------------------------------------------------------------------------------
file_opendir    movem.l d1-d3,-(a7)     ; Those workers need saving
                moveq   #4,d3           ; Directory mode
                bra     fo_params       ; Do the rest via FILE_OPEN
```

So, that was plenty of useful routines. Next issue, we will add more routines, including various input and output (now that you know how to open files).

There will also be a BASIC listing which shows you how to extract details from libraries and how to check for depencies.

Devices in QPAC2, QMenu etc.

Jochen Merz

One or two customers complained about the fact, that QPAC2 (and QMenu's File-Select menu, for example) always underline the first character of a device name in the device select window.
This has not been a problem in the past, at least not until RomDisq came into existance. Both RAM for any RAM-Disk and ROM for the RomDisq start with "R", so there's a clash.
The same has happened with the new QPC2's DOS-Device: it starts with "D", and so does DEV for a long time!
As I am using both DOS and DEV, I can see the problem - it is pretty annoying not to be able to use the selection key.
The solution, however, is quite easy: rename the new devices so that they start with a unique character!
I would not suggest to rename the "well etablished" device names, e.g. MDV, FLP, RAM, WIN and DEV, but I do not see any problems in renaming the new ones. Dream up replacement names, however, is not as easy. How about "PCW" for "PC WIN" or "MSP" for Microsoft Partition?
Any suggestions for RomDisq? Difficult, I have yet to come up with anything sensible yet.

Error Trapping

David Denham

Continuing to try to educate myself on QL SuperBASIC and sharing my learnings with QL Today readers, here's a short article about error trapping on the QL. I write a lot of programs for myself in SuperBASIC, so error trapping is very important for me as some of the programs get used in a business environment.
QL error trapping seems to revolve around a few types of structures:

- the WHEN ERROR keywords, along with RETRY, CONTINUE, ERNUM, ERLIN and REPORT keywords
- the Turbo Compiler's WHEN_ERROR and DEVICE_STATUS structures
- QLiberator's Q_ERR_ON, Q_ERR_OFF and Q_ERR
- Toolkit 2 style functions which add new functions in place of existing SuperBASIC procedures - these functions make it easier to trap file errors.

Error handling in the context of this article falls into four main categories:

- User input errors, e.g. a program expects a number, the user enters text and the program falls over.
- File error trapping
- Bad programming not catering for uncommon or unexpected factors arising.
- Actual programming errors

Not a lot can be done about the fourth example, apart from perhaps a "catch-all" error message which at least allows the program to "fall over" tidily.

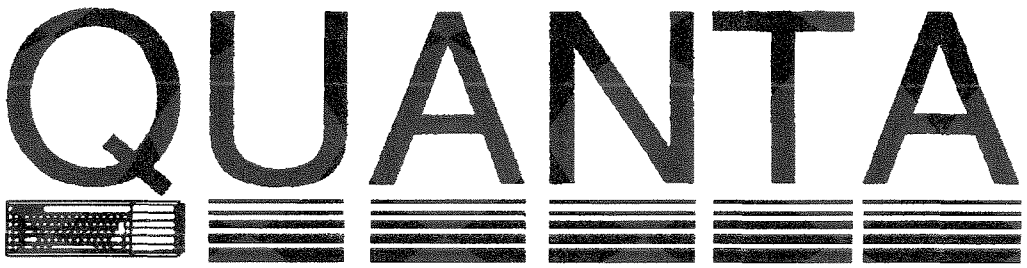
WHEN ERROR

This is a general purpose error trapping structure which is part of SuperBASIC in versions of the QL ROM from version JS onwards. It does not work on early versions of the QL ROM like versions FB, AH and JM. To check which version of the QL ROM you have, enter the command PRINT VER$. This will print a 2, 3 or 4 letter code indicating the version of SuperBASIC (or SBASIC if you are using the SMSQ). Minerva users will get the code 'JSL1', while SMSQ users will get 'HBA'
There are some minor problems with WHEN ERROR in versions JS and MG of SuperBASIC, but the keywords are largely useable.
The structure is defined with a WHEN ERROR statement (SuperBASIC actually lists it as WHEN ERRor - you can just type in WHEN ERR and BASIC will expand it to WHEN ERRor. The end of the structure is indicated with an END WHEN statement. Any lines of code between these statements are ignored until an error occurs, then the program executes what's between these statements when an error happens.

```
100 WHEN ERRor
110  PRINT"Oops...!"
120 END WHEN
130 INPUT "Enter a number › ";number
```

The above little program asks you to enter a number. Enter some text or some other non-numerics and the program will say 'Oops' if you manage to cause an error. In practice, it does little to prevent the program falling over when an error happens, but it's a start.

The functions ERNUM and ERLIN will tell you the error number code and line at which the last error occurred respectively.



Independent QL Users Group

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

Barclaycard: Visa: Access: Mastercard

*** Now in our Eighteenth successful year ***

Further details from the Membership Secretary

Bill Newell, 213, Manor Road
Benfleet, Essex, SS7 4JD
Tel. +44(0)1268 754407
or
Visit the Quanta Web Site
http://www.quanta.uni.cc
email: quanta_membership@uk2.net

```
100 WHEN ERRor
110   PRINT"Oops! Error ";ERNUM;' at line
      ';ERLIN
120 END WHEN
130 INPUT "Enter a number › ';number
```

The keyword REPORT will print the error message corresponding to the error code number. By specifying a channel number the message can be printed to a screen window of your choice.

```
100 WHEN ERRor
110   PRINT"Oops! Error ";ERNUM;' at line
      ';ERLIN
115   REPORT #1,ERNUM
120 END WHEN
130 INPUT "Enter a number › ';number
```

If you want to turn off error trapping for any reason in any part of your program, you can just execute an empty WHEN ERRor clause:

```
1000 REMark oh oh, no error trapping after
     this!
1010 WHEN ERRor
1020 END WHEN
```

In practice, this doesn't seem to work too well on some ROM versions.

You can add CONTINUE or RETRY statements to specify how the error handler will pick up after the error. CONTINUE will (as the name implies) let the program just carry on after the error - your program can check what happened and take appropriate action - while RETRY will make it go back to the statement which caused the error, useful for INPUTs for example.

```
100 WHEN ERRor
110   REPORT #1,ERNUM : PRINT" at line ";ERLIN
120   PRINT"You should enter a number, silly!"
130   RETRY
140 END WHEN
150 INPUT"Enter number 1 › ";num1
160 INPUT"Enter number 2 › ";num2
```

When it asks you to enter number 1, just press ENTER (i.e. a blank entry). Normally this would cause an "error in expression" but the error handler here tells you that you should have entered a number and then directs you back to that statement to re-enter the number. Some care is needed with the layout of the messages - REPORT normally issues a linefeed after the message, so the above program may result in untidy output like:

```
Number 1 › error in expression
 at line 40
You should enter a number silly!
Number 1 ›
```

I leave that to you as an exercise!

One point to note here is that the WHEN ERROR structure cannot as far as I know be used to trap BREAK. You can still CTRL SPACE out of a program which is error trapped, so you don't have to worry about creating a BASIC program which will trap you in an endless loop!

It may have occurred to you that the program may not run correctly on an older ROM version such as JM. I suspect that not a lot can be done about this if the ROM version concerned won't even load a program containing WHEN ERROR and REPORT statements, but if these can be loaded at all then a piece of code like this may help to reduce the ill-effect by letting the program check the ROM version and not execute the code concerned if it is an AH or JM ROM for example:

(I honestly don't know if this will work but it's a useful hint for various other ROM-version dependent issues)

```
100 v$=VER$
110 IF NOT(v$="AH" OR v$="JM") THEN
120   WHEN ERRor
130     REPORT #1,ERNUM
140     PRINT" at line ";ERLIN
150   END WHEN
160 END IF
```

Another useful application of this little trick is when a program needs to access the system variables (OK, I know I'll be lined up against a wall and shot at dawn for suggesting a program messes about with the system variables). Minerva (and, I think SMSQ) has a special version of VER$ which can tell you the base address of the system variables. Most Sinclair ROMs have the system variables at a fixed address in memory, so accessing them from this address should be OK.

```
100 v$=VER$
110 IF v$="JSL1" OR v$="HBA" THEN
120   sysvar_addr=VER$(-2)
130 ELSE
140   sysvar_addr=163840
150 END IF
```

TURBO

The Turbo compiler has a slightly different implementation of WHEN ERROR which although a bit harder to learn at first, it does have a few extra features, is more reliable on different ROMs and in theory at least works on all ROM versions. Turbo uses the WHEN_ERROR keyword (note the underscore between the two words) and this works at two levels. WHEN_ERROR 0 is a 'global' error handler which can be used as a general way of reporting errors and closing programs down more tidily, while WHEN_ERROR 1 is a second level of error trapping which can be varied depending on the routine in use at the time.

Turbo also has the functions ERLIN% and ERNUM% to complement WHEN_ERROR and END_WHEN, but no equivalent to REPORT unless you write your own, using a SELECT ON clause to match short messages up to the various error codes. CONTINUE and RETRY work differently under Turbo. CONTINUE turns off error processing at the level given - WHEN_ERROR 1:CONTINUE:END_WHEN will cancel level 1 error trapping, and WHEN_ERROR 0:CONTINUE:END_WHEN will turn off level 0 error processing. RETRY makes program execution resume at the statement indicated by a RETRY_HERE keyword. It can often be useful to place RETRY_HERE keywords in front of INPUT statements or file accesses.

```
100 IF COMPILED THEN
110   WHEN_ERROR 0
120     PRINT "Oops, due to error ";ERNUM%;"
    at line ";ERLIN%
130     PRINT "this program will now close
    down."
140     CONTINUE
150   END_WHEN
160 END IF
170 REMark a useless little test program
180 CLS
190 IF COMPILED THEN
200   WHEN_ERROR 1
210     PRINT"Enter a number, silly!"
220     RETRY
230   END_WHEN
240 END IF
250 FOR entry=1 TO 10
260   RETRY_HERE:INPUT"Enter number ";(a);" >
    ";number
270 END FOR entry
280 REMark turn off level 1 error trapping
290 IF COMPILED THEN
300   WHEN_ERROR 1
310     CONTINUE
320   END_WHEN
330 END IF
```

This error trapping only works in Turbo compiled programs, so it is a bit hard to test it running the program uncompiled. The best you can do is to surround the Turbo-specific bits with IF COMPILED THEN statements to prevent the interpreter trying to execute them. COMPILED is a Turbo function which returns a value of 0 if a program is running under the interpreter, or 1 if running as a compiled task.

I dislike Turbo's error trapping because it is a bit fiddly to use in my opinion, and difficult to test uncompiled, although it is doubtless a very well thought out and comprehensive system of error trapping for compiled programs.

QLIBERATOR

QLiberator's approach to error trapping is very different to Turbo's system. It has the advantage of working in uncompiled programs, but is non-standard and perhaps leads to bulkier code as you have to write a piece of code for each possible error trap rather than write a general error handler.

It works by converting QL keywords which are written as procedures into functions so that they return error codes which can be tested with the Q_ERR function. Thus, INPUT would work as normal, but if an error occurred the program would not stop, but rather you could test if an error occurred using Q_ERR and take appropriate action.

The command Q_ERR_ON 'command_name' (where 'command_name' is a procedure like INPUT, OPEN, etc) turns on error trapping for that command. Now any error which happens is trapped and Q_ERR will give you a number indicating what went wrong:

```
100 Q_ERR_ON 'input'
110 REPeat input_loop
120   INPUT'Enter a number > ';number
130   IF Q_ERR=0 THEN EXIT input_loop
140   PRINT\'Oops, error ';Q_ERR
150   PRINT'Enter a number, silly!'
160 END REPeat input_loop
170 Q_ERR_OFF 'input'
180 PRINT'Number entered OK'
```

The Q_ERR_OFF statement removes the error trapping from that procedure name and turns things back to normal.

File Acces Error Trapping

As with general error handling, the various compilers and interpreted BASIC differ in their approach to this.

A command as simple as OPEN_IN #3,filename$ can cause a program to stop with an error if there is no disk in the disk drive concerned, or if the file does not exists.
SuperBASIC can use WHEN ERROR to work around this as described above.

```
100 WHEN ERRor
110   PRINT"That file can't be found! ";
120   REPORT #1,ERNUM
130   INPUT"Press ENTER to retry.";z$
140   RETRY
150 END WHEN
160 CLS
170 INPUT"Enter filename › ";f$
180 OPEN_IN #3,f$
190 CLOSE #3
200 PRINT "Managed to open ";f$
```

This little routine would give you a chance to insert the correct disk if an error occurred during the first attempt.

Turbo users have an additional weapon in their armoury. The DEVICE_STATUS function tests a file first and returns a code to tell you if the file is accessible or not. A negative number usually means some form of error occurred while zero or a positive number lets you know it managed to access the file. There are several versions of DEVICE_STATUS all using different parameters, and I don't fully understand them all, but the main thrust is the same:

`PRINT DEVICE_STATUS("filename")` will return an error code you can act on accordingly:

```
100 INPUT"Filename › ';f$
110 file_error_code = DEVICE_STATUS(f$)
120 IF file_error_code ‹ 0 THEN
130   PRINT"Oops, error
    ';file_error_code
140 ELSE
150   PRINT"File accessed OK"
160 END IF
```

One bug-bear with this command is that the 'basic' version does not manage to catch all errors, for example, it doesn't test if you can write to the file, if the disk is write protected and so on. You need to use the more advanced form of the command for that.

QLiberator's approach to file access is exactly the same as that described for INPUT above, just use Q_ERR_ON 'OPEN_IN' (or whatever file command you intend to use) and take appropriate action depending on the value returned by Q_ERR.

```
100 Q_ERR_ON 'OPEN_IN'
110 REPeat file_loop
120   INPUT'Enter a filename › ';f$
130   OPEN_IN #3,f$
130   IF Q_ERR=0 THEN CLOSE #3:EXIT
    file_loop
140   PRINT\'Oops, error ';Q_ERR
160   PRINT 'Could not access ';f$
150   PRINT'Enter a number, silly!'
160 END REPeat file_loop
170 Q_ERR_OFF 'OPEN_IN'
180 PRINT'File accessed OK'
```

Another approach to protected file access is offered by extensions such as the file open functions in Toolkit 2. Functions like FTEST, FOP_IN, FOP_NEW and so on return error codes indicating success or failure of the file operation rather than just stopping the program with an error if an error occurred. This allows the program to take corrective action when errors occur and generally makes programs more robust for users.
The FTEST function works rather like DEVICE_STATUS described above. You give the filename of the file to be tested and it tries to open it for input (sadly it doesn't tell you if you can write to the file) and tells you via the function value returned if the access worked or not:

```
100 INPUT"Filename › ";f$
110 file_error_code=FTEST(f$)
120 PRINT"FTEST returned the value
    ";file_error_code
```

The other five Toolkit 2 functions are

FOPEN	tries to open a file like OPEN
FOP_IN	tries to open a file like OPEN_IN
FOP_NEW	tries to open a file like OPEN_NEW
FOP_OVER	tries to open a new file and if that file already exists, it is overwritten and a new file created in its place
FOP_DIR	Opens a directory on the disk – use this one with care unless you know what you are doing!

These 5 functions can be used in two ways.
If you supply just the filename as a parameter, the functions will try to find an unused channel number and if the file was opened successfully, that positive channel number is returned. If it failed to open a channel to the file, a negative error number is returned.

```
100 INPUT'Filename › ';f$
110 file_error_code=FOP_IN(f$)
120 IF file_error_code<0 THEN
130   PRINT"Error ";file_error_code
140 ELSE
150   PRINT"Channel ";file_error_code;" opened
    to ";f$
160   CLOSE #file_error_code : REMark tidy up
    after use
170 END IF
```

If you supply two parameters, the first is taken to be the channel number you want used to open the file:

```
100 INPUT'Filename › ';f$
110 file_error_code=FOP_IN(#3,f$)
120 IF file_error_code<0 THEN
130   PRINT"Error ";file_error_code
140 ELSE
150   PRINT"Channel 3 opened to ";f$
160   CLOSE #3 : REMark tidy up after use!
170 END IF
```

Obviously, programs written to use the FOP_xxx extensions will only work if Toolkit 2 is present on the system. It is built into just about any QL system worth having these days ***[with the exception of some emulators which can't include it due to obvious copyright issues - Editor]*** so it need not be an issue really.

This is probably the main issue about error trapping - trying to ensure that whatever technique you use is likely to work on all of the systems the program is likely to be used on!

I learned a lot while preparing this article, I hope someone else finds it as useful!

Gee Graphics! (on the QL?) - Part 24

H L Schaaf

Circle packing

The cover picture on the April 21, 2001 issue of Science News was of many circles packed within a circle. So how would we use the QL to create a similar picture? That's the subject of this article and the program listing "CM7k3_bas". 'CM' for 'Circle Math', 7th version, kth modification, 3rd revision.

The cover story tells how statistician Allan Wilks at a 1998 conference in Germany was talking about the homework geometry problem of another colleague's daughter. The problem was to neatly fit another circle into an outer unit circle that already had two identical half-unit circles side by side inside touching each other and the outer circle.

This turns out to be an old problem going back to least the time of Apollonius of Perga (born 262 BC, died 200 BC) who described it in his book "Tangencies". He also wrote 8 books on "Conic Sections" and

```
100 REMark CM7k3b_bas
110 REMark HL Schaaf September 15, 2001
120 REMark for use with GG#24
130 :
140 REMark Science News Vol. 159 Apr 21, 2001 p.254+
150 REMark thanks to Pete Goffinet for the challenge!
160 REMark 4 mutually tangent circles
170 REMark curvatures a, b, c, d  as reciprocals of radii
180 REMark (a*a + b*b + c*c + d*d) = (1/2) * (a + b + c + d)^2
190 REMark use negative value for outer circle
200 REMark given a,b,c  ; find d
210 :
220 REMark example shown in article
230 REMark given outer circle radius 1, curvature -1
240 REMark and two inner circles radius 1/2, curvature 2
250 REMark find 4th circle tangent to all, curvature = 3
260 :
270 WTV : PAPER 0 : INK 7 : CLS
280 MODE 4
290 INPUT "how many circles ? ",num_cir
300 :
310 DIM Cir(num_cir,4)
320 REMark store curvature, x and y centers, radius, circle #
330 REMark   in     0,     1,    2          3         4
340 :
350 REMark a spawn list, which 3 generated the 4th ?
360 DIM spawn%(num_cir,3)
370 REMark have these in order and check for prior usage?
380 REMark use zeroth element for timing
390 :
400 REMark check for tangency between circles
410 REMark could treat as a vector in half the space ?
420 REMark using strictly triangular matrix conversion ?
430 DIM Tn%(num_cir,num_cir)
440 Tn%(0,0) = 2
450 :
460 lastgood$=''  : REMark last good circle string
470 lastcands = 0  : REMark last candidate circle
480 :
490 REMark take input as a positive value and convert to negative
500 INPUT#0, "first (outer) circle curvature ?",a
510 REMark scale so always fills screen
520 y_size = 2.2*ABS(1/a)
530 x_off = -1.6*ABS(1/a): y_off = -1.1*ABS(1/a)
540 x_ctr = 0 : y_ctr = 0 : y_siz = y_size
```

```
550 SCALE y_size, x_off, y_off : CLS
560 REMark provision to 'zoom in' on a region
570 :
580 Cir(0,0)=Cir(0,0)+1 : REMark count = 1
590 Cir(1,0)= -a : Cir(1,1)=0 : Cir(1,2)=0 :Cir(1,3)=-1/a
600 Cir(1,4)= 1
610 INK 244 :FILL 1
620 CIRCLE 0,0,-1/a
630 FILL 0 :INK 7
640 AT 12,60:PRINT 1
650 :
660 INPUT#0, "second (inner) circle curvature ?  >"&a,b
670 REMark center on x = (1/a) - (1/b), y = 0
680 REMark center on x-axis and tangent at 3 o'clock
690 Cir(0,0)=Cir(0,0)+1  : REMark count = 2
700 Cir(2,0)=b : Cir(2,1)=(1/a)-(1/b) : Cir(2,2)=0 :Cir(2,3)=1/b
710 Cir(2,4)=2
720 INK 2 : FILL 1
730 CIRCLE ((1/a)-(1/b)),0,1/b
740 :
750 REMark now figure minimum curvature for 3rd circle
760 min_3 = 1/((1/a) - (1/b))
770 INPUT#0, "third (inner) circle curvature ? >="&min_3,c
780 REMark start clock for timing
790 startime = DATE
800 REMark solve for three mutually tangent circles
810 REMark one outer w/negative, two inner w/positive
820 s1=1/b +1/c : s2 = ABS(1/a) - 1/b : s3 = ABS(1/a) - 1/c
830 alph_arg = ((s2*s2 + s3*s3 - s1*s1)/(2*s2*s3))
840 IF ABS(alph_arg)>1:  alph_arg = SGN(alph_arg)
850 alph = ACOS(alph_arg)
860 x3 = COS(alph)*s3 : y3 = -SIN(alph)*s3
870 Cir(0,0)=Cir(0,0)+1   : REMark count = 3
880 Cir(3,0)=c : Cir(3,1)=x3 : Cir(3,2)=y3 :Cir(3,3)=1/c
890 Cir(3,4)=3
900 INK 4 : FILL 1
910 CIRCLE x3,y3,1/c
920 INK 7 : FILL 0
930 FOR i = 2 TO 3
940  CURSOR Cir(i,1),Cir(i,2),-5,-5
950  PRINT i
960 END FOR i
970 :
980 REMark find the 4th circle that fits
990 fourth_circles Cir(1),Cir(2),Cir(3)
1000 REMark in case we get 5, pick the largest
1010 IF Cir(0,0)= 5 THEN
1020  IF Cir(5,0) < Cir(4,0) THEN
1030   INK 242 : FILL 1:CIRCLE Cir(4,1),Cir(4,2),Cir(4,3):FILL 0
1040   CURSOR Cir(5,1),Cir(5,2),-5,-5: PAPER 0: INK 7 :PRINT 4
1050   FOR i = 0 TO 3
1060    Cir(4,i) = Cir(5,i)
1070    Cir(5,i) = 0
1080    spawn%(5,i)=0
1090   END FOR i
1100   Cir(0,0) = Cir(0,0)-1
1110  END IF
1120 END IF
1130 check_tangency
1140 acir = 4
1150 pack_circles
1160 :
1170 AT 0,0 : PRINT Cir(0,0)
1180 PRINT #0;'touch [space bar] for curvatures '
1190 PAUSE :CLS #0
1200 show_curvatures
1210 PRINT #0;"touch [spacebar] for list of curvatures ";
1220 PRINT #0;" & centers, 20 at a time"
1230 PAUSE : CLS #0: CLS
1240 list_curve_ctrs
1250 PRINT #0;'any key for ctr2ctr'
1260 PAUSE :CLS #0 :CLS
1270 redraw_circles
1280 ctr2ctr
1290 PAUSE 300
```

is credited with the first use of the expressions 'ellipse', 'parabola', and 'hyperbola'.

In 1643 Rene Descartes described a general solution with "Cartesian" geometry in a letter to his student, Princess Elizabeth of Bohemia. Descartes expressed the size of the circles in his formula by their curvature (reciprocal of the radius).

Wilks found the problem intriguing and with fellow workers at Bell Labs found new and interesting mathematical patterns and relationships and developed new theorems and formula, extending them into hyperbolic and spherical packings as well.

Book VIII of Apollonius is missing, as well as "Tangencies", so could it be that Wilks et al are just rediscovering things that might have been known to the ancients?

Wilks' team had Bell Labs computers, matrix math, complex number systems, etc. available and they used them. Let's see what we can do with the QL.

In order to pack in order with next largest circle possible, we build a list of candidates, then SORT them. This means we need a way to sort them. LRESPR "SORT" from the Quanta Library before running "CM7k3_bas".

INPUT the number of circles wanted. Try 19 to start with. You are then asked for the curvatures of the first 3 circles. You can try 1, 2, and 3 as a starter set. In a few seconds you should have the picture. Touch the [spacebar] until you get to the little menu at the bottom. Try the different options in various orders. Enjoy!

JOCHEN MERZ SOFTWARE

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

QPC2 Version 2

We have told you about the new features many times in advance: they are all fully functional: QPC2 is now running inside a Windows window (but, of course, full screen can be used as well). You have access to 65536 colours etc. Not to forget the DOS device. You can now read and write from and to DOS and Windows harddisks. DOS1_ = C:, DOS2 = D: and so on!

See review previous issue!

To avoid confusion: If you already own QPC2 (Version 1), then the upgrade is NOT free!. The naming is a bit confusing (we should have called it QPC3, but now it is too late, isn't it?), so here is the complete price list:

Upgrade to QPC2 Version 2 Final

From QPC1 (or just QPC - the first QPC!)	£41.90
From QPC2 Version 1 (or just QPC2)	£25.90
From QPC2 Version 2 any Pre-Release or Version 2 full release	FREE!

You can get free Updates in the JMS BBS (+49 203 502013 and 502014)
You will also find the free Update on the Internet: http://qpc.j-m-s.com

TERMS OF PAYMENT

Postage and package **[Germany]** DM 8,99 (if total value of goods is up to DM 50,- then only DM 5,99).
[Europe] £4,50 (if total value of goods is under £15 then only £3).
[Overseas] between US$7.50 (1 item) and US$17.50 (maximum).
All prices incl. 16% V.A.T. (can be deducted for orders from non-EU-countries).
Cheques in DM, EURO, International Money Order in DM or EURO (they all have to be drawn on German banks, Eurocheques and Credit Cards accepted. Please note that Eurocheques will not be valid anymore in 2002!
Prices are based on an average exchange rate of £1 ranging between DM 3,10 and DM 3,20.
Prices may be adjusted in case the exchange rate falls out of this range - in both directions!
We can charge your credit cards in £'s, US$, EURO or in DM - please state the currency you prefer.
Please do not send any UK bank cheques in £ - our banks have increased the fee for handling them by 600% (no joke!) so we cannot accept them anymore, unless you add £6 for clearing the cheque.
E&OE.

```
1300 number_centers
1310 PRINT #0;"any key for options menu" : PAUSE
1320 CLS # 0 : options_menu
1330 STOP
1340 :
1350 REMark the 'main engine'
1360 DEFine PROCedure pack_circles
1370  REPeat runarnd
1380   check_tangency
1390   cand$='' : REMark candidate list
1400   cand$=lastgood$ : REMark candidate list
1410   candidate$ = ''
1420   cands = lastcands
1430   last_cir = acir - (Cir(0,0) = 4)
1440   acir=Cir(0,0) : REMark active circles ?
1450   AT 0,0 : PRINT Cir(0,0);' of ';num_cir
1460   FOR per1 = 1 TO acir
1470    AT 11,5 :PRINT (per1);'  '
1480    FOR per2 = per1 +1 TO acir
1490     IF NOT(Tn%(per2,per1)):  NEXT per2
1500     AT 12,5 : PRINT (per2);'  '
1510     FOR per3 = last_cir TO acir
1520      IF NOT(Tn%(per3,per2)) : NEXT per3
1530      AT 13,5:PRINT (per3);'  '
1540      IF Tn%(per1,per2) AND Tn%(per1,per3) AND Tn%(per2,per3) THEN
1550       DIM soper(3)
1560       soper(1)=per1:soper(2)=per2:soper(3)=per3
1570       SORT soper ,1
1580       candidate$="#"&soper(1)&','&soper(2)&','&soper(3)
1590       cand$ = cand$&candidate$
1600       cands = cands + 1
1610      END IF
1620      IF (Cir(0,0) > num_cir) : EXIT runarnd
1630     END FOR per3
1640    END FOR per2
1650   END FOR per1
1660   REMark now to simplify the candidate string ?
1670   good$ = ''
1680   REPeat duplicates_out
1690    s1="#" INSTR(cand$)
1700    IF (s1<>1) THEN
1710     good$=good$&cand$
1720     EXIT duplicates_out
1730    END IF
1740    s2 = 1 + "#"INSTR(cand$(2 TO))
1750    IF s2 = 1 THEN
1760     good$ = good$ & cand$
1770     EXIT duplicates_out
1780    END IF
1790    test$= cand$(s1 TO (s2-1))
1800    match = test$ INSTR(cand$(s2 TO))
1810    IF match THEN
1820     cand$=cand$(s2 TO)
1830     cands = cands-1
1840     ELSE
1850     good$ = good$&test$
1860     test$ = ''
1870     cand$ = cand$(s2 TO)
1880    END IF
1890   END REPeat duplicates_out
1900   :
1910   cand$=good$&'#'
1920   DIM cand(cands,3):DIM cantag%(cands)
1930   FOR i = 1 TO cands
1940    DIM num(3)
1950    sh1 = '#' INSTR cand$
1960    com1 = ','INSTR cand$
1970    com2 = com1 + ','INSTR cand$(com1+1 TO)
1980    sh2 = 1 +'#'INSTR cand$(sh1+1 TO)
1990    num(1) = cand$(2 TO com1-1)
2000    num(2) = cand$(com1+1 TO com2-1)
2010    num(3) = cand$(com2+1 TO sh2-1)
2020    FOR j = 1 TO 3
2030     cand(i,j)=num(j)
2040    END FOR j
```

Wilks' cover picture has about 405 circles, and the QL with SGC and SMSQ/E takes less than a half-hour to work them all out. In order to see the smaller details, use the Zoom option from the menu, they're all there!

For those with internet access: website for Science article on Circle Math
http://www.sciencenews.org/20010421/bob18.asp
updated version of "Homogenous Coordinates" The Visual Computer 1994 Jules Bloomenthal's website
www.unchainedgeometry.com/jbloom/ papers/index.html

For future articles? There must have been some changes to SMSQ/E since the Graspix_bas exercise (GG#3, March/April 1998). It shows up in the MODE 4 testing when BLOCKS and PIXELS don't always match up any more. This also shows up in the later versions of QPC2. I hope to look into this, and also get back to the use of matrices and homogenous coordinates in graphics.

A few comments regarding the layout

Herbs listing is fairly long this time. As we know that some of our readers type the listing in or read them with interest, we try to make their life a bit easier (as requested). First, the listings are on the **outer** side of each page. This makesit easier to keep the magazine opened while typing. We have not reduced the font size too much, and we have not printed in two columns sideways. I hope this is OK - if we can do anything else to help you reading or typing, please let us know.

COMPETITION

Recently on the QL Users Email Mailing List a short dialogue took place, initiated by Norman Dunbar I think. The idea was to pursue something Roy Wood had said in QL Today before, namely make up witty acronyms or whatever out of well known computer names or terms. In order to add a bit of fun to QL Today, I asked Jochen if he'd stump up a prize for the wittiest entry (well, I didn't give him much choice actually). The competition means you have to come up with something like the letters QL standing for Quaint Lump or Quite Late. Per Witte on the mailing list from The Netherlands came up with the letters of my name (DILWYN) standing for Devil Is Living in Wales You kNow, which would hardly win the prize but made me smile nonetheless. Anything I've seen before such as in Roy's list or which are unprintable will be disqualified. Entries which make me laugh will win marks. Entries which belittle other computing systems will also gain a few points. Personal insults will lose marks.
I will judge this one myself and my decision is final. (Got that Jochen and Bruce?)

Some examples of possible targets to aim for:
DILWYN, BRUCE
JOCHEN MERZ
SINCLAIR
QDOS (Quite Decidedly OS Supreme – thanks Per Witte!)
SMSQ, MSDOS
LINUX, WINDOWS
AURORA
SGC

Looking forward to spending a long time judging all the entries. Hopefully, you all want a software voucher from JMS!

```
2050    cand$=cand$(sh2 TO)
2060   END FOR i
2070   :
2080   REMark figure 4th curv
2090   FOR i = 1 TO cands
2100    d1 = curvs4(Cir(cand(i,1),0),Cir(cand(i,2),0),Cir(cand(i,3),0))
2110    cand(i,0)=d1
2120   END FOR i
2130   AT 15,5: PRINT cands
2140   REMark sorted candidates = socan
2150   DIM socan(cands)
2160   FOR i = 1 TO cands
2170    socan(i)=cand(i,0)
2180   END FOR i
2190   I_FILL cantag%, 0, 1
2200   SORT socan ,1 , cantag%
2210   nc = 1
2220   low_val = cand(cantag%(nc),0)
2230   REPeat new_circs
2240    cn = cantag%(nc)
2250    fourth_circles Cir(cand(cn,1)),Cir(cand(cn,2)),Cir(cand(cn,3))
2260    IF Cir(0,0)=num_cir :EXIT runarnd
2270    nc = nc + 1
2280    IF cand(cantag%(nc),0)>low_val : EXIT new_circs
2290   END REPeat new_circs
2300   lastgood$=''
2310   FOR i = nc TO cands
2320    lastgood$=lastgood$&'#'
2330    FOR j = 1 TO 3
2340     lastgood$=lastgood$&cand(cantag%(i),j)&','
2350    END FOR j
2360   END FOR i
2370   lastcands = cands-nc+1
2380  END REPeat runarnd
2390 END DEFine pack_circles
2400 REMark end of 'main engine'
2410 :
2420 DEFine PROCedure fourth_circles (cir1, cir2, cir3)
2430  LOCal a, b,c,d1,d2
2440  new_found = 0
2450  d1 = curvs4 ( cir1(0), cir2(0), cir3(0) )
2460  d2 = curv4b
2465  FOR d_1_2 = d1, d2
2470  REMark now try to place it ! solve for x,y
2480  REMark cir4 & cir3 are tangent c2c = line A
2490  s1 = 1/d_1_2 + cir3(3) :REMark side a
2500  REMark other sides are r2+r3 and r2 + 1/d_1_2
2510  s2 = cir2(3) + cir3(3) : s3 = cir2(3) + 1/d_1_2 :REMark sides b & c
2520  REMark given 3 sides can find angle A
2530  Arg_A = (s2*s2+s3*s3-s1*s1)/(2*s2*s3)
2540  IF Arg_A < -1 : Arg_A = - 1
2550  Ang_A = ACOS(Arg_A)
2560  Ang23 = RAD(angl_frm (cir2(1),cir2(2),cir3(1),cir3(2)))
2570  x = 0 : y = 0
2580  xa = cir2(1)+(s3*COS(Ang23-Ang_A))
2590  ya = cir2(2)+(s3*SIN(Ang23-Ang_A))
2600  xb = cir2(1)+(s3*COS(Ang23+Ang_A))
2610  yb = cir2(2)+(s3*SIN(Ang23+Ang_A))
2620  DIM ctr_used(2)
2630  DIM cir4a(3) : DIM cir4b(3)
2640  cir4a(0)=d_1_2 :cir4b(0)=d_1_2
2650  cir4a(1)=xa :cir4b(1)=xb
2660  cir4a(2)=ya :cir4b(2)=yb
2670  cir4a(3)=1/d_1_2 :cir4b(3)=1/d_1_2
2680  :
2690  REMark check for previous usage
2700  FOR ci = 1 TO Cir(0,0)
2710   REMark another way to check?
2720   REMark distance from xa,ya to center of circle cir(ci,1),cir(ci,2)
2730   REMark compared to radius of cir(ci,3) ?
2740   IF dist_btwn(xa,ya,Cir(ci,1),Cir(ci,2)) < Cir(ci,3) THEN
2750    ctr_used(1)=ctr_used(1) + 1
2760   END IF
2770   IF (xa+1) == (Cir(ci,1)+1) AND (ya+1) == (Cir(ci,2)+1) THEN
2780    ctr_used(1)= ctr_used(1) + 2
2790   END IF
```

```
2800   IF dist_btwn(xb,yb,Cir(ci,1),Cir(ci,2)) < Cir(ci,3) THEN
2810    ctr_used(2)= ctr_used(2) + 1
2820   END IF
2830   IF (xb+1) == (Cir(ci,1)+1) AND (yb+1) == (Cir(ci,2)+1) THEN
2840    ctr_used(2)=ctr_used(2) + 2
2850   END IF
2860  END FOR ci
2870  REMark check for meeting with cir1,2,3
2880  chk1a =  tan_cir(cir4a,cir1)
2890  chk2a =  tan_cir(cir4a,cir2)
2900  chk3a =  tan_cir(cir4a,cir3)
2910  chksa = (chk1a AND chk2a AND chk3a AND NOT(ctr_used(1)))
2920  IF (chk1a AND chk2a AND chk3a AND NOT(ctr_used(1))) THEN
2930   x = xa : y = ya
2940  END IF
2950  chk1b =  tan_cir(cir4b,cir1)
2960  chk2b =  tan_cir(cir4b,cir2)
2970  chk3b =  tan_cir(cir4b,cir3)
2980  chksb = (chk1b AND chk2b AND chk3b AND NOT(ctr_used(2)))
2990  IF (chk1b AND chk2b AND chk3b AND NOT(ctr_used(2))) THEN
3000   x = xb : y = yb
3010  END IF
3020  IF (chksa OR chksb) THEN
3030   REMark we have a valid addition
3040   Cir(0,0)=Cir(0,0)+1
3050   Cir(Cir(0,0),1)=x  :Cir(Cir(0,0),2) = y
3060   Cir(Cir(0,0),0)=d1 :Cir(Cir(0,0),3)=1/d1
3070   Cir(Cir(0,0),4)=Cir(0,0)
3080   INK  Cir(0,0) : FILL 1
3090   CIRCLE Cir(Cir(0,0),1),Cir(Cir(0,0),2),Cir(Cir(0,0),3)
3100   FILL 0 : INK 7
3110   IF NOT Cir(Cir(0,0),0) THEN
3120    INK 7
3130    CIRCLE Cir(Cir(0,0),1),Cir(Cir(0,0),2),Cir(Cir(0,0),3)
3140   END IF
3150   CURSOR Cir(Cir(0,0),1),Cir(Cir(0,0),2),-5,-5
3160   PAPER 0: INK 7 :PRINT Cir(0,0)
3170   new_found = new_found+1
3180   spawn%(Cir(0,0),1)=cir1(4)
3190   spawn%(Cir(0,0),2)=cir2(4)
3200   spawn%(Cir(0,0),3)=cir3(4)
3210   SORT spawn%(Cir(0,0))
3220   spawn%(Cir(0,0),0)= DATE - startime
3230  END IF
3240  END FOR d_1_2
3910 END DEFine fourth_circles
3920 :
3930 DEFine FuNction tan_cir(cir1,cir2)
3940  REMark center to center distance
3950  c2c = dist_btwn (cir1(1),cir1(2),cir2(1),cir2(2))
3960  REMark sum of radii
3970  sumrad = ABS(cir1(3) + cir2(3))
3980  IF (ABS(sumrad - c2c) < 5E-4) THEN
3990   REMark 5E-4 just a hunch, there may be a better value ?
4000   RETurn 1
4010  ELSE
4020   RETurn 0
4030  END IF
4040 END DEFine : REMark FuNction tan_cir(cir1,cir2)
4050 :
4060 REMark angl_frm
4070 DEFine FuNction dist_btwn(xpt,ypt,x,y)
4080  REMark distance between two points xpt,ypt as point of origin
4090  xdis = (x-xpt) : ydis = (y-ypt)
4100  sqdist = ((xdis*xdis)+(ydis*ydis))
4110  IF sqdist > 0 THEN
4120   dbtw = SQRT(sqdist)
4130   ELSE
4140   dbtw = 0
4150  END IF
4160  RETurn dbtw
4170  RETurn xdis
4180  RETurn ydis
4190 END DEFine :REMark FN dist_btwn(xpt,ypt,x,y)
4200 :
```

The Letter Box

Richard W Parker jr. writes from Florida:

Thanks to you, Dilwyn, Bruce and all the team for a great magazine.

There was some criticism in the last issue, Jul/Aug, on your repeating articles or subject matter. I don't agree with that at all. I've been QL-ing since '83 and love the machine, but for the last 10 years I've been retired in a lonely, QL-less world.

Most of your repeat articles are timely as they remind me of something I had forgotten or give me a new aspect. I especially like looking at other boot programs and keep improving mine from the ideas and skill of others.

I was forced onto a Windoze machine last year to stay on the net and the Q40/Q60/QPC discussion is very interesting. QL Today is really my lifeline to the QL world and to a feeling of community.

Frank Gutteridge writes:

I have "progressed"? from ZX81 through QL through THOR I through Mega/STE to AMD PC with QPC, as well as IBM Thinkpad with ditto, so it cannot be said that I have any particular regard for platforms. I entirely agree with Roy Wood that one can ignore the silly semantics about platforms. Indeed, when one reads the series of articles in QLtoday for July/August 2001, it is clear that what the authors like is not so much the platform as the operating system.

I have felt for some time that QDOS and SMS have reached a dead end. While they remain pleasant to use and do provide an outlet for tinkerers, difficult on other systems (how

many Windows users can quickly program to write to screen a little window saying "Hullo World"?), these systems do not provide the things that everyone expects today, such as Internet, recording to CD, etc. It is unlikely that new major software will ever come, so that users will inevitably find themselves in a slowly declining market that will in the end become irrelevant.

Readers of "Electronics World" and "The Scientific American" will recently have come across articles on Linux, with reasons for switching from Windows to Linux now that GUIs have made the latter more approachable, and the use of Linux for the Beowulf project for creating super computers out of computer farms made up of old PCs linked together through Ethernet. Apparently commercial firms are becoming interested in the Beowulf project, which would give a

```
4210 DEFine FuNction angl_frm(xf,yf,xt,yt)
4220  REMark angle in degrees from origin(xf,yf) to (xt,yt)
4230  REMark —› = 0´ or 360´, @ = 90´, ‹— = 180´,   = 270´
4240  IF dist_btwn(xf,yf,xt,yt) = 0 THEN
4250   PRINT #0;"same spot!" :PAUSE 20:spolangle = 0
4260  END IF
4270  qdx=1 :qdy = 2
4280  IF ydis‹0 : qdy=qdy+2
4290  IF xdis‹0 : qdx=qdx+2
4300  qprd = qdy*qdx  : qsum =qdy+qdx+qprd
4310  quadA = ((INT((qsum+1)/3))/2)
4320  quadB = (quadA-INT(quadA))*5
4330  quad = quadB+quadA  :REMark Trig quadrant I, II, III, or IV
4340  IF dbtw‹›0 THEN
4350   sinrat = ydis/dbtw
4360   IF sinrat ›  1 THEN sinrat =  1
4370   IF sinrat ‹ -1 THEN sinrat = -1
4380   sangle = DEG(ASIN(sinrat))
4390   SELect ON quad
4400    = 1 :spolangle = sangle
4410    = 2 :spolangle = 180-sangle
4420    = 3 :spolangle = 180-sangle
4430    = 4 :spolangle = 360+sangle
4440    = REMAINDER : PRINT #0;"quad error arcsine":STOP
4450   END SELect
4460  END IF
4470  :
4480  angle_from = spolangle
4490  RETurn angle_from
4500  RETurn quad
4510 END DEFine  :REMark FN angl_frm(xf,yf,xt,yt)
4520 :
4530 DEFine PROCedure SWAP (n1,n2)
4540  n1 = n1 + n2
4550  n2 = n1 - n2
4560  n1 = n1 - n2
4570 END DEFine SWAP
4580 :
4590 DEFine FuNction CYC (Number%,cycle_length%)
```

```
4600  REMark cyclic modular with option base = 1 vs option base 0
4610  REMark Number% is an integer number
4620  REMark  cycle_length% is the Length of the cycle
4630  RETurn ((Number%-1) MOD cycle_length%)+1
4640 END DEFine CYC
4650 :
4660 DEFine FuNction SGN(n)
4670  RETurn (n>0) - (n<0)
4680 END DEFine SGN
4690 :
4700 DEFine FuNction MAX(a,b)
4710  RETurn a*(a>b)+b*(b>a)+a*(a=b)
4720 END DEFine
4730 :
4740 DEFine FuNction MIN(a,b)
4750  RETurn a*(a<b)+b*(b<a)+a*(a=b)
4760 END DEFine
4770 :
4780 DEFine FuNction curvs4(curv1,curv2,curv3)
4790  LOCal qa,qb,qc,qdet,sum_sq,sum_curv
4800  sum_sq = curv1*curv1 + curv2*curv2 + curv3*curv3
4810  sum_curv  = curv1 + curv2 + curv3
4820  qa = 1
4830  qb = -(2*sum_curv)
4840  qc = (2*sum_sq) - (sum_curv*sum_curv)
4850  qdet = qb*qb - (4 *qa * qc)
4860  IF qdet < 1E-6 : qdet = 0
4870  IF (qc) THEN
4880   curv4a = 2*qc/(-qb-SQRT(qdet))
4890   curv4b = 2*qc/(-qb+SQRT(qdet))
4900  ELSE
4910   curv4a = (-qb + SQRT(qdet))/(2*qa)
4920   curv4b = (-qb - SQRT(qdet))/(2*qa)
4930  END IF
4940  RETurn curv4a
4950  RETurn curv4b
4960 END DEFine
4970 :
4980 DEFine PROCedure check_tangency
4990  LOCal i,j
5000  FOR i = 1 TO Cir(0,0)
5010   FOR j = Tn%(0,0) TO Cir(0,0)
5020    IF tan_cir(Cir(i),Cir(j)) THEN
5030     Tn%(i,j)=1 : Tn%(j,i) =1
5040    END IF
5050   END FOR j
5060  END FOR i
5070  Tn%(0,0)=Cir(0,0)-1
5080 END DEFine check_tangency
5090 :
5100 DEFine PROCedure options_menu
5110  PAPER #0,0 : INK#0, 4 :CLS#0
5120  PRINT #0;'1 = redraw circles,  ';
5130  PRINT #0;'2 = number circles,  3 = curvature'
5140  PRINT #0;'4 - list curvatures and centers,  5 - spawn & timing'
5150  PRINT #0;'6 - path of centers,  7 - Zoom,  ';
5160  PRINT #0;'8 - STOP,  9 - RUN,  0 - CLS'
5170  INPUT#0;"Select number and then ENTER", choice
5180  SELect ON choice
5190   = 0 : CLS
5200   = 1 : redraw_circles
5210   = 2 : number_centers
5220   = 3 : show_curvatures
5230   = 4 : CLS# 0 :list_curve_ctrs
5240   = 5 : CLS: show_spawn
5250   = 6 : ctr2ctr
5260   = 7 : Zoom_in
5270   = 8 : CLS# 0 : STOP
5280   = 9 : RUN
5290   = REMAINDER
5300  END SELect
5310  CLS#0 : options_menu
5320 END DEFine options_menu
5330 :
5340 DEFine PROCedure redraw_circles
```

singular boost to Linux. Linux is now offered by IBM, Compaq and Dell and applications are being produced by companies such as Oracle and Dell. Indeed one can imagine that many users running big programs and being frustrated by crashes in Windows and NT might well switch to Linux for its comparative stability.

Linus Thorvalds and his supporters did not waste their time in endless discussions over hardware, nor did they consider that some particular computer was to be treated as something sacred, the abandonment of which would amount to apostasy. They wrote amongst others for the most popular and widespread platform which, like it or not, is the PC. Many complaints are made against this machine, particularly with regard to crashes, but I think that this is unfair, because it is Windows/DOS which is the cause of the crashes rather than the machine itself, as is evidenced by the relative reliability of PCs running under Linux.

My feeling for some time now has been that imitating Linux, SMS should be made Open Source under a protective consortium and rewritten in the first instance for PCs. My understanding has always been that SMS is both UNIX and POSIX compatible, so that it should then be possible to run UNIX software, including Internet access. It should not be very difficult to adapt the existing QL/SMS software and eventually, I suppose Windows software could run under APIs (Application Protocol Interfaces), as suggested in "Electronics World" for Linux. SMS, with its "atomic" approach, is unique and its inherent stability should prove even more attractive than either UNIX or Linux

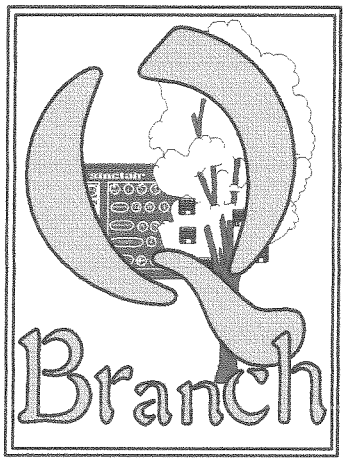
My further feeling is that those who at present supply SMS and the relevant utilities would benefit from such a scheme because, rather than operating in a technical and operational backwater, they would come into the mainstream and enjoy vastly increased demand; even their present prices would not seem unreasonable for sales under an open system and they would of course have an advantage over new outside competitors because of their intimate knowledge of the system. Evidently "SMS" would have to be replaced as a name, this acronym, as for others on the QL scene, such as QXL, having been "stolen" by the mobile phone industry and the Internet. Even Microsoft should be pleased, because, with Linux, SMS would offer increased competition, thus diminishing the risk of lawsuits by the monopoly authorities. Another point is that open source SMS would be initially, at any rate, a European initiative and thus looked on kindly by the European Union authorities and perhaps also by a number of users in Europe who would like a system closer to home. Finally, even if an open source project for SMS failed to generate interest and further development, no one would be any worse off than they are at present.

All I can add is that if open source SMS came on the market, I would immedialy buy it. This note was written in T87 under QPC and SMSQ.

Editor: Quite an interesting point. Maybe a good start for another themed issue - not necessarily as hot as the previous one! Readers, do you think that open source would generate more interest?

```
5350  CLS
5360  FOR i = 1 TO Cir(0,0)
5370   SELect ON i
5380    = 1 : INK 242
5390    = 2 : INK 2
5400    = 3 : INK 4
5410    = REMAINDER : INK i
5420   END SELect
5430   FILL 1
5440   CIRCLE Cir(i,1),Cir(i,2),Cir(i,3)
5450   FILL 0
5460  END FOR i
5470 END DEFine redraw_circles
5480 :
5490 DEFine PROCedure number_centers
5500  PAPER 0 : INK 7
5510  CURSOR -Cir(1,3),0,50,-5: PRINT '   '
5520  FOR  i = 1 TO Cir(0,0)
5530   CURSOR Cir(i,1),Cir(i,2),-5,-5 : PRINT i
5540  END FOR i
5550 END DEFine number_centers
5560 :
5570 DEFine PROCedure show_curvatures
5580  PAPER 0 :INK 7
5590  CURSOR -Cir(1,3),0,50,-5: PRINT Cir(1,0)
5600  FOR i = 2 TO num_cir
5610   CURSOR Cir(i,1),Cir(i,2),-5,-5: PAPER 0: INK 7 :PRINT Cir(i,0)
5620  END FOR i
5630 END DEFine show_curvatures
5640 :
5650 DEFine PROCedure list_curve_ctrs
5660  REMark analysis of coordinates and curvature
5670  REMark do they give integer values ?
5680  PAPER 0: INK 7 :CLS
5690  FOR i = 1 TO num_cir STEP 20
5700   FOR j = 0 TO 19
5710    IF i+j > num_cir :EXIT i
5720    bx = Cir(i+j,1)*Cir(i+j,0)
5730    REMark use 2^-24 ? or some larger value such as 1E-6?
5740    IF ABS(bx)< 2^-24 : bx = 0
5750    by = Cir(i+j,2)*Cir(i+j,0)
5760    IF ABS(by)< 2^-24 : by = 0
5770    PRINT i+j, Cir(i+j,0),bx,by
5780   END FOR j
5790   IF num_cir - (i+j) >= 20 THEN
5800    PRINT #0;'touch [spacebar] for next 20'
5810   ELSE
5820    PRINT #0;'touch [spacebar] for last ';num_cir - (i+j)
5830   END IF
5840   PAUSE :CLS #0
5850  END FOR i
5860 END DEFine list_curve_ctrs
5870 :
5880 DEFine PROCedure show_spawn
5890  PAPER 0 : INK 7 : CLS
5900  FOR i = 4 TO num_cir STEP 20
5910   CLS #0
5920   FOR j = 0 TO 19
5930    IF (i+j)>num_cir : EXIT j
5940    PRINT spawn%((i+j),1 TO),'- >',(i+j),
5950    PRINT 'in ';spawn%(i+j,0);' seconds'
5960   END FOR j
5970   IF (i+j) < num_cir THEN
5980    PRINT #0;'touch [spacebar] for more'
5990   ELSE
6000    CLS#0 : options_menu
6010   END IF
6020   PAUSE
6030  END FOR i
6040 END DEFine show_spawn
6050 :
6060 DEFine PROCedure ctr2ctr
6070  INK 7
6080  POINT Cir(1,1),Cir(1,2)
6090  FOR  i = 2 TO Cir(0,0)
```

Q Branch



PROGRAMMING

QD 98	£ 45.00
QD + QBasic	£ 59.00
QD + Qliberator + QBasic	£ 100.00
Qliberator	£ 50.00
Master Spy v 3.3	£ 30.00
QPTR	£ 30.00
Easyptr pt 1 & 2 (together)	£ 30.00
Easyptr pt 3 (C library)	£ 14.00
QMake	£ 15.00
QMon / JMon	£ 22.00
Basic Linker	£ 19.00
DISA 3	£ 31.00
QMenu	£ 14.00

Text 87

£ 79.00

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

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

Q s Autumn Bargains

We Have:
3 Full superHermes @ £ 60.00 each
1 QXL II 8Mb @ £110.00 each
2 x Qplane backplanes @ £ 4.00 each
There may also be a couple of Aurora motherboards available by the time of going to press.
We have been advertising the Epson Stylus 880 colour inkjet printer for sale. At the moment these do seem to be in short supply but there may be some available soon so call us if you want one. The best colour printer for all QL systems.
£ 99.50 + shipping

'Just Words' by Geoff Wicks

THESAURUS, STYLE CHECK

£ 10.00 ea / ANY 2 PROGRAMS £ 18.00 / ALL 3 PROGRAMS £ 25.00
(Includes Pointer and non-pointer driven versions)
(P.E. versions need Hot_rext, WMAN and PTR_GEN or SMSQ/E to run)
Upgrades from previous versions £ 2.50 + S.A.E. New Manuals £ 1.50

QL2PC

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

Only £ 10.00 *Now with HTML support !*

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

NEW PROGRAM !
QL Rhymes £ 10.00
Rhyming Dictionary

NEW PROGRAM !

UTILITIES

FiFi 2	£ 18.00
QSup	£ 28.00
QSpread v2.04	£ 48.00
Cueshell 2	£ 15.00
Qload / Qref	£ 15.00
Disk Mate 5	£ 16.50
QPAC 1	£ 20.00
QPAC 2	£ 40.00
QTYP 2	£ 30.00
QLQ	£ 28.00

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

The SBASIC / SuperBASIC Reference Manual

The complete definitive guide to BASIC programming in QDOS / SMSQ including three disks of PD toolkits, example procedures and an electronic index.
compiled by Rich Mellor, Franz Hermann and Peter Jaeger

Over 500 pages !
£ 40.00
+ postage

Tel +44 (0) 1273-386030 fax +44 (0) 1273-381577
Mobile +44 (0) 7836-745501
email qbranch@qbranch.demon.co.uk web : http://www.qbranch.demon.co.uk

Q Branch

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

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

Please Note our new address and phone numbers

ProWesS

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

Paragraph

The ProWesS word processor

¶ Demo version £ 1.50 + postage
Full Registered version £ 18.00

Version 2.03 available now !

Hardware

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

QXL II	**£ 100.00**
Recycled superHermes	£ 65.00 *
Recycled Gold Card	£ 50.00 *
Recycled Aurora	£ 75.00 *
Qubide	£ 55.00
Qplane	£ 25.00
Aurora cables	£ 3.00
Aurora rom adaptor	£ 3.00
'Arfa Braquet'	£ 8.00
'Son of Braquet'	£ 18.00
The 'Braquet'	£ 16.00
MC plate	£ 6.50

* when available.

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

SMSQ/E

Gold Card / Atari / QXL Version

£ 69.00

Various Atari versions : call for details

QPC 2 full colour version!

£ 82.00 (£ 67.00 SMSQ/E Owners)

Upgrades from QPC 1 £ 35.00 (return master disk)
Upgrades from QPC 2 v1 £ 30.00
Special offer !

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

Q Branch Programs

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

The Fractal Collection !

This is a brand new program which will produce stunning animated fractal patterns. It will run on anything from a Gold Card to the Q40 and will be capable of using the power of the colour drivers when they are released. Complete with many example files and routines to design your own screens.

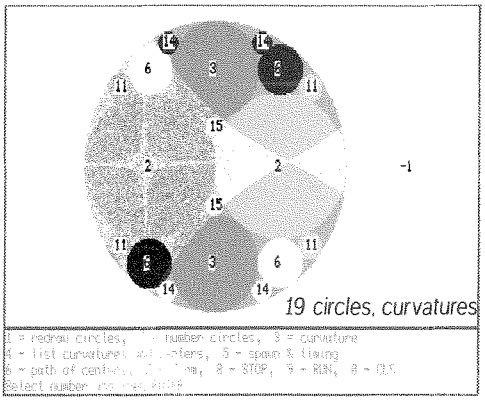
Only £ 35.00

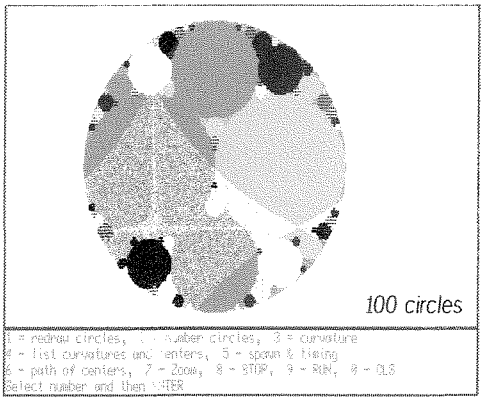
QPC 2 v2 FULL VERSION AVAILABLE NOW

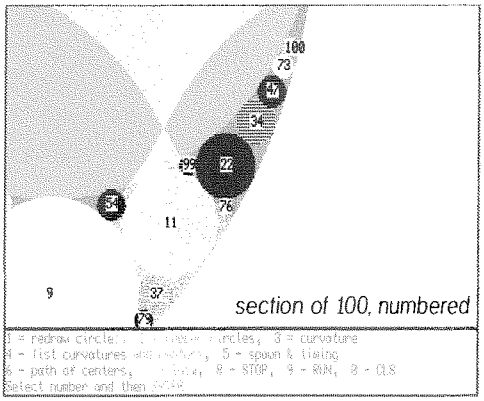
Qubide upgrades to version 2.01 £ 8.00

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

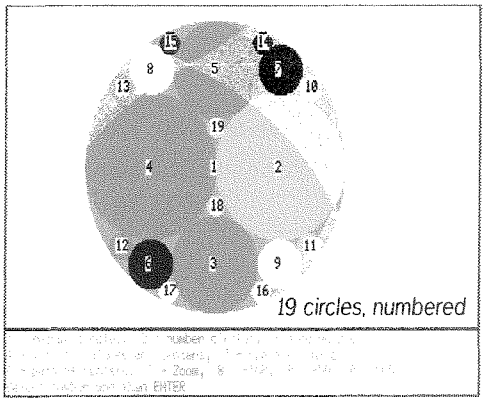
```
6100   LINE TO Cir(i,1),Cir(i,2)
6110  END FOR i
6120 END DEFine ctr2ctr
6130 :
6140 DEFine PROCedure Zoom_in
6150  OVER 0 : CLS#0 : redraw_circles
6160  OVER -1  : INK 242
6170  CIRCLE x_ctr,y_ctr,y_siz/4
6180  CIRCLE x_ctr,y_ctr,y_siz/5
6190  PRINT #0;'use arrows to put rings around region of interest'
6200  PRINT #0;'with SHIFT key for large movement'
6210  PRINT #0;'with CONTROL key for small movement'
6220  PRINT #0;'touch [space bar] when selected'
6230  REPeat indicate_region
6240   move_it = CODE(INKEY$(-1))
6250   CIRCLE x_ctr,y_ctr,y_siz/4
6260   CIRCLE x_ctr,y_ctr,y_siz/5
6270   SELect ON move_it
6280     = 32 : EXIT indicate_region
6290     = 192 : x_ctr = x_ctr - y_siz/64
6300     = 194 : x_ctr = x_ctr - y_siz/256
6310     = 196 : x_ctr = x_ctr - y_siz/8
6320     = 200 : x_ctr = x_ctr + y_siz/64
6330     = 202 : x_ctr = x_ctr + y_siz/256
6340     = 204 : x_ctr = x_ctr + y_siz/8
6350     = 208 : y_ctr = y_ctr + y_siz/64
6360     = 210 : y_ctr = y_ctr + y_siz/256
6370     = 212 : y_ctr = y_ctr + y_siz/8
6380     = 216 : y_ctr = y_ctr - y_siz/64
6390     = 218 : y_ctr = y_ctr - y_siz/256
6400     = 220 : y_ctr = y_ctr - y_siz/8
6410     = REMAINDER
6420   END SELect
6430   CIRCLE x_ctr,y_ctr,y_siz/4
6440   CIRCLE x_ctr,y_ctr,y_siz/5
6450  END REPeat indicate_region
6460  CIRCLE x_ctr,y_ctr,y_siz/4
6470  CIRCLE x_ctr,y_ctr,y_siz/5
6480  OVER 0 : CLS #0
6490  PRINT #0;' negative numbers zoom out to cover larger area'
6500  PRINT #0;' positive numbers zoom in to enlarge details'
6510  PRINT #0;' enter 0 to restore to original size'
6520  INPUT #0,'Zoom factor ? ',zoof
6530  SELect ON SGN(zoof)
6540   = 1 :  y_siz = y_siz / ABS(zoof):
6550   = -1 : y_siz = y_siz * ABS(zoof):
6560   = 0 : y_siz = y_size : x_ctr = 0 : y_ctr = 0
6570   = REMAINDER
6580  END SELect
6590  SCALE y_siz, -.75 * y_siz + x_ctr, -.5 * y_siz + y_ctr  :CLS
6600  redraw_circles : CLS# 0 : options_menu
6610 END DEFine Zoom_in
6620 :
6630 REMark end of listing CM7k3b_bas
```

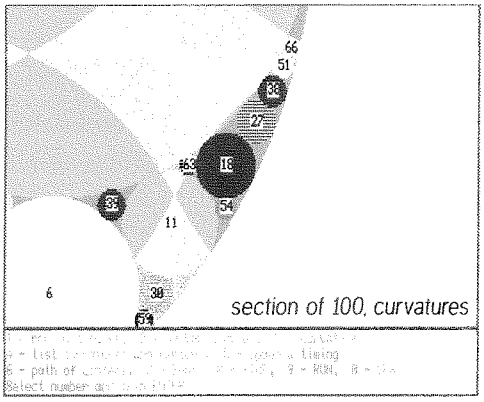






Finally, a few screenshots to give you an idea how the results will look like:





Rubber emails

Tony Firshman

I have just gone through an emailshot experience I would like to forget.
Before the QL 2000 bash last year, we snailshot some 2000 people. We thought it about time that the QL mailing list was weeded down to current QL people. We were extremely disappointed at the small number of postcards returned. However we asked for email addresses, and got a considerable number. I also had quite a few on my own database.
Good I thought – we now have 440 QLers with emails – this will ensure we can afford to tell more people about QL shows etc.
What a foolish idea.
I sent a test email, and immediately got some 50 bounced emails, and very often ISP autoresponders 'forget' to say what address failed. This is worse than it sounds, because a lot of ISPs try dozens of times, and send boring reports of their attempts.
OK – so I remove these 80 (less a handful which I got updates).
I then send the main mailing for three forthcoming shows (Byfleet, S Germany and Paris).
I then get ONE HUNDRED bounced emails!
I found later that a small number of these addresses were OK – but were bounced anyway.
.... so the QL mailing list is now down to a touch over 300.
In order to keep me sane, would people who have received the emailshot, please keep me up to date with their email address.
Also anyone who did not receive the circular and wants it, please let me know their email address.
tony@firshman.demon.co.uk
Emailing is not all it is cracked up to be.

Spam or information?

Jochen Merz

Well, everybody using an email account seems to suffer more and more from this. It is getting more and more annyoing.
However, when Tony Firshman sent out information regarding the forthcoming QL shows he told me he got a very rude reply. Although I think that nobody who gave us his/her email addresss, for example on the postcard last year, wants to be spammed it is pretty clear that information regarding QL shows together with information regarding the dealers attendance, addressed to the group of people which ***is*** interested in QL matters (and not to 10,000,000 "bought" email adresses, hoping that 0.0001% will we interested).

If you don't like that, tell us, and you will be removed. You are speaking to real persons you know, not anonymous spammers.
And now for a bit of fun! Although I just asked you to reply ***to us***, don't do it to real spam. I found some very funny texts while searching for spam matters, here is an extract of what the phrases in spam email really mean. If you want more, go to
http://www.lodz.pdi.net/~eristic/ junkmail/

It is called
Spamfordian != English
an abbreviated dictionary of Spamspeak

absolutely guaranteed
(1) ...or your money back; but first you'll have to find us (see From:; Received:)
(2) ...to make us a stash, if we can get away with it.

free
(1) Mine
(2) Free to me

From:
From anywhere but:
usage: standard smtp header field

legitimate business
scam, fraud; shady, dubious or illegal dealings

make
lose
usage: "make money fast"

May I have your permission to send you...
See if I care.

only
First (of a sequence)
usage: "this is the only mail you will receive"

Received:
Forcefully shoved up the communications pipes of: ‹victim›
usage: standard smtp header field

remove
Add

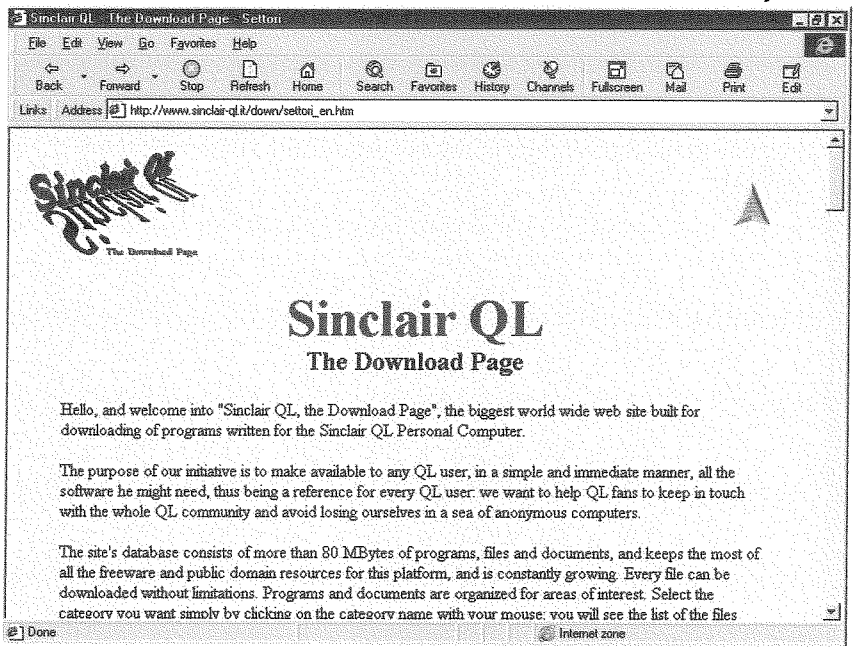
Sinclair QL The Download Page

by Giorgio Garabello

Hello, and welcome to "Sinclair QL, the Download Page", the biggest world wide web site built for downloading of programs written for the Sinclair QL Personal Computer:
www.sinclair-ql.it

The purpose of our initiative is to make available to any QL user, in a simple and immediate manner, all the software he might need, thus being a reference for every QL user: we want to help QL fans to keep in touch with the whole QL community and avoid losing ourselves in a sea of anonymous computers.
The site's database consists of more than 80 MBytes of programs, files and documents, and keeps most of the freeware and public domain resources for this platform, and is constantly growing.
Every file can be downloaded without limitations. Programs and documents are organized by areas of interest. Select the category you want simply by clicking on the category name with your mouse: you will see the list of the files available for downloading, along with a description of the files, their dimensions and the creation date of the software.
You can also upload new software: just ask about this. The only limitations are:
1. the software to be uploaded must be somehow related to the Sinclair QL
2. the software to be uploaded must not be commercial or illegal software (demo versions of commercial software are allowed to be uploaded)
3. upload of pornographic, offensive or somehow illegal material is strictly forbidden

Only the site mantainers can decide if something that has been uploaded will be inserted into the site.



To ask about software uploading, send an e-mail message to the address
webmaster@sinclair-ql.it
attaching, along with the software, any kind of related documentation and (if available) source code and examples.

IMPORTANT - DISCLAIMER

Site mantainers are NOT responsible for either the contents of the downloaded files, nor the use you will make of it, whether legal or illegal. This is not a commercial activity, but a not-for-profit initiative: access to the site and software downloading is free to anyone, and nobody is authorised to ask for money for any reason.
In the category listing below, (active) means that this section was active at the time of writing.

Categories	Description
Terminal	Terminal emulation programs for the most frequently used standard (active)
Compressors	Compression / decompression programs (active)
BBS managing	BBS managing programs for QL (active)
FAX managing	Useful programs for receiving, sending and printing FAX sheets (active)
Printer managing	Programs to configure and manage the most known printers (active)
Off-line readers	Programs to read BBS messageboard while being off-line (active)
Editors	Program editors for the Sinclair QL (active)
XCHANGE and utilities	The Xchange package, accessories and related utilities (active)
Games	QL Games (active)
Graphic	Graphic programs and viewers (active)

Categories	Description
Graphic	Format conversion
Graphic	format conversion program (active)
Screensaver	Screen saving programs for the Sinclair QL (active)
Benchmark	Benchmark programs (active)
Utility	Various utilities (active)
Utility - Floppy	Utilities/data recovery prorams for floppy disk
Utility - Hard disk	Utilities/data recovery prorams for hard disk
Utility - SuperBASIC	SuperBASIC programming utilities
Utility - Sistema	Utilities to better take advantage of the QDOS capabilities
Utility - C68	Support programs for the C68 compiler (active)
Utility - Zyxel	Zyxel modems support utilities
Utility - Text 87	Utilities for the word processor Text 87
Utility - QXL	Utilities for the QXL board
Utility - Text conversion	Text format conversion programs
Extension	Extension to the QL operating system
Demo programs	Limited version of commercial programs (active)
Languages - C68	The C68 compiler: binaries, source and documentation
Languages - ANSI Forth	The ANSI Forth compiler
Languages - APL	The APL languages for QL active
Languages - Prolog	The Prolog programming language for QL (active)
Languages - Assembler	680x0 The Assembler language for the 680x0 family active
Languages - Intercal	The Intercal language for the QL (active)
Languages - ReXX	The Regina ReXX language for the QL (active)
Languages - Perl	The Perl languages for the QL (active)
Languages - Scheme Lisp	The Scheme Lisp languages for the QL
Languages - C (old)	Others C compilers (old programs)
Languages - others	Others languages
Emulators	Emulation program for the QL
Documentation	Technical documentation, manuals and examples
Documentation - Prowess	Prowess and Proforma documentation
Magazine - Qitaly magazine	The whole collection of the Qitaly Magazine (closed) (active)
Magazine - QHJ	The whole collection of the QL Hacker Journal (active)
Magazine - QL magazine	The whole collection of the QL Magazine
Clip art	Clip art collection and images, for use as wallpapers or the like (active)
BBS listing	BBS program listing
Other platf. - Emulators	QL emulation programs under other platforms
Other platf. - Utility	Utility programs for the QL under other platforms
Other platf. - Cross comp.	QL Compilers for other platforms (active)

This site is owned by a free group of QL italian users.

HP Printers & QL's (Part 2)

by Peter Fox

I have noticed that I did not use the shift in defining the last character in Underline Off in Part 1 of my article. The decimal equivalents are correct but not the character itself which should have been ':'.

Another thing that can be done with the 'printer_dat' file is to insert translates to produce both bold and underlining in Abacus.

The way this is done is to choose an obscure character from the character set to start bold and a second to stop bold and then repeat the process for underlining characters. My obscure characters are ‹CNTL›‹1› to ‹4› for Bold On and Off and Underline On and Off respectively.

Bold On

```
'ê'+'ESC'+'('+'s'+'3'+'B'
145,27,40,115,51,66
```

Bold Off

```
'ï'+'ESC'+'('+'s'+'0'+'B'
146,27,40,115,48,66
```

Underline On `'i'+'ESC'+'&'+'d'+'1'+'D'`
`147,27,38,100,49,68`

Underline Off `'i'+'ESC'+'&'+'d'+'@'`
`148,27,38,100,64`

With these translates within your printer_dat, insertion of the correct character or characters will produce bold or underlined characters subject to one proviso: All of the characters in these translates are non-printing characters so that, especially in Abacus, your formatting will go astray by a character or two.

In my previous article, I mentioned fixed and floating underlining. Floating underlining does not underline spaces but fixed does. To give two examples:

The quick brown fox jumped over the stile
The quick brown fox jumped over the stile

The first of these examples is a fixed underline and the second is a floating underline. Whilst Underline Off is the same in every case the Underline On changes:

Single Fixed `'ESC'+'&'+'d'+'1'+'B'`
Double Fixed `'ESC'+'&'+'d'+'2'+'B'`
Single Floating `'ESC'+'&'+'d'+'3'+'B'`
Double Floating `'ESC'+'&'+'d'+'4'+'B'`

I shall not give the decimal equivalents of the whole sequence, simply to say that:

decimal equivalent of ASCII '1'	49
decimal equivalent of ASCII '2'	50
decimal equivalent of ASCII '3'	51
decimal equivalent of ASCII '4'	52

so that substitution of the appropriate 'digit' will achieve the desired effect.

Most HP printers have a carriage width limited to A4 portrait which amounts to 210mm. Unfortunately DL envelopes (the common variety are 220mm wide which means that the only way they can go through an HP printer is in landscape mode. This has the effect of slowing up the printer and restricting the fonts available. This will depend on the actual printer itself.

Portrait Mode `'ESC'+'&'+'l'+'0'+'O'`
`27,38,108,48,79`
Landscape Mode `'ESC'+'&'+'l'+'1'+'O'`
`27,38,108,49,79`

Therefore if you wanted to reset the printer and then set it into Landscape mode, you would need to combine the commands for these instructions as follows:

Reset + Landscape
`'ESC'+'e'+'&'+'l'+'1'+'O'`
`27,101,38,108,49,79`

In my previous article, I mentioned that combining printer commands can be done provided that only the last character in the command is upper case. The printer will ignore any characters after an upper case character.

In my next part I will tackle the subject of sub- and superscripts

Editor: On the transfer from Text87 to Calamus, this DTP program, the £ character in the last sentence of the previous article got lost. Sorry about that.

BYFLEET 2001

Dilwyn Jones

Quanta held its autumn workshop in the Byfleet village hall on Sunday 23rd September. This started slowly, but became quite a well attended show in the end, leading Geoff Wicks for example to declare that he'd had an excellent day's trading there. Geoff himself had launched a new product there, QL-Rhymes, a Rhyming Dictionary program. This program will be useful for budding poets and anyone needing to use Rhymes.

Thankfully, it was a well attended show, possibly because the Byfleet show was postponed last year due to the staging of the QL 2000 event, in order to promote it. However Byfleet returned this year in fighting form. And all the major QL traders were there as usual. Entrance was via the side door from the car park, and first on entering the door to the right was Q-Celt Computing and Darren Branagh. Darren had his usual collection of CD's and Disks, QL 2nd User Hardware and Software, Cables and Peripherals. He launched some products and some updates - namely the ZeXcel Spectrum Emulator CD-ROM for the QL, which debuted at last depite a lot of production problems! In fact, this proved to be his best seller at the show - he sold out of all his stock and had to promise to send on additional CD's to some customers!! Also debuting at the show was the

RWAP SOFTWARE

QL Cash Trader v3.7 £5

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

QL Payroll v3.5 £5

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

Q-Help v1.05 £10
Q-Index v1.04 £5

Q-Help: on-screen help for SuperBASIC commands, including TK2, Turbo Toolkit, SMSQ/E and PD toolkits. Can be used to add help to your own programs - simply produce ASCII text for each help page, add an index and Q-Help automatically cross-references and displays the links.
The PD toolkits referred to are available for £2.
Q-Index: The SuperBASIC index supplied with the Reference Manual - enter a topic such as 'screen resolution' and find out the commands which relate. Launch Q-Help for further info on the chosen command.

Sidewriter v1.08 £10

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

ProForma ESC P2 Drivers v1.03 £8

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

QL Genealogist v3.26 £20
Genealogy For Windows £50

Store your family tree for posterity. Add individuals with details of their parents and children, watch all of those links build up into a formal family tree layout. Text files and pictures may also be linked to individuals as well as notes and events, making this the perfect way to preserve the history of your family. QL version now supports FileInfo II and QMenu as well as allowing you to link both male and female trees. Sample tree of the Royal family since 1066 included. PC version is event driven - enter the details as they appear in documents and it generates the tree from these. QL data and GEDCOM can be transferred to the PC version. Upgrade to latest PC version (v5.21) for £8 Both programs easy to use and complete with a step by step tutorial.
** QL USERS upgrade to PC version for £25 ONLY **

D-Day MKII v3.04 £10
Grey Wolf v1.8 £8
War In The East MKII v1.24 (Upgrade Only) £5

For the gaming enthusiast - D-Day is a classic table top wargame for one or two players - you control either the Allies or the Axis forces during WWII. With the ability to define your own army set ups and a choice of 4 different scenarios, this should keep you entertained for a while.
Grey Wolf is a graphical simulation of a submarine - can you sink the enemy shipping whilst avoiding their planes and destroyers??

Image D v1.03 £10

Produce graphical representations of 3D objects - view them as wireframe, hidden line and shaded. Perspective and magnification can be controlled and views can be saved to file for subsequent printing. Multiple objects can be defined and positioned relative to each other. Simple to use yet produces excellent results.

SBASIC SuperBASIC Reference Manual £40
Updates £6 each, £10 for 2 (Current Version - Rel 3)

Have you ever tried to write a program, but been lost as to the means of performing a certain action? This Reference Manual provides you with a full description and examples of how to use all of the keywords found on each of the different QLs, plus SMSQ/e, Toolkit II and many different public domain toolkits. Details of any possible problems are provided, together with descriptions of how to use the device drivers and how to ensure that your programs are compatible across the range of QL platforms.
This book is ideal for all QL users and is kept up to date with regular updates.
Orders are currently being taken for the next print run of this popular tome.
(Note: Price for the book does not include post & packing).

QL Cosmos v2.04 £5

Ever wondered what the stars in the sky looked like 100 years ago? Or, maybe you want to learn the constellations and names of what you see in the sky. This is the program for you - generates pictures of the stars and planets for any given place or time and provides details on these objects. Includes Halley's Comet, the Moon and the Solar System planets.

Q-Route v2.00 £25
Upgrade from v1.xx £5

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

Flashback SE v2.03 (Upgrade only) £5

The ultimate database program - extremely fast and flexible, easy to use, updated to cope with the latest versions of the QL operating system and still maintained. A report module is included to allow you to format output in any way, including mail-merge. Unfortunately only available as an upgrade from the original version (original still available from Sector Software).

Return To Eden v3.08 £10
Nemesis MKII v2.03 £8
The Prawn v2.01 £8
Horrorday v3.1 £8
West v2.00 £5
The Lost Kingdom of Zkul v2.01 £5

A wealth of QL adventures - mainly text only.
Save the Galaxy from the ambitions of the evil dictator Nemesis.
Battle against werewolves and dracula look-alikes on a Hammer Horror set in the comical Horrorday.
Take the part of a prawn with a hangover, lost in a strange land in the hilarious Prawn.
Solve a bank-robbery by fighting the bad guys and collecting the loot in real-time old West.
Battle countless dwarves in the atmospheric Lost Kingdom of Zkul.
Return to Eden is a massive adventure over 3 disks with colourful graphics - control 3 characters in their quest to find the missing Prince.
All six adventures are available together for only £25.

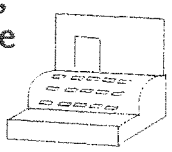
Open Golf v5.20 £8
QuizMaster II v2.07 £5
Stone Raider II v2.00 £5
Hoverzone v1.2 £5
Deathstrike v1.5 £5
Flightdeck v1.05 £10

A range of games to keep both the young and the young at heart amused. Some are old favourites, like Golf and a pub quiz program (500+ questions). Others are fast, colourful arcade games. Flight simulator also now available. Plenty of variation and skill required - what more can you ask for? All 6 programs only £28.

Britain.map v1.11 £2
BIG Britain Map (needs 2MB) v2.03 £5
Various Britain Area Maps (ask for details) £2 ea.
Ireland Map v1.00 £5
Belgium Map v1.01 £2
Catalonia Map v1.03 £2

The latest maps for Q-Route. Maps of various areas of Britain have been created by cutting them out of Big Britain Map - they will use less memory and can contain more detail. Areas covered: Scotland, NE England, NW England, S&W Yorkshire, Wales & Derbyshire, London area and South England.
Latest version of Q-Route is recommended.

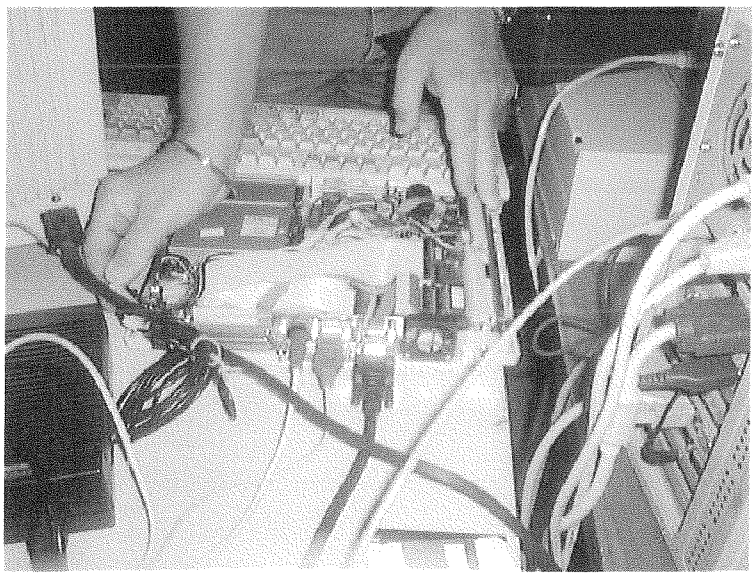
RWAP Software, 7 Common Road,
Kinsley, Pontefract, West Yorkshire
WF9 5JR
TEL: 01977 614299



Cheques in £sterling payable to 'R.Mellor'

* Also known as Trading Accounts

Phoebus Dokos Interactice Fantasy collection, which contains several hundred Infocom Zip adventures ported to the QL from the PC world. If you are a text adventure enthusiast, this is the one for you. A contender for the most expensive QL emulator ever made an appearance on the Q-Celt stand, courtesy of Ken Brickwood's new Apple iBook running Daniele Terdina's QemuLator Lite and displaying its video output on a Hitachi AV-laptop (the world's first truly portable VCR recorder with built in TFT LCD screen, circa 1989 vintage) – if you know of a more expensive QL system let us know, the combined value today of these two items would be about £4,500 we guessed!



Now Darren, I know you are Irish, but the shade of green on that sweatshirt was a bit much. Even your Irish girlfriend Sandra said the same thing.

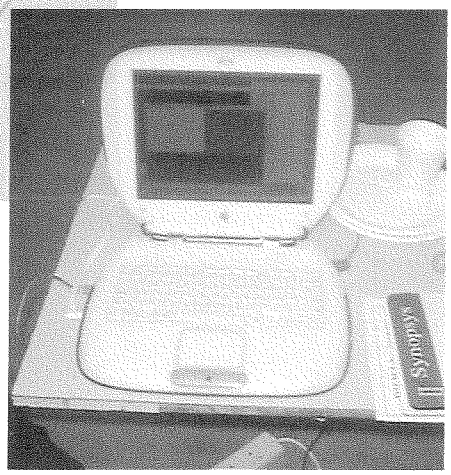
Alongside Darren I launched my new PD software library service. Phil Jordan of The Library has not had enough time on his hands to be able to run The Library to his satisfaction, so I am starting my own service up to copy freeware QL software on disk for QLers not lucky enough to have internet access to get hold of the wide range of free QL software available out there. The launch was a bit of a mess, unfortunately, as problems with the laptop computer I was using at the show meant I was unable to copy or print anything until late afternoon. Still, lesson learned.

Quanta's stand was in front of the stage, just below the bring and buy stall, which was well stocked with a variety of older and harder to obtain items. Most of the Quanta officials were there, including the newest committee member, John Gilpin, who was tucked away between Pandora cased QLs working on Quanta library material. There was a lot of interest in the Quanta library CD, and they also took a copy of the QL Emulators CD-ROM to see if there was enough interest among Quanta members to supply the material on this CD to members. John's wife Sarah was also canvassing opinion on whether the Quanta AGM to be held in Manchester next April should be a one or two day event – one suggestion was that it be a one and a half day event in effect, giving the traders a chance to drive there saturday morning, a half day workshop in the afternoon, stay overnight and then a full meeting and AGM on the Sunday – QL2002 perhaps? If you have views on this, let John and Sarah Gilpin know, as they are organising the event.

Dave Walker (of C68 compiler fame) had his own table there and impressed everyone with his ultra light ultra thin Fujitsu-Siemens Lifebook laptop, then proceeded to spoil it all by lifting up a huge bag containing plug in CD drives, floppy drives, Jazz drive (I think) etc etc. I thought the whole idea of having a lightweight laptop was not to have to carry all these plug in bits and pieces, Dave? Reminds me of my extra wide QL back in the 1980s with plug in bits hanging off every side of it!

Tony Firshman's stand was a

project, in favour of the 680x0 series of chips, now that he has located a cheaper source of these, so he seems to be regaining interest in the Goldfire card, although by his own admission he will have to reconsider the name if he moves away from the Coldfire chips to, say, a 68060 chip.



hive of activity as usual, although most of the activity seemed to be coming from his son Ben's computer alongside. Not quite sure what Ben was up to but the level of activity seemed to imply a great future in computers for this lad (hopefully on QLs!). Whilst Tony had no major new products on sale, he did hint at the possibility of a new version of the Qubide IDE interface for the QL. Although not fully confirmed at the moment, he was in discussion with Qubide designer Nasta concerning the specifications and so on, ideas included the possibility of an on-board Compact Flash or other flash memory card slot, a connector for 2.5 inch IDE hard drive cables and a redesigned, lower noise printed circuit board. The one thing he seemed to be struggling with was a name for the new board, although one name suggested was QLIDE (QL IDE). This product is in its infancy and may well change before launch. Nasta has recently also mooted the idea of moving away from the Coldfire chip for his eagerly awaited Goldfire

Roy Wood had the QBranch stand packed with all the usual QBranch goodies. He had cased Aurora systems on sale for about £180.00 – some of these had brand new desktop cases and had quite reasonable capacities of hard drives on them and so on and looked like quite a bargain as he was selling them off to clear. Like many, Roy had been struggling with cabling for connecting two computers with Sernet. Author of Sernet Bernd Reinhardt happened to be at the workshop and was forced to look for a needle in a haystack to find a suitable serial cable to connect two computers together to prove to Roy it was possible. Eventually, Tony Firshman came to the rescue and made a cable, from there on it didn't take Bernd long to convince Roy.

Jochen Merz is usually too busy copying upgrade disks at these shows to get near to for conversation. His stand included back issues of QL Today, printer ink, cheap floppy disks and very cheap CDRs, along with the usual JMS software. Although not quite finished, I realised by the end of the afternoon he had this whole issue (except for this article) of QL Today ready on his laptop at the show.

Bill Richardson seemed to have a quieter than usual day at the show. Although he had plenty of Z88 and Spectrum products at the show, they did not seem to be selling that well. Bill is still interested in obtaining Spectrum interfaces etc for an overseas client – if you know of a pile of Spectrum products available for sale, Bill may be interested in hearing from you.

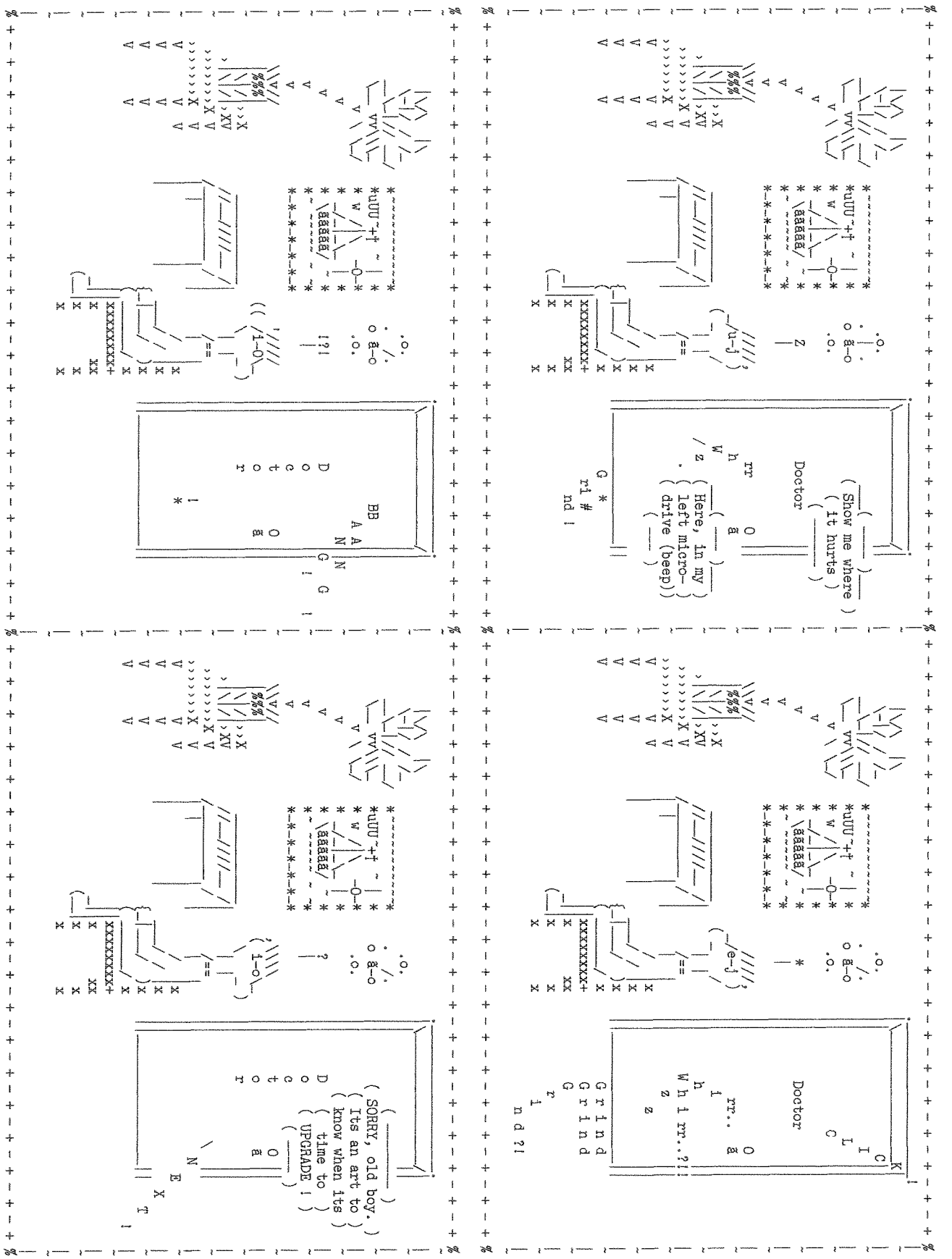
Sadly at this workshop, there was none of the usual talks and lectures which are a normal feature of Quanta workshops. Perhaps not too unexpected, although it was far from being a quiet show, it was steady most of the day rather than extremely busy and there was little by way of major new product announcements. An enjoyable workshop, if not a spectacular one.

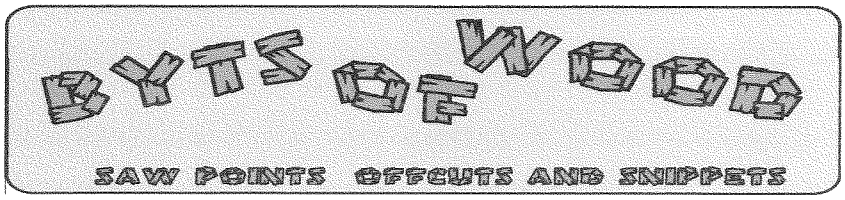
HELP!

QL Today needs a reviewer for the Geoff Wicks QL-Rhymes program. A specialised program like this needs someone who knows what is required of a program like this and not afraid to say if it doesn't do what it should! If you think you can help, contact the editors. You won't get paid, but you will get to keep the program!

At the Doctor

Per Witte





It has been a couple of months of experimentation here at Q Branch Towers. I became very interested in the problems with the newer versions of SMSQ/E for the Q 40 and the patch for the older version and ProWesS. In the last issue Jerome Grimbert says that the patch works fine on his system but I find ProWesS unstable with it running. In my last column I mentioned the new version of SMSQ/E for the Q 40 and the fact that ProWesS will not run under that either. I also mentioned the fact that, if you want to run ProWesS and use the patch program you cannot turn the COPYBACK CACHE on until after ProWesS is loaded.
After I sent the copy for that column to Jochen for publication I found out, purely by a chance mention of the COPYBACK extensions, that the new version of SMSQ/E for the Q 40 (v2.99) has the COPYBACK CACHE turned on. There would seem to be no way to turn this off in this version. This may account for some of the problems I have been having with v 2.99 of SMSQ/E for the Q40

By co-incidence it also means that if I had had the COPYBACK extensions running in the Q 40 when I ran the test that so irked Peter Graf I would probably have had problems running LINEdesign at all so the whole thing would have been pointless.
I have found that turning the cache on after loading ProWesS does speed things up a bit but it also makes LINEdesign lock up when printing.

Joachim van der Auwera replied to my emails about this and said that ProWesS would have problems with these caches so it would seem that the problem would rest there. I will keep my ear to the ground on this. As I have said before I rely on ProWesS for a large amount of the work I do on the Q 40 so I am stuck using an earlier version until there is a way around it. Rest assured I will keep you all informed.

And is There a Manual With This Wheel Thingie?

Wolfgang Lenerz asked, in his comments in the 'big debate', why I had written the QPAC 2 series. The truth is that this was something I wrote some years ago and gave away free with copies of QPAC 2. It was a selfish move really because I often sold the program to customers who then could not get it working. I have often thought that the way the suite was presented was its fatal flaw and that it should have had a standard BOOT file loading and running its main features with the tutorial on a button. That was why I wrote the manual extension myself and gave away a boot file on a disk with it. Jochen asked me if he could publish it in the magazine and I was happy for him to do so.
It was only after I started to give the tutorial away with QPAC 2 that other people came up to me and said that they had never got QPAC 2 working and had given up. I let them have copies of the tutorial and they soon had working versions of QPAC 2 as well. The problem with some software is that it is so hard for the user to get started. There is a tendency to think 'this is not for me', throw it into the box and forget about it.
This has meant that a lot of people bought QPAC 2, in my view the single most indispensable program for all QDOS/SMSQ systems, and either never used it or used only part of it without gaining its full benefit. When a program is finished the hardest thing to do is to write the manual. The QPAC 2 manual never really showed it's real potential. I hope that my contribution cleared some of the fog from the windows.

One of Life's Little Joys

One of the things which has kept me using and supporting the QL, even though it is really financially unviable, has been the number of really enjoyable people I have been introduced to through it. Most of these started out as customers and have now become firm friends. A long time ago, before Q Branch ever existed, I wrote something in Quanta about portable QLs. In those days the cost of a small flat screen was really high and the technology was not really there to make such a thing possible but I got a reply from Roger Godley which really spiked my interest. Roger is an Englishman who lives in Spain and is a long time QL tinkerer and user. He sent me pictures of QLs he had gutted and rebuilt which were very interesting indeed. Since then Roger has visited me at the Portslade Q Branch HQ on many occasions. He usually comes over once a year, in the summer, and spends the summer months travelling around with his two sons. He visits other QL Users

and, usually, manages to attend one of our local user group meetings.
He always has something to show me and has a pretty unique view of things. The conversations usually goes along the lines of 'You know so and so program... well I liked that but I did not like the way it did this so I patched it...' He has a whole library of patched and doctored programs all of which do strange and interesting things. Not only this but he has a collection of hybrid QLs which are all self contained units. This summer he brought two QLs welded together with zip drives and hard drives built into them. I have been trying to get him to write a bit about his activities for the magazine and I hope that he takes us up on the offer.
Roger is one of those people for whom many of the arguments about native hardware are very pertinent. He has a deep understanding of the code at the heart of Motorola 68x chips and the machine code on which QDOS runs. On his last visit he bought Jochen's QDOS Reference manual, a book about which he had not heard before. For those of you who are in a similar position it is a complete and up to date guide to all of the keywords, calls and traps in both QDOS and SMSQ/E - an indispensable bible for machine code programmers.
I have often asked Roger to release some of his stuff to the public either commercially or in the libraries. Maybe this will give him the confidence to do so.

Can the Black Box be Recovered?

While we are on the theme of native hardware Q Branch passed a milestone this month when we sold our last QL Keyboard membrane. I believe that Bill Richardson does not have any left now and that T.F. Services have stopped selling them, keeping the few that remain for QLs that come in for repair.

This is a solemn state of affairs because there is very little hope of any more being manufactured now and the little black boxes which are still in use, therefore, have a limited life unless a superHermes is fitted. This is sad in many ways because, whatever opinion you may have of the original QL, it is the reason that we are all here. I have one very good specimen at the Q Branch HQ which still gets used when I have to format DD disks and a couple of others which I am in process of fixing. I have to treat the membranes with extra care now.
When Tony Firshman looked into the possibility of getting more made the minimum quantity that the company would consider was high enough to make the project unfeasible. When you think that there has to be a template made and a machine specially set up to run the membranes off you can see how high the cost would be. If anyone out there knows of a small company that might be able to make these in low quantities do let us know - a project for Quanta maybe?

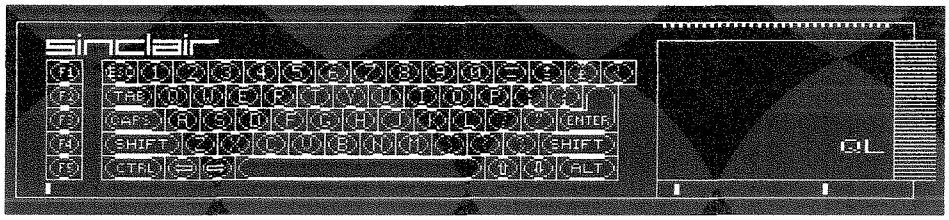
DD Ho Hum

Another item on the endangered list is the DD disk which is now very hard to find and highly priced in relation to its higher capacity brother, the HD disk, when you can get hold of them. QL Today has a small horde of these for the cover disks but, when they are exhausted, as they will be one day, we may have to resort to HD disks for the covers. There are still people who only have DD drives and they are not able to access HD disks. Not only that but, as many of you know, the older interfaces such as the Trump Card and everything except the Gold and Super Gold Cards will not see these disks either.
[Editor - JMS has a large stock of new, brand DD disks, in packs of 10, which will last for many, many issues of QL Today. They can also be purchased in packs of 10 from JMS at reasonable prices]

Winter is Coming but our Drives Have No Jumpers

HD drives are fairly cheap although they are not, these days, supplied with jumpers to allow drive selection. If you do want to replace your DD drives with HD ones and need to set up flp1_ and flp2_ with a pair of modern drives the way to do it is to put a twist into the cable that feeds the drive that you want to flp1_. The PC twisted cable will not work for this (what a surprise!) but the twist is simple to do.
Here then is my 'Teaching Grandmothers to Suck Eggs' section for this month. All you know need to know to make a drive cable.
If you have a PC floppy cable with three connectors on it you can cut the last plug off using a sharp modelling knife leaving you with a cable which can be adapted for use on a QL.
It is worth noting here that, although most PC hardware is approaching rocket science with fast processors, memory and drives, the humble cable has remained firmly in the stone ages. Most new cables come with the old connector on them for the 5.25" drive which was obsolete when the

DILWYN JONES



QL P.D. SOFTWARE LIBRARY SERVICE

A brand new PD software library service from QL author Dilwyn Jones. Hundreds of freeware, shareware and PD programs available on floppy disk at a price of just £1.00 per disk (or just 75 pence if you supply the floppy disk). Programs are normally supplied on HD disks unless you specifically request DD disks.

CATALOGUE

To obtain a free catalogue of the entire software library:

1. Send me a formatted HD or DD disk and return postage, or
2. Download the catalogue from my website (see address below), or
3. By email – just send me an email to ask for the catalogue as a text file.

CD-ROMs

(Also available from Q-Celt Computing in Ireland)

QL EMULATORS CD **£5.00** - All of the available QL emulators!
QL PD-CDR **£5.00** - Gérard Plavec's software collection!
LINE DESIGN CLIPART CD **£10.00** - Huge collection of Line Design clipart!
QL RELIGION CD **£10.00** – A collection of bibles, religious texts, clipart etc

The CD-ROMs are supplied in QXL.WIN format on an ISO-9660 CD-R for use with QXL, QPC, QemuLator v2, uQLx, Q40, Q60 (systems able to read QXL.WIN media)

POSTAGE UK/Europe: add £1.00, Rest Of World: add £2.00.
PAYMENT In UK Pounds Sterling currency. Cheques payable to DILWYN JONES.

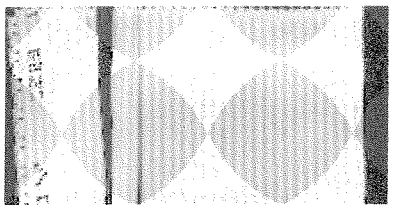
Dilwyn Jones, 41 Bro Emrys, Tal-y-bont, Bangor, Gwynedd, LL57 3YT, U.K.
Email:dilwyn.jones@dj.softnet.co.uk
Website: http://www.soft.net.uk/dj/index.html

Note: Although this service is run by the editor of QL Today magazine, it is a completely separate venture which has no connection with the publisher of this magazine. Software is supplied free of charge, apart from copying and media charges etc to cover costs, and without any warranty other than to the replacement of defective media (e.g. "bad or changed medium" errors etc.).In other words, usual PD library terms. E & OE.

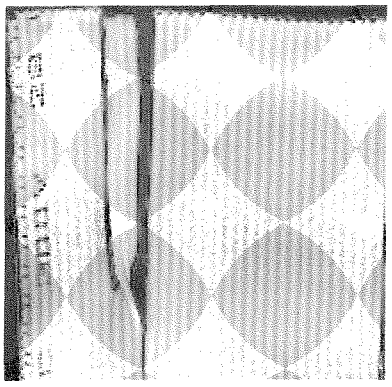
QL was in its prime. They almost all have at least three 3.5" drive connectors on them even though no PC has more than one floppy and very few PC files have not been bloated to a size that will not fit on a floppy anyway. That, however, is another argument.

Meanwhile back on the workbench...

Having successfully decapitated the original cable lay it flat on a cutting surface and locate the first wire. This is usually marked in red or black. Count outwards from that and make an incision separating lines 1 to 9 from the other 25. then count three wires and make another incision. You should cut about one and a half inches back along the plastic which divides the lines being very careful not to expose the wire. See picture 1.



Once you have done this take the three centre wires and twist then through 180 degrees. See picture 2.



Line up the twisted wire with those which remain. Place the whole ribbon cable end into a new 36 way IDC header so that the wire marked as '1' coincides with the little triangle marked on the connector and clamp the unit together. This should then be plugged into the drives and the last one on the line will be flp1_. The first connector on which the wires pass straight through will be flp2_.

And now, as they say in the dreadful shampoo advert, 'here comes the science' - just in case you want to know.

Shugart 34 way.

Common Earth				
	1	2	DISK CHANGE
	3	4	IN USE
	5	6	SEL 4
	7	8	INDEX L
	9	10	SEL 1
	11	12	SEL 2
	13	14	SEL 3
	15	16	MOTOR ON L
	17	18	DIR SEL L
	19	20	STEP L
	21	22	WR DATA L
	23	24	WR GATE L
	25	26	TRACK 00 L
	27	28	WR PROTECT L
	29	30	RD DATA L
	31	32	SIDE SEL
	33	34	READY

The diagram above came from Keith Mitchell's hardware file and shows an IDC connection. As you can see all of the odd numbered cables are ground and the even ones are the only ones to do anything. By cutting the cable at lines 9 and 12 and reversing it you are, effectively, changing the drive selection. A crude way to achieve it but there you are. On those drives which have no bump polarisation (ie. the little bit of plastic that protrudes from the plug) you can easily reverse the cable, thus applying signal where there should be earth. This usually results in the drive spinning continuously with the light on and the complete destruction of any unprotected disks in the drive.

Going Native Can Have Advantages

Uses of the Internet will, no doubt, have encountered the nasty SirCam virus over recent months. it has become the most prevalent virus on the internet and, in it's new guise, it does the neat trick of sending itself without displaying the address of the host. This means that the virus is impossible to block and recipients cannot warn the sender that he has a virus. Most of these viruses do little damage to the host computer and just get into the address book of the machine and use that to send themselves to other users. Like many Demon customers I use the Turnpike program to send and receive email and that has been immune to virus attack because it does not follow the Micro$oft model. The latest version of Turnpike, however, is closer to the M$ original so I wonder quite how vulnerable it is.

This virus does one other thing when it sends itself to addresses in the list. It selects a file at random from your documents directory and sends that. When the recipient opens the file the virus attacks. The whole point of this discourse is that we received one of these at work a week or so ago. The person I work with was very interested in the file he received because he knew the sender as a business rival and it seemed to be an interesting document. Being sensible he did not open it but was consumed by curiosity. The QL's immunity to virus

activity came in very handy and I took the file home and converted it to a plain text file for him.

A Qubide system can also prove very useful to non QL users. In the past I have been able to connect drives which have had their maps so badly trashed that the PCs FDisk cannot get into them. Once formatted on the Qubide they can then be resurrected on the PC with no trouble. I used this method when I had the shop and rescued many drives this way but I am not sure what the upper limit, in Mb, that the Qubide can see. The Qubide seems fairly unique in doing this since the Q 40 partition program provided by Tony Tebby does not seem able to do it.

And Finally a Little on the CD

Another thing I have been looking into this month is the CD driver and QXLtools suite. My total failure to get this working has probably been due to the CD drive which is connected to the Q 40. The drive does not seem to yield its ID to the system although I can open and shut the doors.

There have been reports of people getting this to work on the user group and Wolfgang Lenerz (two mentions in this column, begad!) has reported a staggering length of time to get his QXL.WIN file copied over to his new Q60. The slowness of the operation has many causes, Slave Blocks in SMSQ/E and a lack of optimisation in the driver software to name but two but the driver is an Alpha version and, as Wolfgang himself remarked, you have to be impressed that both the software and the hardware worked continuously for more than a day without crashing or corrupting data. As he also said Micro$oft cannot even get some of their release versions to do that most of the time. I hope to get a new CD ROM Drive next month and try it out then.

Snail Mail or Email?

Tony Firshman has completed the task of re-assembling the mail database for the QL Users. This has been done to allow those of you who have email access to get a direct mail with news of coming shows. The emails will only be used for this purpose so you will not be bombarded with advertising from us (as if there was anything new to sell). If you have not received an email from Tony about the coming raft of events then please let him have your email address as soon as possible:

tony@firshman.demon.co.uk

We have already had a complaint, in particularly haughty terms, calling this spam. Considering we are mostly sending very occasional show notifications with a bit at the bottom saying who will be attending that is a bit rich! If you don't want to receive these mails then a polite note to Tony will remove you from it. Each email we send will mean that we do not have to send out paper copy saving both trees and our hard earned cash so please let us have your details. If you have received neither a email nor a paper notice from us about the show contact Tony to get your details onto the mailing list.

Help request

Jeff Wass asks:

When ProWesS is loaded with QPC2 in Windows98 the buttons in the button frame will appear in a random order each time. Do you know of a reason for this?

We don't know, but our readers may. Any idea? Please write to QL Today!

Any other help required? Write to us too!

Humour

John Taylor writes:

This chap knows his PC.

Try all the options, but NOT crash, It's hilarious. Try this internet site:

http://128.241.244.96/portal/uploads/27000/27549_winrg.swf

The editors amused themself "checking out" all the features of Windows RG - it seems to be Really Good!

After reading through QL Today - did you notice something? We have printed the masters in 1440 dpi on an EPSON Stylus Color 900 instead of a 600dpi laser printer. Quite a difference, isn't it!? Let's hope the printer's copiers do net get stuck on the special 1440 dpi paper. If it works then we will keep this quality!

The QL Show Agenda

QL Meeting - (NL) Eindhoven

Saturday, 10th of November, 10:00 to 16:00
Pleincollege St. Joris, Roostenlaan 296

This time, the dealers plan to attend again.

QL Show - (F) Paris

Saturday, 13th of October, 10:00 to 16:00
Université Paris, 82 Rue de la Liberté, Saint Denis
Room number B231
Métro - Ligne 13, terminus Saint-Denis - Université.
Bus - lignes 253, 254, 255, 260, 268, 356, 361
http://www.univ-paris8.fr/up8/moyens_d_acces.shtml

Security first!

We would like to stress that credit card orders without card verification code cannot be processed anymore as of October, 1st, 2001.

All orders which we received so far did not specify any card verification code, but at the moment, it is an option. This will stop from October on! Also, this code will not be stored in any database: therefore you have to provide the code with EVERY order, even if it is a successive order. Please add this code to every order, otherwise there will be unnecessary delays!

Remember: the card verification code is on the reverse side of your card, in the signature field: the three digit to the very right (next to all or part of your credit card number)!