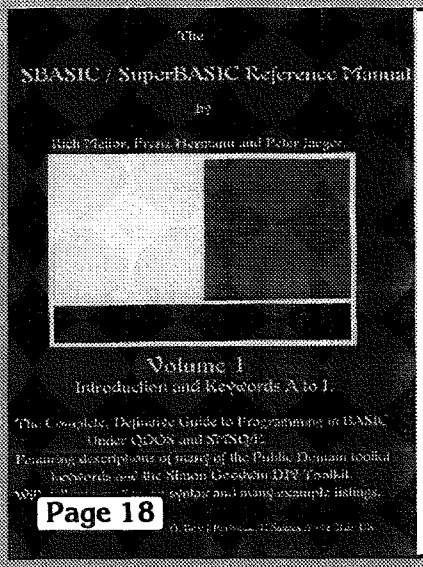
QL Today

Volume 3
Issue 6
March/April
1999

ISSN 1432-5454

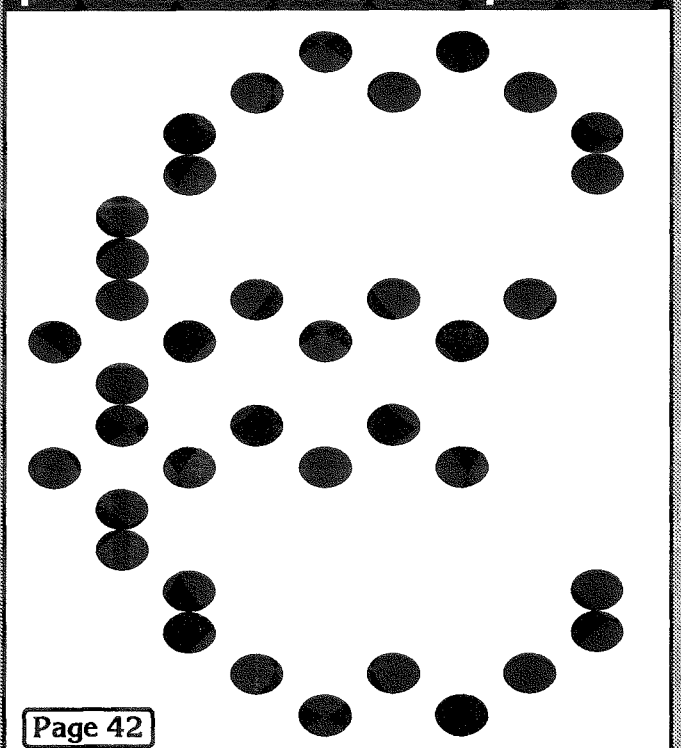
The Magazine about QL, QDOS, Sinclair Computers, SMSQ...



Page 18

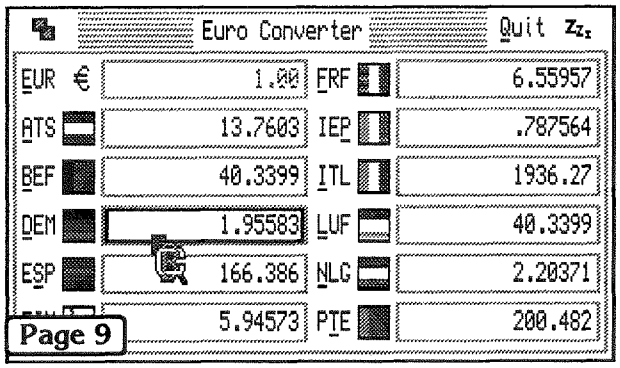
Al Boehm takes a close look at Rich Mellor's SBASIC/ Super-BASIC Reference Manual and the discs enclosed

Dietrich Buder shows you how to print a EURO on EPSON printers!



Page 42

Confused about the Euro conversion? Your QL does it for you!

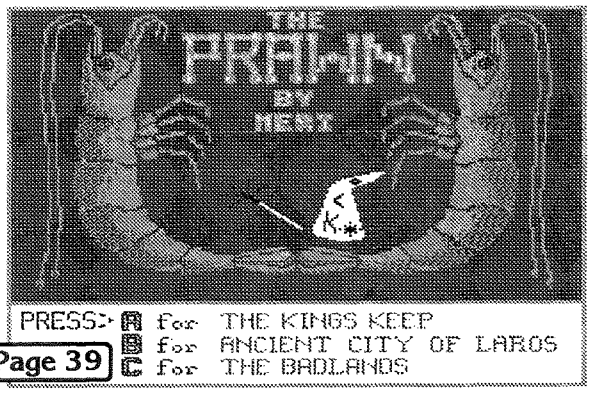


Page 9

Once again! Another very interesting cover disk for you!

Page 8

Darren Branagh starts a series of QL Adventure reviews



Page 39

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
email: JMerz@j-m-s.com

English office:

Q Branch
The Bank Volt
6 Coronation Buildings
Ham Road
Sussex BN11 2NN
United Kingdom

Tel. +44 1903 200212
Mobile +44 836 745501
Fax +44 1903 208070
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

QL Today is published bi-monthly, our volume begins on 15th of May. Subscriptions begin with the current issue at the time of sign up. Please contact the German or English office for current subscription rates.
Payment in DM (drawn on a German bank) can be made by either Cheque or Euro-Cheque, Payment in £ (drawn on an English bank) can be made by Cheque. Cheques should be made payable to Jochen Merz Software (German office) or QBranch (English office).

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

We welcome your comments, suggestions and articles. YOU make **QL Today** possible. We are constantly changing and adjusting to meet your needs and requirements. Articles for publication should be on a 3.5" disk (DD or HD) 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 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: 30 February

Editorial

Dilwyn Jones

It's cover disk time again! I hope you enjoy the goodies we've compiled onto it for you. Our thanks to the various authors who helped out and contributed to this disk. Please read the page in this issue about the cover disk before attempting to use it!

One of the regular items on an end-of-volume cover disk is an index to the contents of the current volume. I compile the index myself and it always pleasantly surprises me just how much has gone into just six issues and just how much is still happening on the QL scene.

TF Services and QBranch have recently launched Peter Graf's Q40 board. This is a brave step. It's probably the most advanced dedicated QL hardware yet. It comes with the QDOS Classic variant of the Amiga QL emulator, and rumour has it at the time of writing this that Linux has also been ported onto the Q40 in Germany. Considering that the price of the Q40 makes it almost as expensive as, say, a low end PC, it will be interesting to see whether users would prefer high performance native QL hardware without being bound by another operating system, or a dual platform emulator compromise system. Of course, when full email and internet access and Web browsers are available for QDOS/SMSQ the whole balance of things may swing heavily back in favour of native QL hardware once people are able to do without other operating systems. As work is progressing on the internet access side of things in Switzerland, and a browser exists in the form of Jonathan Hudson's Lynx port which can already access TCP/IP on an uQLx emulator system, this might just happen this year!

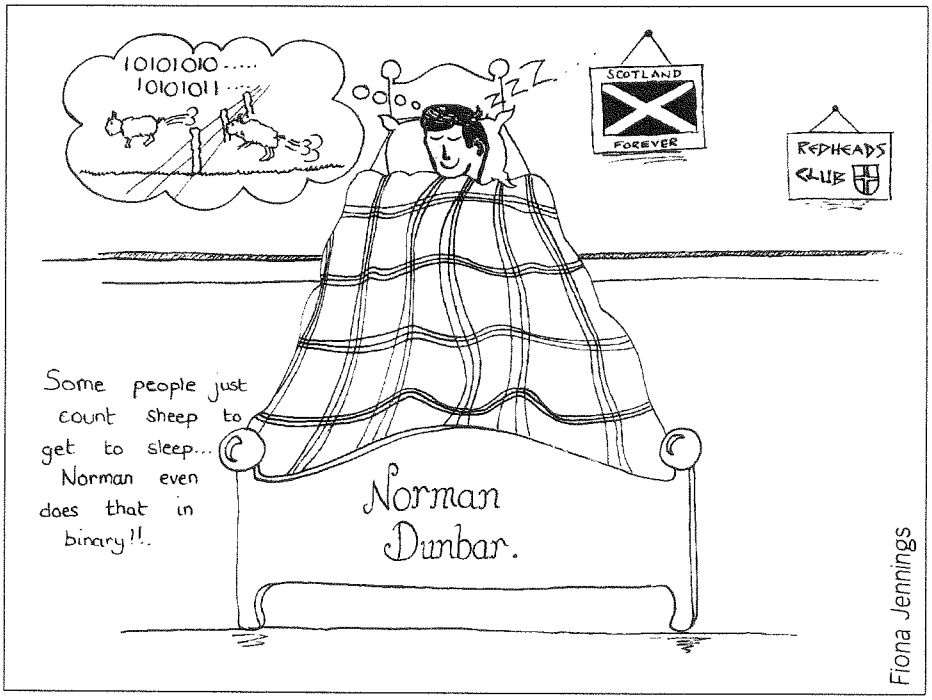
Any readers with, or planning to get, a Q40 willing to write a user report for us? I'm sure those readers who are undecided about getting one would love to read such a report! Please get in touch with me if you are willing to write about a Q40.

I did a simple Web search recently for the phrase "Sinclair QL". I was surprised to find over 800 results! We've come a long way - the QL presence is firmly established on the Internet. We really need that Internet access software soon.

We had hoped to have an article about Things in this issue. Unfortunately a combination of lack of room and the quantity of rather technical articles such as the assembler series caused us to believe it would be better to hold it back until a future issue.

From the next issue on, the magazine release date will shift by two weeks. This allows us to deliver the issues personally to the US readers at the show end of May, and will hopefully be useful in the future - most shows seem to fall at the end of a month/beginning of the next month. As QL Today is not quite a profitable venture, we have to save money wherever we can to be able to continue with this service for you. Handing over the magazines at shows in the USA in particular helps, as well as giving the bulk of magazines to Roy Wood instead of sending them by UPS. The deadlines will shift by two weeks as well, of course.

Cartoon



Fiona Jennings

News

News from London.

Turbo Toolkit:
The new Turbo Toolkit disk has been updated again after valuable input from Mark Swift. Turbo Toolkit currently stands at version 3c27, the previous distribution was 3b27. This toolkit is freeware and should be used in place of all previous versions of Turbo Toolkit and the Runtime Turbo Toolkit.
What you get with this update is the following:
TurboPatch, an EXECutable file that patches Turbo compiled programs to make them both 32-bit clean and able to deal with system variables not at the usual QL address. TurboPatch is at version 1.10, and will repatch programs patched with previous versions, removing the old patch and replacing it with the new. It will refuse to waste its time and yours patching programs already patched, or not compiled with the last official version of Turbo.
CONFIGURE, Mark Knight standard configuration program to configure defaults for TurboPatch.
New TURBO_TK_CODE file, works better than previous versions on all known platforms including SMSQ/E, QXL, Q40 etc. Includes support for high resolution screens and some other new keywords.
Altered DEMOS_BAS listing, accesses system variables using SYS_VARS instead of hard-coded addresses.
ReSize_BAS listing, shows how to provide your own programs with resizeable repositionable windows that will work on high-resolution screens; uses the FWINDOW% function from the new Turbo Toolkit.
Some _BIN files to allow daughter SBASICS and Minerva MultiBasics to use some of the basic structure keywords if desired.
Present_EXN file, extension file needed to allow TurboPatch program to run.
NewTurboToolkit_TXT, a text file explaining the changes made to Turbo Toolkit and the use of the _BIN files.
TurboPatch_TXT file, explains how to use TurboPatch program.
NewTurboTK_ZIP is a file that contains all the above compressed using the QL version of InfoZip. It should soon be available from QL public domain sources including bulletin boards. A disk containing the files uncompressed should soon be doing the rounds for those who haven't learned/won't learn to use UnZip.
Work on Turbo Toolkit seems to be nearly finished and the update is freeware.

Perfection:
Another version of Perfection is about to ship to members of the beta-test team, with a new CONFIGURATOR so the extended defaults can be used. This version of Perfection uses high resolution screens properly, so you may resize its windows to use an Aurora screen, and preconfigure large windows if you wish. It also works even if you configure it for, say, a 1024x768 screen and then start it up in 640x480, since if the configured windows don't fit it returns to internal defaults. It has better configuration options, including the facility to use a subdirectory to load fonts and the full length of a QDOS or SMSQ/E filename for the default. As this project advances the day when it is released must be drawing near, and I hope this will happen before or during March - no promises, but it looks likely. The new Perfection is likely to be freeware.

K-BASE:
This freeware package (NOT public domain), which received a glowing review recently in QL Today, is currently being updated slightly to give it better screen handling on extended QL and compatible systems. A minor change to the search routine is also planned. The updated disk will then be sent to QUBBESOFT PD and any other PD libraries I can think of and may appear on a QL Today cover disk.

Chinese proverb: "The man who does not make mistakes usually does not make anything". I sent the wrong version of my article on the QL clock problems and it was duly published in the Jan/Feb 1999 QL Today. Throughout the article where dates refer to 1960 Jan 01 00:00:00 as the start of QL clock time they should of course refer to 1961 Jan 01 00:00:00. Sorry if this caused any confusion to anybody typing in listings or examples. The only version on my hard disk refers to 1961 and I don't know how I managed to ship the earlier version to QL Today.

Portable Screen Toolkit:
Release is near, the machine code parts have now been thoroughly tested on everything from a standard QL with Expanderam to a Super Gold Card system with Aurora and SMSQ/E, as well as other SMSQ/E platforms like the Atari ST and TT and a QXL. I am still working on the SuperBASIC example and tutorial files and the documentation.
For those who didn't hear, this toolkit allows programmers to produce good screen handling and/or animations that work on a wide range of QL and compatible platforms. For one test I produced an animated 3-D fractal on my Gold Card QL and ran it on a Super Gold Card/Aurora system, a QXL, an Atari QL emulator and under various ver-

sions of SMSQ/E. The SuperBASIC examples will include 3-D drawing routines to be used in your own programs.
Other uses of the toolkit include simple things like giving a program the ability to resize and reposition its windows. Used properly it will make good screen handling easier and help improve the portability of software that makes effective use of its facilities. The Portable Screen Toolkit will be freeware.

The Fractal Collection:
This is almost finished but the documentation is in a poor state and needs a lot of work. The Fractal Collection programs use high-resolution screens properly to produce spectacular full-screen fractals up to the maximum size your system can display. Major components of the Fractal Collection are Easy Attractors and a pair of Mandelbrot programs. Easy Attractors plots the Attractors of Iterated Function Sets of Linear Affine Transformations, and no, you don't have to understand what that means to use it, just look at the spectacular patterns. The Mandelbrot programs are Mandelbrot Mania and Zoom Mania, produced jointly by Mark Knight and George Gwilt.
There are two Mandelbrot programs because Zoom Mania requires a hard disk and produces animated Mandelbrots (using the portable screen toolkit). There are other more minor fractal programs in the collection too, and it is intended that the manual should combine with the software to provide an educational package to teach users about the history of fractals and chaos theory.
The Fractal Collection will be sold by Q-Branch.

Mark Knight, 304, Portobello Road, Notting Hill, LONDON, W10 5TA. (0181) 932 6987.

C68 Support Pages

Dave Walker's C68 support pages on the Web have now moved to a new site.
His new home page address is
http://www.itimpi.freeserve.co.uk
In addition to C68 support, you can also download George Gwilt's software from this address.

Superbasic Sourcebook

by Tim Swenson
After too much thought and a whole lot less effort, The SuperBasic Source Book is now available. It was to originally be named the Qliberator Source Book, but a fair bit of information applied to more than just Qliberator.
It's just over 30 pages in Quill (about 90K) and covers Qlib, Toolkits, and programming tools. It is available from my web page:
www.geocities.com/SiliconValley/Pines/5865
in the QL PD Documentation Project page.
I plan to add to it as I come up with more material. Looking for lots of feedback also. So get it and give it a read.
swensont@jack.sns.com

Q40/Linux 68k NEWS

Richard Zidlicky has ported Linux-68k to the newly released Q40. This raises an interesting possibility, since Richard is author of the uQLx emulator for Unix/Linux based machines, that the Q40 might itself be able to run a QL emulator under Linux! Linux is used by many QL users, and the original designer of Linux (Linus Torvald) used to be a QL user at one time.
Richard is now looking to have Q40 support added to future releases of the Linux system. This may have the potential of increasing the target market for the Q40 among Linux users too.

MicroEmacs Editor News

Thierry Godefroy has made the QL version 4 pointer driven MicroEmacs editor available from his Web site.
An updated MicroEMACS v4.00 release (18/01/99 release) is available at the following URL:
http://www.imaginet.fr/~godefroy/english/download.html
You may also look at:
http://www.imaginet.fr/~godefroy/english/QLnews.html
for a log of the changes since the last release (the main change being the QTyp II spell checking capability).

'The Library' News

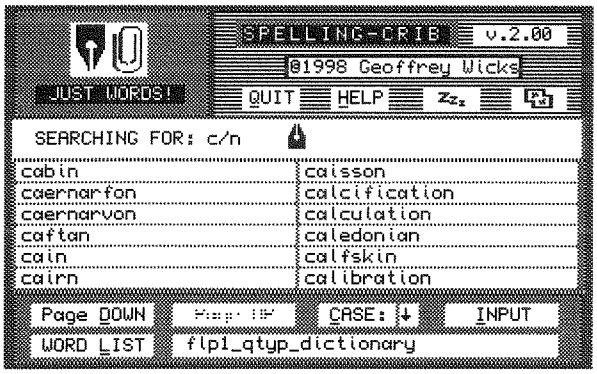
Phil Jordan emailed us to let us know that The Library (as he now calls his PD library service) should be up and running very soon. Things got slightly delayed by equipment failure (no doubt the shock of the workload of transferring the sheer volume of the former Steve Johnson PD library material), but at the time of writing, in mid February, work was progressing well. As soon as The Library is ready he will dispatch the requests received so far for the catalogue. He was hoping to take The Library to attend the Hove Quanta Workshop if ready in time.
Some new PD releases were also being added at the time of writing, for example, Geoff Wicks' his Spelling Crib program complete with Qtyp Dictionary - a new version of the program reviewed recently in QL Today.

Philip Jordan, 42 Hawthorne Crescent, Cosham, Nr. Portsmouth, Hants., PO6 2TP, England. Tel: 01705-370574
philip@jordanplj.freeserve.co.uk

JUST WORDS! NEWS

Spell-Crib now uses QTYP

Version 2.00 of JUST WORDS! freeware program SPELLING-CRIB is now available. This version uses a QTYP dictionary instead of a plain text file and needs just 39k memory. SPELLING-CRIB is a pointer driven program and will work with any QTYP dictionary in any language. On the SPELLING-CRIB disk is an English language QTYP dictionary of about 65,000 words. The standard QTYP dictionary has about 46,000 words.



MISTAKES IN QTYP DICTIONARY

When I was writing SPELLING-CRIB, I came across a mistake in the standard QTYP dictionary. "Voracious" in all its forms is given as "Voratious". I did a check in the 18 volume (!) Oxford English Dictionary and no word "Voratious" exists. Could people let me know if they find faults in any of the English or Dutch QTYP dictionaries, including my 65,000 word version? I understand there are also some mistakes in the QTYP French dictionary. It is important that we ensure these dictionaries are as accurate as possible.

Geoff Wicks, 28 Ravensdale, Basildon, Essex. SS16 5HU. Tel: 01268-281826

FWD Computing News

Frank Davis writes of his decision to cease selling QL products in North America:

Just a short note to let all of you know that North America is getting a new QL and Z88 dealer and place to get upgrades and repairs done for their beloved Sinclairs. This guy has been working with me for the last few years to do upgrades and repairs on Z88s. He is also well versed in Sinclair (yes speccys too!) for repair also. He is also knowledgeable about monitor repair. I highly recommend that you support him.

Dealers from elsewhere. I hope that you will take the time to contact him and get yourselves established with each other so that you can all continue to support each other. His name is John Rish and he lives in San Antonio, Texas here in the USA. He is a subcriber to QL Today and to QUANTA. He needs for the Sinclair user groups to contact him, as well as TSNUG. Give him your support, and I thank all of you for the support that you gave me over the years. I will still be a user and available for advice if I can be of assistance, but no longer as a dealer. Time, space and other considerations have taken me out of that. I am now only dealing in Amiga and some MAC software, no hardware. John is buying a lot of my QL and Z88 stock.

Rita Jean Willis bought part of my stock but will only be dealing with South Americans as she is out of Rio de Janiero. Should she contact you, please be of assistance to her.

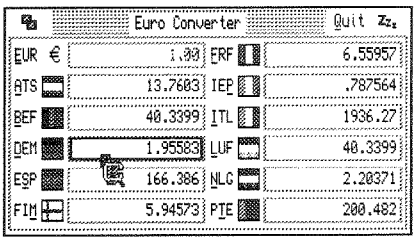
John Rish's email is:
74601.1535@compuserve.com
He has gotten started on his first web page for his business.
http://members.tripod.com/hes_computing/hes1.html

New Software from Beginners' Club

Beginners' Club (of Italy) has announced two new freeware software releases:

* The Reader v1.40, a new version of the QWK Off-line Reader under pointer environment.
* Euro Converter v1.10, a program for converting between Euro currency and the national currencies of the member states of the European Monetary Union.

Both programs may be downloaded from their Web site at:
http://www.geocities.com/SiliconValley/Lab/5011/



New USA Dealer

Home Electronics Service is pleased to announce to the Sinclair community that we have taken over the business from Frank Davis, (FWD Computing).

We have been a part of the Sinclair scene since 1982, repairing ZX Spectrums for Sinclair Research in England.

We provide Sales, Service, and Software for the QL, Spectrum, ZX81 and Z88.

Look for us on the web at
www.members.tripod.com/hes_computing/hes1.html
Our email address is
74601.1535@compuserve.com

Hours of Operation is Monday - Friday 1300 hrs to 2100 hrs central time zone.
Phone number: **210.661.4376**
Our mail address is:
5222 Kazen Dr.
San Antonio
Tx 78219
USA
We look forward to serving you.
John R. Rish Home Electronics Service

QLATter UPDATE

Al Feng emailed us to let us know that QLATter V1.209 is now available free of charge from him, via email, from **alfeng@juno.com**
QLATter is a freeware utility intended for use with Jan Venema's QLAY emulator, but can be used with a regular QL or QDOS compatible.
Al Feng, 914 Rio Vista Circle SW, Albuquerque, NM 87105. USA

RWAP Software News

SBASIC/SuperBASIC Reference Manual

Release 1 of the SBASIC/SuperBASIC Ref. Manual is available as from 28/2/99. This new version is available at the same price of £40 plus post and packing.
For people who purchased the manual prior to this date: updates for release 1 cost £6, users can order this and the next update for a total of £10. It is also worth updating the support disks - send the original disks and £2 for this.

Forthcoming Products

By the time that you read this, Q-Help should be available at the cost of £10. This is a small utility to allow you to find out the syntax of the most common SuperBASIC keywords and a small description of their use.
Q-Help builds on the data provided by the SBASIC / SuperBASIC Ref. Manual and is not intended to replace any part of it. The reference manual provides a lot more detailed information about each keyword and compatibility issues. Q-Help can be called from the interpreter, or run as an executable job. It is easy to add your own help files to the system and the program can be linked with Q-Index supplied with the reference manual.

Flightdeck

Flightdeck will also soon be available at the price of £10. This is an excellent flight simulator (originally from DeltaSoft in Bristol) which already has a large following. If you want to learn how to fly a twin jet passenger plane on instruments, this is the program for you. Fly between any of 24 UK airports and even add your own for more fun. I am currently working on a program to ensure compatibility with SMSQ/E, although the current version (available now) can be used without any problems on standard QLs and Minerva.

New Web Site

My new Web site has moved. It is now at:
http://rwap.webjump.com

Volume 3 Issue 6 Cover Disk

Some of the files on this cover disk have been compressed using a program called ZIP, so that they require less space on the disk and allowing us to pack more onto it.

OBLIGATORY NOTICE

This cover disk contains files compressed using Info-ZIP's compression utility. These files need to be decompressed using the Unzip program supplied on the disk. Info-ZIP's software (Zip, Unzip and related utilities) is free and can be obtained as source code or executables from Internet/WWW sites, including

http://www.cdrom.com/pub/infozip/

You can get versions for QDOS/SMS systems from Jonathan Hudson's Dead Letter Drop web site:

http://www.jrhudson.demon.co.uk

USING UNZIP

You can get full instructions and other files for Unzip and Zip from the above Web sites. Briefly, Unzip decompresses files to the DATA_USE default drive. For example, to unzip the file containing listings published in this volume of QL Today, place the cover disk (or preferably a backup copy) in FLP1_ and a blank, formatted disk in FLP2_ (you can use RAM1_ instead of FLP2_ if your system only has a single floppy disk drive), then enter the following commands in SuperBASIC or SBASIC:

```
DATA_USE FLP2_
EX FLP1_UNZIP;'FLP1_LISTINGS_ZIP'
```

The display now shows a list of files being unpacked by the unzip program. Wait until it has finished - this may take a couple of minutes. You should now find the decompressed files on the disk in drive 2.

There are several _TXT files on the disk. These are plain text files - you can view them with any text editor, or with the VIEW facility in QPac 2, or even with the QED editor on this disk once you have decompressed that program.

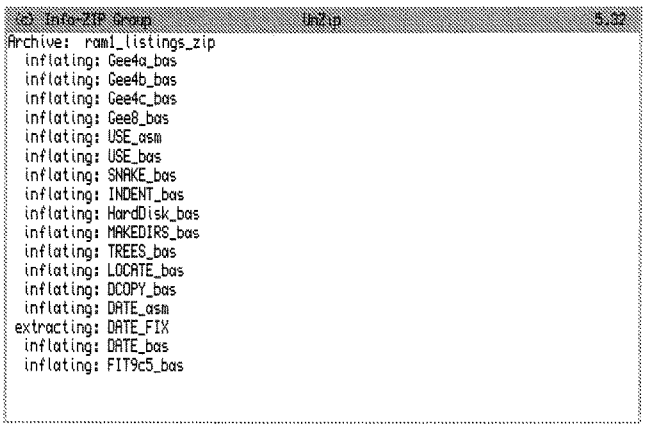
LIST OF FILES SUPPLIED

UNZIP - program to decompress the "_zip" files on the disk - see above.

README_TXT - A plain text file describing the contents of the disk. Please read this file. Among other things, it tells you the names of the instructions files where relevant for the programs on the disk.

GWASL1v2_zip - Version 1.20 of George Gwilt's 68008 assembler program, to accompany Norman Dunbar's assembler programming series. The GWASL software is Freeware.

CONFIG - EXECutable program for configuring the GWASL assembler to accompany the Norman Dunbar Assembler series.



QED_zip - Version 1.01 of Jan Bredenbeek's QED editor (Freeware), to accompany Norman Dunbar's assembler programming series.

68kTEXT1_zip - Plain text file with 68000 instruction set notes for assembler programmers, to accompany Norman Dunbar's assembler programming series.

Trans24_zip - Version 1.03 of Trans24, a 9 pin graphics to 24 pin graphics printer dump converter filter. This is a Cardware program, which means it is free to use and copy, but the author (Ralf Rekönd) would like to receive a postcard from users of this program. If you like this program, send a postcard to the author in Germany: Ralf Rekndt, Steinstrasse 33, D-53773 Hennef/Sieg

KBASE220_zip - Version 2.20 (Jan 99) of K-Base, a simple to use database program by Mark Knight. An earlier version of this program was reviewed by Darren Branagh in Volume 3 Issue 5 of QL Today. This program is freeware.

TUTK3c27_zip - Version 3c27 of the recent Freeware release of Turbo Toolkit. This version is produced by the team of Mark Knight and David Gilham. Also included is a Turbo compiled job patch program based on the work of Davide Santachiara and Mark Swift, which patches the startup code of older versions of Turbo compiled programs.

SPELCRIB_ZIP - Version 2.00 of Spelling Crib by Geoff Wicks of Just Words! (author of Solvit Plus, QL Thesaurus and QL Style Check). This is a more recent version of a program reviewed in V3i4. This version uses dictionaries from the QTYP spelling checker - a sample English dictionary is included. Spelling Crib V2.00 needs Toolkit 2 and Pointer Environment. Spelling Crib v2.00 is Freeware.

LISTINGS_zip - assorted listings from the 6 issues of Volume 3 of QL Today, English language edition. Includes a text file called LISTINGS_TXT which contains details of the programs.

INDEX3_TXT - plain text file containing an index to the contents of the 6 issues of Volume 3 of QL Today.

If the cover disk is damaged, please send an International Reply Coupon to QBranch or JMS to abtain a replacement.

QD98 Review - Jochen responds

When laying out the review by Dave Westbury for the last issue, I was surprised how quickly he got used to many of the features (I know the date I sent it to him).

The review is very positive, but I guess every QD owner is aware that this program is programmed by somebody who uses his own creation. QD is and will always be under development, and suggestions for further improvements are always welcome (no promise that everything will be implemented!).

Now, a few explanations of some "unclear" points:

Although Dave (and most others) got used to a flashing "text changed" item, I've added a way of turning this off (in QD 9 already, in fact). Issue the command

```
SET_DEFAULT 96,"0"
```

and start a QD afterwards - the icon will appear if the text is changed but will not flash (not 100% true, this is the only known bug in Version A.00 of QD98, but it is fixed in A.01).

Marking a block - yes, the way it is done dates back from QD V1. However, I prefer the QD way of marking blocks, especially if you need to scroll. Compare with Windows: once you have started highlighting a block, you must not lift the finger from the mouse button. For long blocks, it is much slower and requires more finger acrobatics. Don't know if I would change it even if the Pointer Environment would support block marking by dragging. Marking smaller blocks in Windows may sometimes be faster, depending on the application (sometimes worse, especially in Word, where the program knows much better than you want you plan to do: parts of a word or expression can become a major fiddle!).

You can pick QD to the top and make the pointer/cursor move to the position at which you left it by "waking" it, i.e. right-click to pick the QD window (or HOT_WAKE, if you prefer hotkeys).

Delete selected ranges of control codes? Does anyone else think this is very useful? If so, tell me why, please!

Automatic line number remove/add on load and save ... hm, nice idea, but very, very difficult to be implemented in a consistent way. I will think about it.

The context menu will get block commands if a block is marked - you're not the first person asking for this, Dave, and it definitely makes a lot of sense. Version A.02 (after the QL Today issue you are reading) will have it implemented, so please be patient.

Finally: the explanatory window colours will be configurable in the next version too!

■

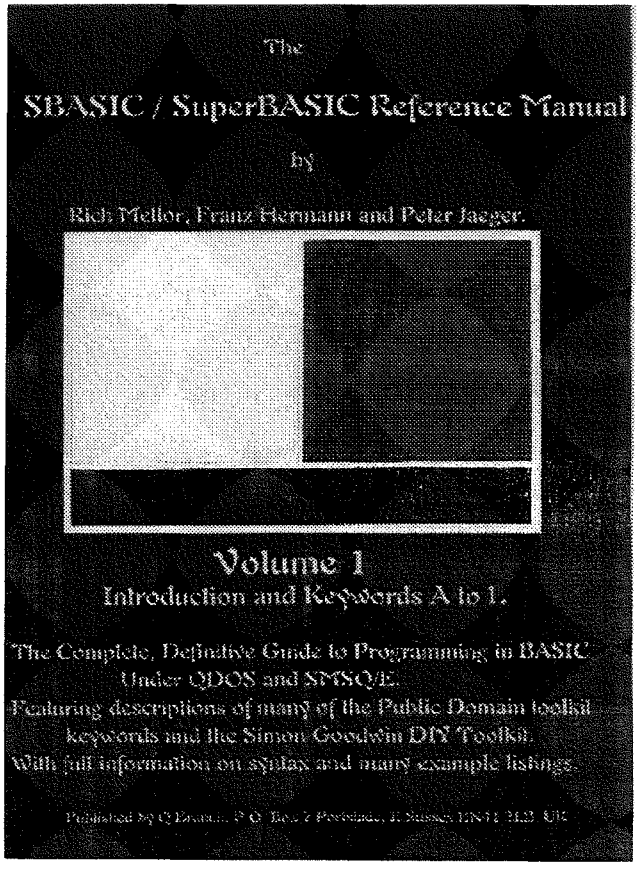
Secrets of the QL

Al Boehm

Some time ago, a NESQLUGer (New England Sinclair QL User Group) wrote in complaining that there seemed to be an "in group" that knew everything while most of us knew only a little about the QL. Recently a writer in QL Today came up with the same idea calling one group the "feeders" and the other group the "chickens". Well now you can be part of the "in group", one of the "feeders" simply by purchasing the new SBasic/SuperBasic Reference Manual by Rich Mellor, Franz Hermann, and Peter Jäger. Don't believe me? Well, read on.

The manual consists of two hefty inch and a half thick A4 volumes. Plus three disks!

The bulk of the manual consists of Keywords not only of SBasic and SuperBasic, and the SMSQ/E and Minerva operating systems, but of all the toolkits I have ever heard of plus many more I did not know existed (more on this below in the disk section), plus Keywords specific to various



boards. Yes, it even includes the Keywords for the QSound board which plays music and which Ed Kingsley and I had been looking for.
But it contains much more than Keywords. The manual has an enlightening introduction which discusses the various operating systems plus the proper way to install toolkits which I thought I knew all about - WRONG there are six distinct stages - speed enhancements, device drivers, toolkits, Pointer System, Secondary set up programs, and finally required jobs such as ALTKEY, FSERVE that need to be initialized before you load your main job(s).
The introduction also provides information on how to compile programs and how to write programs for the Pointer Environment.
There are also no less than 11 appendices: Minerva, SMSQ/E, Emulators, Thor, Expansion Boards, Compatibility, Multiple Basics, Error Messages, Character sets, Designing Fonts, Math, Device Drivers, Pointer Environment, Coercion, Mouse Drivers, QL Displays, and Networks. These are rather complete appendices averaging 14 pages each. For example, the Network appendix covers Qnet, Flexynet, Midinet, SERNet, Amadeus Interlink, and QL-PC Fileserver. I learned that Midinet is not to hook up to a musical instrument but rather a way to use the MIDI ports available on some emulators for a net between computers.
The Keyword section contains not only the syntax and a very understandable description of usage but also contains which toolkit/board it is from, examples, notes, implementation, warnings, and cross-references to other keywords. Here is an Example: (Italics which help in Syntax have been omitted in this review to enable me to send it email.) (Used with permission.)

VER$

Syntax: *VER$*
or *VER$ [(n)]* with *n=1, 0, -1, -2* *(Minerva and SMS only)*
Location: *QL ROM*

The function *VER$*, which is the same as *VER$(0)* returns a short identification code for the version of the current system ROM. Here are most of the possible values (in order of development):

VER$	Explanation
original ROMs:	
FB	This is the first ROM sold in April 1984, QDOS version 1.00. It comes with a 'Dongle' - a board which needed to be plugged into the QL's ROM port. It is very unreliable and should be replaced!!
PM	These three ROMs were developed during the following two months.
EL	
TB	
AH	Released as the "definitive" version in June 1984.
JM	British QL, QDOS v1.03, the first version which was exported.
JS	Released in spring 1985, QDOS v1.10. Also found on early Thors and patched in ST/QL and early Amiga Emulators.
JSU	American QL.
MGD	Danish
MGE	Spanish
MGF	French
MGG	German
MGI	Italian
MGN	Norwegian
MGS	Swedish
MGB	Swedish
MGY	Finish
MG¤	Greek
¤FP	Greek
	These versions were only sold in the respective countries. ... QL QDOS v1.13
Patches:	
MGUK	A version of the MGx ROM produced independently for the UK market.
MG	Another patched version, mainly distributed in Germany.
MGUS	Out of the three patches, this is the only legal one and was produced for the United States

SBASIC/SuperBASIC Reference Manual 6/1/98 Section 5

As you can see a lot more information is provided beyond a simple explanation. The above VER$ entry is two pages long in the manual ***[we scaled it down, original size is A4, like this magazine - Editor]***. Most are typically one page long.
But some are much longer: OPEN is 5 pages with 7 NOTES, DIM is nine pages with 8 NOTES and 19 examples, one of which is two pages long! Of course, some are very short: INVXY has the syntax and brief description and then, WARNING: Do not use this!

Info on Disks

Three disks come with the manual. The first is an electronic index for the Manual. If you look up a Keyword in the manual you find its syntax and usage. But most of the time, you want to do something, but you don't know what the keyword is. That's what the electronic index does. You can start with a topic, say Accessing Memory and the relevant Keywords are displayed: PEEK, PEEKS, PEEK$, PEEK_F, POKE, POKES, POKE$, POKE_F, PROT_MEM, TTPEEK$, TTPOKE$, and TTV. In addition, Topic Cross-References are displayed: Reserving Memory, Memory Fragmentation, Copying/Moving Memory, and more. Each of which can be selected to obtain further Keywords. Also listed when appropriate, is Manual Appendices references. What a brain saver this is!

The second disk contains all the example programs contained in the manual. Short one or two line examples such as found above in VER$ are very useful. But that is not what is meant by example programs. We are talking about stand alone or full procedures that illustrate a Keyword capability. Many can be used in your own programs to accomplish a difficult chore. There are over 200 example programs on the disk.

The third disk contains the Public Domain toolkits covered in the manual with the source code included for many. This third disk is important in two ways.

1. I don't know where I would have gotten my hands on some of these toolkits.
2. The docs on explaining the toolkits make me appreciate the manual more. You see, most of them are in German, French, or Spanish! Ordinarily one uses a reference manual only when needed to look up a particular item. But I find myself leafing through this manual just to find all those tidbits of QL lore that somehow escaped me, and now I can feel like a "feeder" not a "chicken".

The SBasic/SuperBasic Reference Manual is available directly from Rich Mellor at RWAP or you can use credit cards at QBranch for 40 pounds ($67) plus postage. ■

New developments:
CS
PT — Different ROMs used on the THOR XVI
PO

JSL1 — QL with Minerva ROM, a very much debugged and enhanced version of the JS ROM, available in all languages for all kinds of QLs.

HBA — Either the SMSQ or SMSQ/E replacement operating system for QXLs, Atari ST/STE and TT series computers and the Miracle Gold Card family of add-on cards.

NOTE 1:

VER$* can be used to write flexible programs which adapt themselves to specific features of computers and ROM implementations. However, if you intend to test *VER$, for example:

```
IF VER$= 'JM'
```

to retain compatibility with the different ROM versions, you must first assign the contents of *VER$* to a variable:

```
100 a$=VER$
110 IF a$(1 TO 2)='MG':PRINT 'MG ROM'
```

NOTE 2:

The names of the original ROMs were derived from names of Clive Sinclair's secretaries, taxi drivers he met and so on. (Just in case you are looking for any sense behind the abbreviations.)

MINERVA NOTES:

On Minerva, *VER$* accepts a parameter:

VER$(0) — as per above VER$.
VER$(-2) — returns the base address of the system variables (normally $28000 = 163840 on a standard QL).
VER$(-1) — returns the current job identification number.
VER$(1) — returns the version of QDOS (see also QDOS$).

SMS NOTES:

VER$ has been amended to provide the same facilities as on Minerva.

WARNING:

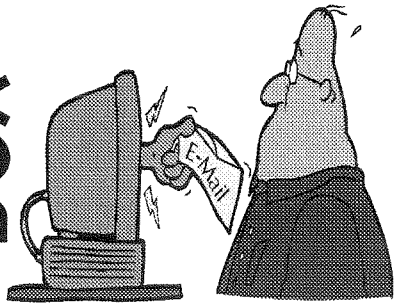
If you fail to assign *VER$* to a variable before testing its value, then you can crash a JS (or JSU) ROM. This will also happen on Minerva ROMs (pre v1.77) with Minerva's extended variant *VER$(n)*.

CROSS-REFERENCE:

QDOS$* returns the version number of QDOS in the same way as *VER$(1). See also *MACHINE* and *PROCESSOR*.

SBASIC/SuperBASIC Reference Manual 6/1/98 Section 5

Letter-Box



George Gwilt writes:

While I agree totally with Norman Dunbar that his Example 2 in QL Today Jan/Feb page 25 is better than Example 1 for the loop he illustrates, I myself do sometimes use the latter form. In a piece of code where the number of iterations is presented in D0.W from somewhere else, rather than being explicitly known, Norman Dunbar's preferred code might become:

```
Start SUBQ.W #1,D0      adjust to 'count-1'
Loop  BSR    Useful_code
      DBRA   D0,Loop
```

Unfortunately you will get a nasty surprise if D0.W happens to be zero. To correct this you might try:

```
Start SUBQ.W #1,D0      adjust count and set
                        condition codes
      BMI.S  End        exclude a zero count
Loop  BSR    Useful_code
      DBRA   D0,Loop    End
```

Even this might not be what you want. Although BMI.S correctly stops a zero count being used, it also stops counts of 32769 to 65536. To allow this range the program might now become:

```
Start TST.W  D0         is D0.W zero? . .
      BEQ.S  End        . . yes - don't loop
                        but . .
      SUBQ.W #1,D0      . . allow counts of 1
                        to 65536
Loop  BSR    Useful_code
      DBRA   D0,Loop    End
```

However, the other method to do exactly the same is, simply:

```
Start BRA.S  Skip Loop
      BSR    Useful_code
Skip  DBRA   Loop       drop through 1st time
                        if zero count
```

This is certainly neater.

I agree too that the DBT instruction is indeed peculiar, but so is NOP. Both instructions do nothing except take up time and space. Two NOPs are the same length as one DBT but take a shorter time. If you want a more efficient delay loop you might replace

```
      NOP
      NOP
```

by

```
      DBT   D0,*
```

(a non-existent branch to itself!) in its core.

P.H. Tanner writes:

On page 48 of the latest (Jan/Feb 1999) issue of QL Today, Mr. Westbury mentions "Master Spy's fatal bug..."

I well remember this as a feature, if that is the word for it, of version 1.8 of this marvellous utility as I used it for many happy years.

It has, however, been eliminated from version 3.3 as currently sold by QBRANCH. Since this also has many other additions, such as support for the larger screen sizes now available, and more flexible set up options, I would urge any users of earlier versions to upgrade.

(Thankee, Mr. Wood. Thankee, surr. That'll do nicely)

Mr. Westbury also refers to Master Spy's ability to edit binary files. It is one of the few editors that I have used which offers this option: and it is definitely the best among them for this purpose. But it entails the use of a full 256-character screen font, such as is the default in MINERVA.

Master Spy is unique, though, in that where other editors only offer Overwrite editing of binaries, it also allows Insert.

This makes many things possible. For instance, I have used it on occasion to add a QDOS serial header to the start of a binary file. Also to change the name of a job file to one of a different length (which may require adjustment of the initial branch displacement).

And, gloriously, it allows the concatenation of object files. This is useful in two ways.

Frequently an assembled program has in its tail a block of data which remains unchanged between versions. M-Spy allows this to be kept in a separate file so that only the active code requires to maintained and assembled: the data may be added to the object file by a simple edit. Of course this could be done "properly" using the LINKer, but M-Spy does it in half the time, and with none of the associated fiddle faddle.

And I suspect that I am not alone in having accumulated over the years a number of favourite enrichments of the basic system, each of which requires to be LRESPRed. It has been for many years my practice to concatenate these into a single file, LBYTESed into a single RESPR, and CALLed from the appropriate offsets. This would have been impossible without M-Spy.

What a crying shame it was that ARK never completed their word processor before they left the QL scene.

David Bunbury writes:

Mark Knight has done us a valuable service in reminding us of the imminence of the "2029 bug" and for suggesting ways of outwitting it; (QL Today Jan 1999, p34).

Unfortunately, it seems that some existing commercial software is already affected. After trying out Mr Knight's routines, I forgot to reset my clock and so inadvertently acquired a couple of post-2029 files on a floppy-disc. I was surprised later to see these turn up at the wrong end of a time-ordered directory listing from QPAC2. As the date on Qdos discs is held as a long integer, exactly as returned by MT.CLOCK, it seems inexcusable that the sorting algorithm should use signed arithmetic. My copy of QPAC2 is version 1.39, so it is possible that this may have been put right by now, but if not, I hope it will be given priority in the next revision. After all, there are only 30 years to go.

A few other points are perhaps worth mentioning briefly in this connection. In SMSQ/E, the DATE function has already been corrected so that it will wrap round in 2029 rather than resetting to zero as in JM and JS ROMs. That means that users of SMSQ/E will not need to replace this function.

The superbasic procedure for making an unsigned comparison of floating point numbers is, of course, still needed to deal with pairs of dates which straddle 19/1/29.

Incidentally, an alternative way of accomplishing this is to make use of the fact that string comparisons are inherently unsigned. Thus if A and B contain dates (as returned by DATE) then

A_before_B = HEX$(A,32) < HEX$(B,32)

will give 1 if true or 0 if false. This requires toolkit2. DATE$ cannot be used in this way as the name of the month is included in the returned string.

Contrary to what I had been led to believe, it seems that the dates on DOS-formatted discs are not held in decimal form after all and, in fact, they should be valid and unambiguous until 1980+128 years, so there is no immediate problem there.

Another interesting curiosity applies only to Gold Card users, as the Gold Card has its own clock. At midnight on 31/12/51 this will reset to 1/01/51. It also cannot be manually set to a later date and so, for those of us far-sighted enough to have kept our Gold Cards in service, it will thereafter be forever 2051, thus avoiding the 2097 bug altogether (not to mention the 2108 bug; see above). It is, perhaps, a bit too early to be speculating on what the wider implications of that might be but all the same, from now on I shall be keeping an anxious eye on the long-range weather forecasts.

Finally, can anyone explain why 1961 was chosen as the starting date of the QDOS clock when the QL did not come on the market until more than 20 years later? *[Yes, according to Tony Tebby that's because this is the date at which Aaron Turner was born - Jochen]*

[Anyone else know of any examples of date handling producing unexpected results in a QL program? - Editor]

Bristol QL User Group, by Henry Orlowski

It has been an honour and a pleasure to have been the Chairman of the Bristol QL User Group for many years. Until the end of 1998 that is.

Alas many of our members have fallen by the wayside and in the end it was with a great deal of regret that the committee put a proposal to the the AGM that the group should fold at the end of 1998.

It was a sad time when we had to sign the final bank forms to close the group account and forward remaining funds to QUANTA for their use as they see fit. I would like to thank all group members, past and present, for their efforts in making our nine years of activity so successful.

What happened? Well, whilst we were a very well organised group, meeting every month in the same venue, the level of attendance reached a low that it was unable to support the activities or the aspirations of the group. In earlier years we were a thriving group with well formatted and presented meetings. Towards the end, sadly, fewer people attended. For some the journey became too much and less regular. Others stopped being as centrally involved in the group as previously. Yet others had further commitments that were difficult to juggle with attendance at the regular meetings.

Many QL'ers will I am sure remember the excellent workshops that the group organised at either Clevedon's Walton Park Hotel or Portishead's Somerset Hall, and may think that these will become just a memory.

Fear not! Roy Brereton and the members of the group who organised the workshops discussed the possibility of holding future workshops in the area if QUANTA so requested. The result is that we can now look forward to the QUANTA Annual General Meeting and workshop at the Walton Park Hotel, Clevedon, on 25th April 1999. Please refer to the separate notices and announcements relating to this event in all the QL publications.

Mike Ashford may be contacted for local accomodation information and workshop table and talk allocation as usual. Mike is also acting as a contact person for previous members of the Bristol Group who wish to remain involved.
(Tel.No. 01934 415416;
e-mail **mjashford@uphill52.freeserve.co.uk**)
In this way we hope that somehow the spirit of the Bristol Group will stay to the fore within the QL community. Please do your best to help by attending this workshop and AGM to demonstrate that there continues to be life in this good old black box. It will be a pleasure to see you all there and for the QL community to continue to thrive as before.
Best wishes. Long live the QL. Henry Orlowski

Gee Graphics! (On the QL?) Part 9

Herb Schaaf

The 5 Platonic Solids

I've sent the editor a S*BASIC program "FIT9c5_bas" which was a 'work-in-progress' in 1996. It is too long to print with 97822 bytes or almost the same as 50+ "tedious pages of S*BASIC". I hope it will be put on the cover disk so that you can load and run it and browse through the listing if you like. Before you load and run it be sure to LRESPR 'SORT' from the Quanta Library. Graphic programs can call for lots of sorting, lots of memory, and never run fast enough. The program worked OK with SMSQ/E SBASIC and SGC, but not with SuperBasic and Trump Card. SuperBasic and Gold Card did work, but was very slow.
I was trying to put a model of Five Intersecting Tetrahedra or "FIT" into my QL in 1996. I'd made a modular origami of FIT from folded paper. After putting the tetrahedron into the QL I then decided to put all 5 Platonic solids into the QL before going on to the FIT.
I thought doing the Platonic solids would be easy after having done the Escher Knot. I was wrong. As I blundered along I tried to make the PROCedures more general so they could work for any of the five regular "Platonic" solids until I finally had all five of them in the QL. A scheme that worked for me in naming the many arrays was to use underscores and an extension of the ideas which you saw in GG#8. That allowed me to make calls to PROCedures with referenced arrays in a more general way.
The algorithm for "building" each Platonic solid within the QL is pretty much the same.
We make use of known values for face angles, inner, middle, and outer radii. We READ in these along with mathematical values and other DATA from Coxeter.
For convenience I set up some arrays to hold the values from Coxeter for Platonic Solids.
PS_name$(5,12) holds the names strings of the solids.
PS_vefpq%(5,5) holds the integer number of vertices, edges, faces, and the "Schlafli" p and q values; p being edges per face, and q being edges per vertex.
PS_angles(5,3) holds the dihedral, phi, chi, and psi angles, using the zeroth place for the dihedral angle.
PS_radii(5,3) holds the inner, middle, and outer spherical radii. The inner just reaches to the center of each face, the middle just reaches to the center of each edge, and the outer reaches to the center of each vertex.
Some mathematical "constants" from the Coxeter book are assigned to constants in the QL: the "golden ratio" tau, and its inverse, itau, along with the selected angles of kappa, lambda, and mu.
After getting the vertices into an array we make an array that associates each edge with its own pair of vertices. We do this by using the edge_length from Coxeter to collect appropriate pairs with the 'space_between' FUNCtion.
The next step is to do the reverse and make an array that links each vertex with all its edges.
What follows then is to get the faces collected into arrays that link faces with vertices and vice versa, also making an array to hold the equations of the planes of the faces, as well as making an array that links faces to edges and vice versa, and finally arranging the edges and vertices in a counter-clockwise order around each face. Have a look at the listing and REMarks to see how all this is done.
We use a variety of PROCedures and FUNCtions; line_n3space, Plane_frm3points, Pt_to_plane, rotate, angl_frm, cyc, swap, etc. to have the QL do the chores for us. I hope some of these may be useful to you in your programs.
There are other procedures not called by the program, but put in as a convenience to check 'progress' and/or debug. One is 'list_array' and another is 'review_links'. I made use of RESTORE as suggested by Mel LaVerne to keep track of PROCedures and FUNCtions

and also to keep track of my last line number. Take a look and see how it might be useful to outline, organize, remind, etc. for you in your own programs.
Because the modular origami was made of interlocking folded paper struts, I created similar struts in the QL, choosing a strut_width that allowed the intersecting parts to fit snugly. I have REMarked out the line that would 'build_fit', as it was not working 'just-right' at the time. You may want to try it and see just how far it goes before getting into trouble. Perhaps in a future article we will describe a later version that does work.
Colors in MODE 4 When it comes to displaying the many faces with different colors we almost run out of easy options in MODE 4. Avoiding horizontal and vertical color patterns (stipples 1 & 2) and not counting the background black as a color we have red, green, white, then the same 3 with a 1/4 black, then those 3 with 1/2 black, then those 3 with 3/4 black. In addition we have the 3 mixes of red and green, 1/4 red, half and half, and 3/4 red. Finally we have the 6 lighter shades of red-white and green-white combinations. It adds up 21 colors plus black, and fortunately the 5th solid has only 20 faces. It is just a matter of artistic license as to which color goes where. Try 'show_inks' to see my choices.
Next time I hope to have a short listing (less than 10K) for you which you can merge with the listing from GG#8 that will let you create and manipulate a stereo pair of tetrahedra in your QL.

catdoc - A Review

By Timothy Swenson

catdoc is a Unix program created by Victor Wagner and ported to the QL by Jonathan Hudson. catdoc takes a Microsoft Word file and converts it to plain ASCII text. And that's it. It is a simple program to run, simple to operate, and it does what it says it does.

So, why would you need catdoc? For Wintel (Win 3.1, Win95, Win98, & NT) systems, Microsoft Word is THE word processor used. A number of documents are created and distributed in Word format, assuming that most people have access to Micrsoft Word or a Word viewer. For those QLers that don't have access to Word, but do run across Word files, catdoc is the utility to convert the Word files into something more useable for the QL.
The catdoc zip file is available from Jonathan's web page 'The Dead Letter Drop' or through the normal freeware distribution channels. The distribution will fit (with a little room left) on a 720K floppy. Before unzipping the distribution, be sure you know how to prevent unzip from converting the dot (.) extensions to underlines. Since catdoc is originally a Unix application it will be expecting files with dot extensions. I unzipped the distribution before I knew of this and had to change a few files by hand.
Once unzipped, you will have a number of files and three subdirectories (src, charsets, & docs). The catdoc executable is found under the src subdirectory. I moved it to the main directory to make it easier to use.
Before running catdoc an Environment Variable is set for letting catdoc know where the character set files are located. Since I had unzipped catdoc on a floppy I set it as:

```
setenv "CATDOCLIB=flp1_
charsets_"
```

Now to run catdoc all you need is a Word file. Since I have Word 7.0 on my PC, I copied over my ToDo list (todo.doc) and let catdoc chew on it.
The simplest way to execute catdoc is this:

```
exec catdoc;"todo.doc"
```

This will take the file "todo.doc", convert it to ASCII, and display it on the screen. If you want to save the output to a file then executecatdoc like this:

```
exec catdoc;"todo.doc › 
todo_txt"
```

catdoc does some fairly simplistic reading of the Word file. I noticed in converting my ToDo file a bunch of extra information and text that I had deleted out of the file. It seems that Word keeps some of this version information in the file and when catdoc processes the file it appears.
When I converted a simple test file with no revisions, the output from catdoc looked better. I even added a table to the second Word document and catdoc was able to handle it.
Any output from catdoc will probably have to be cleaned up before it is presentable. The text file generated by catdoc can easily be imported in to Quill, cleaned up, and formatted to create a final document.
So, if you don't have access to Microsoft Word and need to read a Word file on the QL, catdoc is the tool for you. It may not generate a "pretty" document, but it will extract the text information from the Word document.
Those with Web access can get catdoc free from:
http://www.jrh.demon.co.uk
■

You and Your Programs - Just good Friends?

Geoff Wicks

Part 1 - Screen Design

I once took a newspaper that each day printed a political cartoon drawn by a reader. Usually the cartoons were either beautifully drawn but not funny, or funny but badly drawn. It was not often a cartoon was both funny and well drawn.

It is much the same with software. There are some brilliant programmers who can write clever programs and routines, but who know little about user-friendliness and program presentation. Others can produce attractive, easily used programs, but which are limited in what they do and sometimes full of bugs. I occasionally discuss with Roy Smith how we need to marry the "programmers" with the "presenters" in the QL community. If we could bring these talents together, QL software could blossom.

This series of articles is not about programming as such, but about good software design. About what you can do to make your programs more attractive and more user-friendly. I shall be formulating a number of "Friendly Software Rules."

My first commercial software product, Solvit-Plus 2, was initially distributed by Dilwyn Jones Computing. It was an interesting, and at times humbling, experience to see a amateurish program being whipped into shape by an experienced trader. Through this experience my programming and presentation skills improved greatly, which in turn has benefited the wider QL community.

Even after Dilwyn's work Solvit-Plus 2 had shortcomings. Some users were overwhelmed by the number of commands, and others felt some command names were illogical. I learnt from these comments and just under two years ago radically altered my program design.

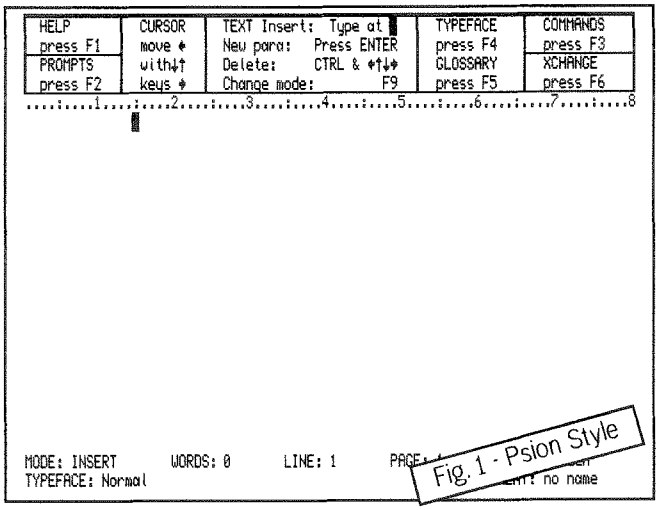


Fig. 1 - Psion Style

From comments I have received from both reviewers and users, it has been a successful redesign.

I set myself an ambitious target. I wanted to get away from a "QL look", and produce a distinctive JUST WORDS! house style.

In my opinion there have been two main QL looks, the "Psion" style and the "Pointer" style. Both are successful designs. The Psion style is characterised by green or white ink on black paper with commands at the top of the screen, and input at the bottom. The pointer look is characterised by white ink on a black background or black ink on a white background. The commands are usually placed in a narrow band at the top of the screen, usually on a background of green and white stipples or stripes.

The first thing you have to decide is basic screen colours. In mode 4 there are only 4 available colours, and this limits the design possibilities, particularly if you are not prepared to experiment with stipples. I soon concluded that for a different look I had to avoid the use of green. This meant using red, which is a difficult colour as it reproduces too harshly on some monitors and is barely discernible on others.

I solved this by making my main border a black and red stipple, and using pure red for internal borders. I then had the problem of a colour to go with red. Black or white contrast too harshly with red, and

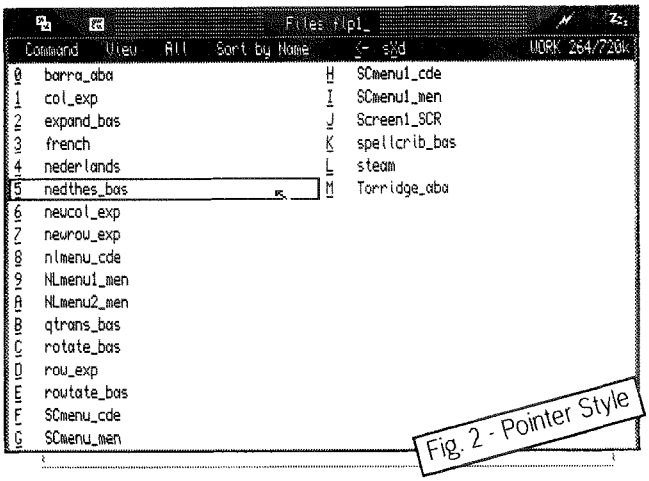


Fig. 2 - Pointer Style

after some experimenting I decided on a white and black

TF Services

superHermes

A major hardware upgrade for the QL

- All Hermes features (see below for list) PLUS full 19200 throughput on ser1/ser2 not affected by sound
- IBM AT keyboard interface (plus foreign drivers)
- HIGH SPEED RS232 industry standard two-way serial port. 4800cps throughput (supergoldcard - qtpi - zmodem) at 57600bps
- THREE low speed RS232 inputs (1200 to 30bps) Driver for SERIAL MOUSE supplied. Other uses include RTTY/ graphics tablet etc
- THREE spare I/O lines (logic) with GND/+5V
- Capslock/scrollock LED connector
- Turbo/keylock connectors
- 1.5k user data permanently storeable in EEPROM

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

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

Hermes available for £25 (£26/£24/£27) **Working ser1/2 and independent input, debounced keyboard & keyclick.**

superHermes LITE

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

Minerva

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

The ORIGINAL system operating system upgrade

OTHER FEATURES COMMON TO ALL VERSIONS

DEBUGGED operating system/ autoboot on reset of power failure/ Multiple Basic/ faster scheduler- graphics (within 10% of lightning) - string handling/ WHEN ERROR/ 2nd screen/ TRACE/ non-English keyboard drivers/ "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 IRC's

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

QL REPAIRS (UK only)

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

£27 including 6 month guarantee

QL RomDisq

Up to 8 mbyte of flash memory for the QL

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

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

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

MPLANE

A low profile powered backplane with ROM port

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

Cost..... **£34** (£36/£33/£35)

I2C INTERFACES

Connects to Minerva MKII and any Philips I²C bus

Power Driver Interface **16 I/O lines with 12 of these used to control 8 current carrying outputs (source and sink capable)**
2 amp (for 8 relays, small motors)..... **£40** (£43/£38/£44)
4 amp total (for motors etc)..... **£45** (£48/£43/£50)
Relays (8 3a 12v 2-way mains relays (needs 2a power driver)..... **£25** (£28/£23/£27)
Parallel Interface Gives 16 input/output lines. Can be used wherever logic signals are required.. **£25** (£28/£23/£27)
Analogue Interface **Gives eight 8 bit analogue to digital inputs (ADC) and two 8 bit digital to analogue outputs (DAC). Used for temperature measurements, sound sampling (to 5 KHz), x/y plotting**.... **£30** (£31.50/£29/£30)
Temp probe (-40°C to +125°C)..... **£10** (£10.50/£10/£11)
Connector for four temp probes..... **£10** (£10.50/£10/£11)
Data sheets..... **£2** (£2.50/£2/£3)
Control software & manual (for all I/F).. **£2** (£2.50/£2/£3)

QL SPARES

Keyboard membrane ..... **£12** (£12.50/£12/£13.50)
1377 PAL..... **£3** (£3.50/£3/£4)
Circuit diagrams..... **£3** (£3.50/£3/£4)
68008 cpu or 8049 IPC..... **£8** (£8.50/£7.50/£9)
8301/8302 or JM ROM or serial lead . **£10** (£11.50/£10/£11)
Power supply (sea mail overseas)..... **£12** (£17/£16/£21)
Other components (sockets etc) also available

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

19 Dec 98

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**

VISA MasterCard

stipple to give a greyish appearance to the bulk of the screen. Unfortunately it was too grey and dull, so I relieved the boredom by placing a bold white stripe at the top of the work area of the screen. In practice this stripe doubles up as a "message window".

After designing colours you then have to decide how the screen will be divided between say menus and work areas. I needed a place for three menus and a large work area. The menus have different functions and are active at different times during the use of the program. One of the menus would be used only occasionally so I could put into a pull down box, but the other two had to be on the screen alongside the work area. I placed these on the left hand side and at the bottom of the screen. I gave the three most used menu items extra emphasis by placing them in a box with a red/white stippled background.

These three commands are input commands. When they are used an input window opens not at the bottom of the screen, but in the work area alongside the commands. This is better for the eyes. When the program has completed its work, the results are printed in black ink on white paper, giving an unintended, but characteristic, slat effect when printed on the grey background.

Above the work area I put the program's name and copyright notices. This is on a background of a black and white stripe to contrast with the grey of the working area. As finishing touches I added my logo to the top right hand corner of the screen, and, very gimmicky, made my pointer the pen nib from the logo.

It may seem egocentric and narcissistic for me to write at such length about my program design, but this is for a purpose. Screen design is not done in five minutes. It is a lengthy process involving complex decision making, and I wanted to show the problems I had to solve and the decisions I took. The redesign of my house style took me longer than a month.

Now test yourself. Look at one of your own programs. Can you describe why you designed it in the way you did? Why you used those colours? Why you placed the menu where you did? Why you put the input area where you did? Why you put the working area where you did? If not, maybe it is time to think of a redesign.

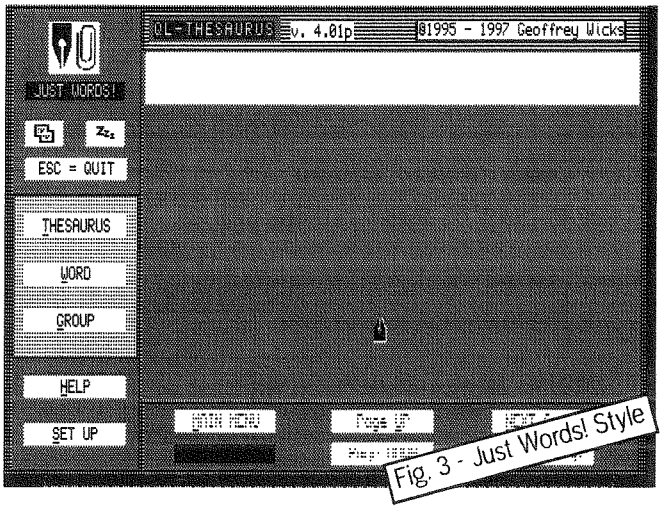


Fig. 3 - Just Words! Style

FRIENDLY SOFTWARE RULE: Allow ample time for your screen design.

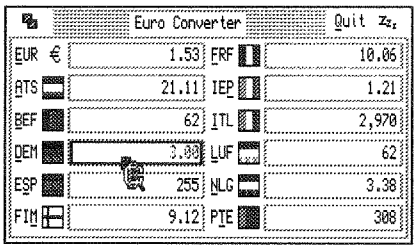
If you write pointer programs I can highly recommend Easymenu from Albin Hessler's Easyptr suite for help in designing screens. Using this program you can edit your design very rapidly. For example, you can change colours, resize or move boxes and see the effects almost immediately. Most important you can continue experimenting until you get it right. And if you later find your design not fully to your liking, it is still easy to make minor changes.

Next time: Everything about menus. Their contents and placing. ■

EURO Currency Converter V1.10

Review by Dilwyn Jones

Beginners Club of Italy have produced a neat and simple to use little pointer driven program for currency conversion between Euros and the currencies of the European Union states which are members of the European Exchange rate system. The Euro was theoretically introduced in 1999 in a majority of EU member states, with the exception of Britain, Denmark, Sweden and Greece. Although these countries do not yet use real Euro coins and notes, the Euro is in use for businesses, bank accounts and other non-cash uses.



Using the fixed exchange rate among member states, this program simply lets you specify the amount in one currency. For example, when the

program starts, it shows the equivalent amounts in the national currencies which equates to one Euro. DO (right mouse button click) on a currency, you can enter a different amount of that currency and the program displays the equivalent amount in all the other currencies. Simple really. And by left clicking on an amount, it is fed into the stuffer buffer ready for you to pick up in another application with the ALT-SPACE keypress. It's that simple.

You don't even have to configure this program, just unzip it and execute it, that's all there is to it!

The package comes with just 3 files, Euro_obj (the QLiberator compiled program), Euro_txt (brief instructions) and FILE_ID.diz, a short description file for bulletin boards, etc. The version I got was downloaded as file called EURO110_zip. You can get it from Beginners Club (Italy) web site at **http://www.geocities.com/SiliconValley/Lab/5011/**

The program covers the currencies of Austria, Belgium, Germany, Spain, Finland, France, Ireland, Italy, Netherlands, Portugal and Luxembourg.

Full marks for a nice, neat and easy to use pointer driven utility to meet the needs of the moment.

Electronic Publishing on the QL

Tim Swenson

As Editor of the QL Hacker's Journal, compiler of the Z88 Source Book and SuperBasic Source Book, and general collector of QL documentation, I have long been thinking about electronic publishing on the QL.

My professional career has been in Computer Support. This has given me access to the Internet for 11 years and the Web for 5 years, including exposure on how electronic publishing has expanded over the years and how the various technologies are used to implement it.

I find the neatest feature of electronic publishing to be the ease and low cost of creating and distributing information electronically. A person sitting on a desert island with a computer and Internet connection can create and distribute a document worldwide, with very little cost. The logistics of creating, printing, and distibuting has virtually dissappeard. Only the cost of creation is the important factor.

As much as the QL has been in the backwaters of the computing world, the concept of electronic publishing on the QL is just as valid as on any other platform. Due to the shrinking user community, electronic publishing is more important now than ever before. The user community is getting less and less able to support the cost of producing hard copy.

A key part of electronic publishing is to decide on a common electronic format to use. The format is broken in to a file format and a viewer of that format. Once a document is created in a particular format there must be a viewer to display the document. In the "real" computer world, the predominant publishing formats are: Text, Adobe Acrobat (PDF), PostScript, and HTML.

In deciding on a particular format we have to determine which formats are the best for creation and viewing on the QL.

Before going too far, I wanted to bring up the topic of HyperText. Traditional text files only have one path to follow when reading them. You start at the beginning and read through to the end. In paper documents, they may have a reference to a different section. You may see something like "For more details see Section 7", or something like this, and you can skip ahead to that section. This is a link from one part of the document to another. HyperText is a electronic way of building these links. In a HyperText document an electronic link is made between the reference and the section. In a HyperText document if you saw a reference to another section of the document, you would just click on it and go straight to the other section. Another term for HyperText is browsable. Browsable means that you can browse through a document or documents, making jumps at the different reference points.

In this article I'll cover the various File formats, the associated Viewers, and the advantages and disadvantages of each format.

Text File

The Text File (AKA ASCII text, Plain Text) is the lowest common denominator for a file format. A Text File created on a computer can be displayed on almost any other computer. It is the oldest format, and still remains a popular one, primarily due to its simplicity and its universality.

Creating

There are many text editors and word processors that can be used to create text files.

Even though word processors do not save their documents in text format, there is usually a way to export or get a text file out of a word processor. Most text editors are not really designed for document creating and do not have any word wrap features, except for MicroEmacs, so you will have to mind the margins yourself.

Viewing

Besides text editors, most word processors can read in text files and display them. There are also a number of text file viewers. Unfortunately the name "Viewer" seems to be very popular.

Viewer - A PE text file viewer written by Christopher Cave, called MView.

Viewer - By Dilwyn Jones. Besides having its own File format, it does display text files.

Advantages:

- Easy to Create. Any text editor and word processor can be used to create a text file. Given the number of different text editors on the QL, one is bound to fit your style.
- Universally Viewable. An ASCII text file can be read by 99% of the computers in the world. If there are any non-ASCII characters (those above 127), such as non-English characters or special graphic characters, then different computers will have different problems. In the MS-DOS world, it was popular to embed IBM graphics characters in text files to get a better look. This make displaying these files on non-MS-DOS systems rather difficult. ***[Recent versions of my text file viewer do have an IBM character set mode - Dilwyn Jones]***
- Many Native QL Viewers. Besides using text editors or word processors, there are other text viewers available on the QL.

Disadvantages:

- Limited Formating. There is only so much you can do in formating with a text file. You can't have different fonts (bold, underline) or different font sizes (large, small).
- Not Easy to Reformat. Unless you have a text editor that supports word wrap, importing a text file in to a word processor to edit and reformat, can be a fair bit of work.
- Non-Browsable. There is no way to create any hyperlinks in a text file.
- Text-only "graphics". Since a text file is only text and does not allow any graphics, drawing pictures is done only with text characters. Creating these text "graphics" can be time consuming and they can take up a bit of space.

Quill Document

Quill and its _DOC format has become the standard word processer and document format for the QL. Because Quill came with every QL, every QLer should be able to handle a _DOC file.

Viewing

Besides using Quill, there are some Quill _DOC file viewers available. DocView is a PE program that displays a _DOC file. This saves the time necessary to execute Quill or Xchange. A PE viewer for Quill documents has been written by Pal Monstand and Arvid Borretzen called, DocView 1.0.

Advantages:

- Easy to Create. We all know how to use Quill and create documents. Even for those that might not be familiar with Quill, it is easy to use and easy to format documents.
- Universal in QL community. Everbody should have a copy of Quill in one form or another. Xchange is now freely available.
- Editable format. No conversion is necessary to edit the document. Reformating is is very simple. Also supports formating such as bold, underlining, left and right justification, page numbering, and so on.

Disadvantages:

- Unknown in Non-QL Community. Unless they have the PC version of Quill, non-QLers will not be able to handle _DOC files.
- Non-Browsable. There is no way to build any hyperlinks in Quill.
- No Graphics. Quill does not support embbeded graphics.

DJ Viewer Format

Dilwyn Jones (DJ) has written a viewer that supports hyperlinks and PE PIC images. Called Viewer, it does more than just view text file. Besides hyperlinks and PIC images, the Viewer has commands for finding text, extracting a block of text, merging text files, and printing the text. The screen size is configurable along with the colors used.

Advantages:

- Could be Universal in QL Community. DJ's Viewer is freeware and freely distributable. To save data space, the Viewer need not be part of the document distribution, but available seperately to those that do not yet have it.
- Browsable. DJ's Viewer does support hyperlinks. The links are only to other files and shows the file name. This makes the document look a little cluttered, but with creative file naming, this limitation can be worked around.
- Graphics. DJ's Viewer supports PE PIC images. The images do not show up as part of the document, but hyperlinks are made to them, where they are displayed by themselves. This means that the Viewer

can either display text or PIC images, but not both at the same time.
- Easy to Create. There is very little to the Viewer format. It is a regular text file with some embedded commands for the hyperlinks. It only takes a few minutes to learn the embedded commands.
- Native QL Viewer. This means that it is relatively fast. Ported software usually suffers speed problems from not being written specifically for the QL. The Viewer is quick and a relatively small executable (about 63K)

Disadvantages:
- Unknown in Non-QL Community. This format is not used outside of the QL community, including the PE PIC image format. The text of the document can be read on other computers, but the hyperlinks will not work. There are tools available to convert the image files to more portable formats.

HTML/Lynx

HyperText Meta Language (HTML) is a text file format with embedded formating commands. It is the format used by web browsers. It was derived from SGML. As HTML has grown over the years, different versions of the language have been created. HTML have gone from HTML 1.0, HTML+, HTML 2.0 and HTML 3.0. Each version is an extension of the previous version and are backwards compatible. If you create a file in HTML 1.0, a browser that supports HTML 3.0 will be able to read it.

Lynx is a text-only web browser. It was originaly created for Unix users that only had access to simple text terminals. Two versions have been ported to the QL, one that views only local files and one that does connect to the World-Wide-Web via uQLx and the underlying network on the computer the emulator is running on.

Advantages:
- Known Outside QL Community. HTML is a standard format and is used outside the QL community. As more of the World comes to the Web, HTML is becoming more used.
- Browsable. HTML was expressively designed to handle hyperlinks and browsable documents. Links can be made in the same document or to other documents.
- Supports Graphics. Lynx supports graphics by calling another application to show them.

Disadvantages:
- Non-Native QL Viewer. Lynx is a very large application (600K executable). It takes a bit to start up and. It requires a fair amount of memory and CPU.
- Difficult to Create. Because HTML is a language, it takes some time to learn it. There are no HTML editors for the QL, so we are forced to actually learn the language.

There are two other HTML browsers available for the QL. A couple of versions of QMOSAIC were released. They were really beta releases and a lot of the features were not implemented or had problems. For some reason, development of QMOSAIC was stopped. ProWess comes with a HTML browser that supports multiple fonts. This is probably the best looking browser available for the QL, but it does require ProWess. In the future this browser may be more prevalent.

There are a number of tools available to assist in creating HTML documents on the QL:
- Quill to HTML. This is a printer driver that outputs HTML command tags instead of printer commands. The document is created in Quill and then printed via the driver to disk. It only supports a limited set of HTML. This would be a good tool to start the document. It can be refined by hand or by using another tool.
- HTMLMachine. This is a PE application that will run "on top" of any word processor or text editor. It pops up on command, the user selects the HTML tag that they want, and it is placed in the document where the cursor is located. This saves the user the time and trouble of memorizing the HTML tags. The user will still need to know exactly what tag they will need.

MicroEmacs HTML Macros

This is a collection of macros for MicroEmacs that place HTML tags in a document. This is a fairly sophisticated set of macros that can query for information needed, has a preset layout for frames (including how to handle browsers that don't support frames) and tables, and includes a short menu system for creating pages. If you already know how to use MicroEmacs then this is the tool to use. The HTML code generated by the macros is fairly complex and looks professional. If you need additional commands or tags supported, adding them is fairly easy, using the existing macros as examples. If you are new to MicroEmacs and are serious about creating web pages, it might be worth it to learn MicroEmacs for these macros.

Choosing the Right Format and Viewer

When choosing a particular format and viewer there are a number of different items you must consider.
- Audience. Audience is who you expect to read the docu-

ment. If you want the share the document with people outside of the QL community, then you must choose a format that will be useable to them. If your audience is only the QL community then you don't have to worry about using a portable format.
- Graphics. A key part of document. If graphics will be a key part of the document, then you should choose a format that will support the graphics you need.
- Level of Effort. If you are just going to type up a quick document, they you may not want to spend a lot of time making it look real nice. If you are creating a definitive document that will be around a while, then it may be worth it to spend some time making the final document look it's best.

When I compiled the Z88 Source Book, I chose to use a Text File format, primarily because it did not have any graphics and I wanted a universal format. In doing the SuperBasic Source Book, I will make it available as a Text File and as a Quill Document. Being a Quill document, the end user will be able to print it without worrying about where the page breaks will lie. If I have the time, I would like to do a DJ Viewer version, for those that are interested in reading and browsing the document on the screen.
■

The structure of _ABA files

Christopher Cave

In the course of enabling the text viewer MVIEW to display ABACUS spreadsheets (I at least still use ABACUS), I was forced to disentangle much of the structure of _ABA files. The Editor asked me to record my findings. The layout of _ABA files is shown below. It is NOT complete and I would welcome any filling in of omissions and any corrections of my errors and misunderstandings.
One of the good things about _ABA files (and as it happens LOTUS 1-2-3 and Symphony files) is that they record the calculated and displayed results for each cell as well as the formulae required. For a viewer, this means that one does not have to write a spreadsheet program (and in particular the expression evaluator) but rather just has to look up the values. In _ABA files, the non-empty cells are recorded as 14 byte structures of 4 types (see below for details). The structures are stored row-by-row at the end of the file just before the 2 byte EOF marker 0xFFFF.
In the case of simple numbers (type = 2), i.e. those not calculated by some formulae, the relevant structure is self-contained; the number is recorded in PSION 8 byte floating point in the last 8 bytes of the structure. Since this floating point format (see structure ps_float laid out below for details) is neither IEEE 64 bit nor QL 6 bytes format, I had to write a bit shuffling routine to convert the PSION format to IEEE format for C68 to use. This code is appended.
The other types (0, 1 & 3) each require reference(s) to strings or formulae. In the case of strings which are the results of calculations, two references are needed - that for the formula and that for the result. The references are 2 byte integers starting, rather curiously, from 0x14. The whys and wherefores of this remain unknown to me. The structures referred to are stored as an array immediately before the cell data structures and after the 34 unknown bytes in offsets 0x15A - 0x17B. The strings/formulae are stored in identical structures with only the variable number of characters comprising a string in the first case and tokenised algebra in the second. I do not know what the 3rd and 4th bytes (the second short integer) in the structures are for. The tokens, in so far as I have identified them, are listed below in the section headed OPCODES.

ABACUS - layout of _ABA files.

```
bytes 0x00-0x04 'ABM1'

byte 0x0a last column starting from 0
byte 0x0b last row starting from 0

byte 0x0e default text justification
byte 0x0f default numeric justification

bytes 0x10-0x11 control default format
(PLACES refers to number of decimal
places where appropriate):

default             03 00
general             03 00
integer-            12 00
integer()           0a 00
decimal             00 PLACES
percentage          05 PLACES
monetary-           14 02
monetary()          0c 02
exponent            01 PLACES
```

```
bytes 0x12-0x110 record the number of
 columns per row for up to 0xff rows
bytes 0x112-0x150 record the column
 widths+1

byte 0x151 controls auto-calculate on input
byte 0x152 controls 0=BLANK switch
byte 0x153 controls ROW/COLUMN order of calc
byte 0x154 controls display width 2=40  1=64
                                        0=80
byte 0x155 controls form feeds on printing
byte 0x156 controls gaps between lines on
                                    printing
byte 0x157 controls lines per page on
                                    printing
byte 0x158 is the monetary symbol
byte 0x159 is the printer paper width

bytes 0x15a-0x15b ? 0x01 00

bytes 0x15c-0x17b comprise 32 nulls

byte 0x17c is start of data storage
```

1st section is for algebra and strings
2nd section is a row-by-row list of numbers and placeholders for strings/algebraic expressions

If section 1 starts with a 2-byte zero it is empty. In section 2 such zeroes stand for empty rows. The total length of section 2 can be calculated from the columns data in bytes 0x012-0x110. Stepping back from the EOF marker defines the boundary of the two sections.

```
struct string
{
   short len;/* of remainder of structure */
   short ?;
   short string_len;
   char chrs[string_len]; /* N.B. strings do
NOT end in NULLS */
}
```

In section 2, the start of a row is signified by a short for the total length of that row's numeric data and algebra/string refs. i.e. no_of_columns X 14.

Each used cell occupies 14 (0x0e) byte data field:

```
struct datafield
{
   char column; /* counting from 0 */
   char type; /*
      0 string-valued reference
      1 string
      2 number
      3 numeric valued reference */
   char format_byte; /* constructed from
            byte-pairs above - 1st+16*2nd */
   char justify; /*LCR text195
                  num021                  */
   union {
      struct {
         short ref1; /* algebra */
         short ref2; /* string result */
         char spare[6];
      } string_val;
      struct {
         char ?[2];
         short ref;
         char spare[6];
      } string;
      struct {
         char ?[2];
         ps_float val; /* PSION 64-bit
              floating point representation */
      } number;
      struct {
         short ref; /* algebra */
         ps_float val; /* numeric result -
                          c.f. number.val */
      } num_val_ref;
   };
}

struct ps_float {
byte1.bit7 = mantissa sign
byte1.(bits6-0)&byte2.bit7 = 8-bit exponent
offset by 0x81
byte2.(bits6-0)&(bytes3-8) = 55-bit mantissa
}

EOF - 2 bytes 0xff 0xff
```

N.B. 1) Split windows are not saved!

ABACUS ALGEBRA

A line of algebra is:

```
struct {
   short length_of_this_structure;
   short ?;
   short length_of_algebraic_formula;
   char algebraic_formula[];
};
```

PSION to IEEE floating point format

```
;===========================================
;
; converts 8-bytes corresonding to PSION 64
; bit floating point to IEEE64 floating point.
;
; C-equivalent defn.
; double ps_to_ieee64(*double);
;
; (c) C.J.Cave Oct. 1998
;
;===========================================

        .text
        .even

.globl _ps_to_ieee64
```

```
_ps_to_ieee64:
        movem.l d2-d4/a0,-(sp)

        move.l  20(sp),a0
        move.l  (a0)+,d1 ; d1 has first 4 bytes
        move.l  (a0),d2 ; d2 has last 4 bytes

        move.l d1,d3

        moveq #2,d4
12:
        lsr.l #1,d3
        roxr.l #1,d2
        dbra d4,12

; d2 is last 32 bits of mantissa

        andi.l #0x000fffff,d3 ; d3 is first 20 bits of
                                                        mantissa
        lsl.l #1,d1
        bcc 10
        ori.l #0x80000000,d3 ; add sign bit if appropriate

10:
        swap d1

        lsr.w #8,d1 ; 8-bit PSION exponent in bottom byte
        addi.w #(0x3ff-0x81),d1 ; adjust for different
                                                exponent biases
        lsl.w  #4,d1 ; exponent in 12 bits 15-4 inc. of d1

        swap d3
        or.w  d1,d3
        swap d3
        move.l d3,d0
        move.l d2,d1

11:
        movem.l (sp)+,d2-d4/a0
        rts

        end
```

OPCODES

+	0x07
-	0x08
*	0x09
/	0x0a
^	0x0b
args. end	0x12
args. sep	0x13
EOL	0x14
date	0x15
days	0x17
str	0x1c
deg	0x1f
len	0x20
rad	0x21
atn	0x22
code	0x23
cos	0x24
exp	0x25
int	0x26
log	0x27
pi	0x28
sgn	0x29
sqr	0x2a
tan	0x2b
val	0x2c
sin	0x2d
instr	0x2e
chr	0x30
month	0x33
time	0x36
if	0x3c
width	0x3d
index	0x3f
col	0x40
row	0x41
sum	0x43
ave	0x44
count	0x45
max	0x46
lookup	0x47
npv	0x48
irr	0x49
min	0x4a
rel. ref	0x50
abs. ref	0x51
range	0x56

About "Classic Computer Club"

by Ian Pizer

On a BBC TV program there was shown the complete simulation of the old Pegasus Computer on a powerful computer by members of The Classic Computer Club. You could see on the screen an image of the Pegasus control panel, the display screen, the teletype machine. The mouse of the simulating computer could manipulate the Pegasus operating switches; paper tape programs could be put into the reader and made to run; etc.; as though you were using the real Pegasus. Not sure if sound was included. It was a very impressive example of simulation of both software and hardware.

A real hardware Pegasus computer has also been revived to a working order by the Club which keeps models of many extinct computers including some Sinclairs.

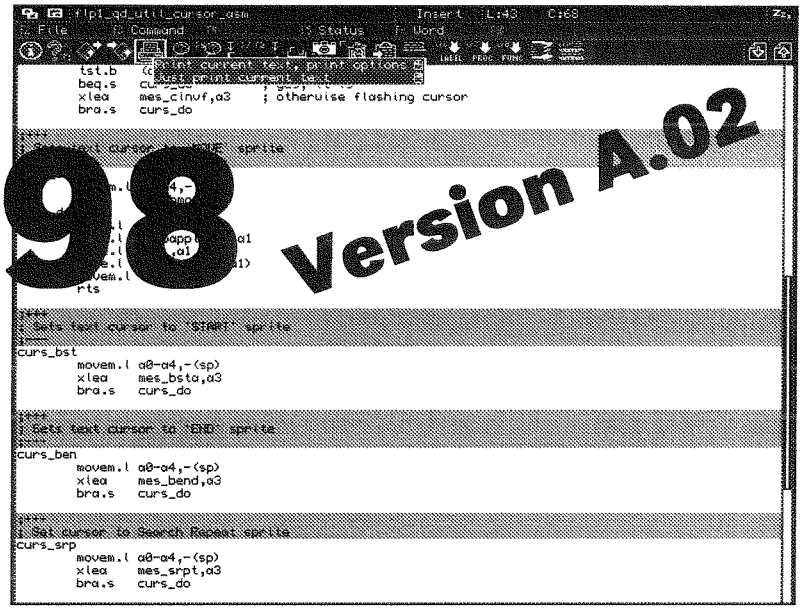
If you look at the WWW site of Classic Computer Club you will find a list of most current computers but nothing for Sinclair nor QL nor AURORA nor Q40. Should not QUANTA or some body take action? I did not find their address.

■

JOCHEN MERZ SOFTWARE

Im stillen Winkel 12 • 47169 Duisburg • Germany
☎ 0203-502011 (Fax 0203-502012 Mailbox 0203-502013 & 502014)
http://www.j-m-s.com/smsq/ email: smsq@j-m-s.com

QD 98 Version A.02

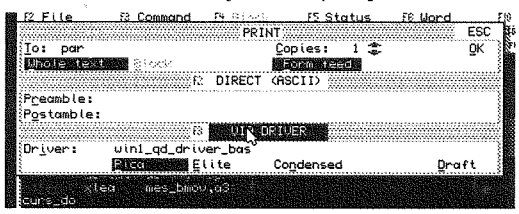


This ad does not give a list of all the features of our excellent Editor QD98, it will mainly list the new features which were introduced from the previous version QD Version 9 to the current one.

Here are its main features: "Hints" are given on menu items and the toolbar. Quite useful, because we have added many extra items and also added some additional features on "DO"ing some existing icons. Of course, this feature can be turned off.

The toolbar can be turned off and on while QD is running. A number of QL users asked me to do this because they gain two extra lines - it is done now!

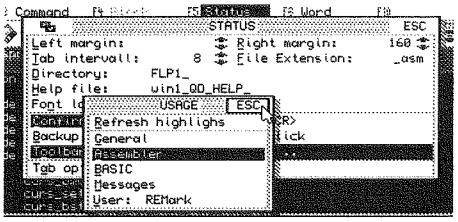
Lines can be ordered, you can specify the start column in the text which is used as the sort criteria.



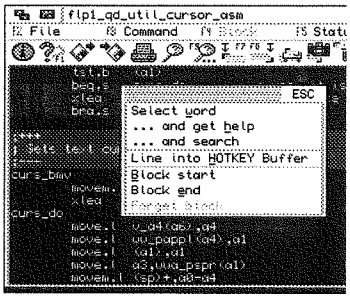
The GOTO label/proc/fn are now better accessible. We have also introduced a GOTO User-defined list. This means, you define a string for which QD looks and all the lines containing this text are put into the GOTO list (in fact, only the bit of the line after the string). This is very useful to create lists of all sorts, just use REMarks or other language-dependent combinations (C...)

We have added scroll arrows next to every numerical value entry in all menus - you will find this very useful if you use the mouse quite a lot.

PRINTing has been completely redone. You can print as before, or via driver. You can program the driver yourself, because it can be a BASIC filter. BASIC and Assembler examples are on the disk.



Line highlights are now possible and VERY, VERY useful. You tell QD what kind of highlights are required and it will highlight these lines in red paper. It can, for example, highlight all assembler comments (; or *), all BASIC Functions and Procedures (which is a kind of visual "folding"), or any lines with ">" in the first three characters (useful for message editing, PBOX etc.) and, of course, user-defined string. This will highlight all lines containing the string you gave - quite a neat way of doing a search, if you think about it.



"Context" menu which can be popped up with the right mouse button has also been introduced. This is much easier to handle than previous cursor-dependent functions.

These are the major changes, plus various minor ones, of course. You will get a new manual, not just additional pages.

The upgrade price is DM 39,90 for owners of QD9, and DM 49,90 for owners of older versions. Please return master QD disk for upgrade. A new QD98 still costs only DM 125,-

TERMS OF PAYMENT

Postage and package **[Germany]** DM 8,99 (if total value of goods is up to DM 50,- then only DM 5,99). **[Europe]** DM 14,50 (if total value of goods is up to DM 50,– then only DM 9,50). **[Overseas]** between DM 14,50 (1 item) and DM 35,– (maximum). All prices incl. 15% V.A.T. (can be deducted for orders from non-EEC-countries). E&OE. Cheques in DM, £'s, Eurocheques and Credit Cards accepted.

EUROCARD. VISA MasterCard

QuillRTF - A Review

By Timothy Swenson

While checking out new QL web pages, I ran across a utility written by Pedro Reino of Spain. The program, called QuillRTF, converts Quill files to Rich Text Format (RTF) files. RTF was an early format created to share formatted documents between different word processors. RTF does a fair job of keeping the formatting information of a document, such as boldface, underlining, paragraphs, and so on.

I have used plain ASCII text to convert Quill documents to Microsoft Word. Once the document is in Word I would have to clean the document up and rejoin all of the paragraphs from single lines. Using RTF would save me hours of reformatting.

I downloaded Pedro's program and unzipped it. Giving that Pedro is from Spain, the program and the documentation is in Spanish. My Spanish is limited to ordering off a take-out menu, so I could not read the documentation. But, not reading the documentation has not stopped me before.

I executed the program and was preseted with a Spanish Menu. One menu item "Fichero" was highlighted. Maybe it meant "files"? I hit the return key. A submenu came up, also with a menu item highlighted (Convierte). I hit the return key. The program read FLP1_ and now a directory of files on FLP1_ displayed. Ah, now I should be able to select a file to convert. I used the arrow keys to move down until I had highlighted an example _doc file. I hit the return key. Now the program started reading the file, printing out a byte count as it went along. After a minute or so, the program stopped and I was back to the submenu. Guessing it was done, I hit the ESC key until I had exited the program. I now did a directory of FLP1_ and found a file with the same name as the _doc file I selected, but it had an _RTF extension.

I copied this file to an MS-DOS disk, sneaker netted it over to my PC, and read the file into Word (telling Word it was an RTF file). Now I had my document, nicely formatted, in Microsoft Word. Ah, another successfull attempt at blundering my way through a program.

If you are doing any conversion of Quill documents to other word processors and they can handle RTF files, then Pedro's QuillRTF program is just what you need. It's not the fastest program, but the minute it takes to covert the file saves at least 10-15 minutes of reformatting. I figure it takes me about 15-30 seconds to reformat an average paragraph, so any document over a single page would be faster to use QuillRTF than to do it by hand.

As for what else the program can do, I have not a clue. But, if you can read Spanish, mabye you can find out.

The site address is

http://www.anit.es/pedro

Assembly Language Programming - Part 4

Norman Dunbar

Initial Ramblings!

Another week, another city, another hotel. This week I am in Manchester for a week of Oracle database training, so this part of the series has been written out in longhand using the original word processor, the ball point pen! (No laptop this trip, oh dear!) Enough of this, on with the article.

Last Issue's Bugs

In the previous instalment (part 3) there was a glaring and very silly error on page 25, where the instruction 'MOVEQ #1000,D1' appears. This is totally wrong as the MOVEQ instruction only allows signed 8 bit values from -128 to +127. The correct instruction is 'MOVE.W #1000,D1'. Sorry about that.

Logical Stuff

Logic is the heart of all computer systems: well, all digital ones anyway. Logic is how the central processor works. The 68000 series of processors are no exception and in the instruction set, there are a few logical operations that can be carried out. This article discusses those instructions.

Tie The NOT

The logical NOT instruction is probably the simplist of all this family of instruction. It converts the destination address from its current state of ones and zeros into the exact opposite to zeros and ones. The format is:

NOT.size destination

Size can be byte, word or long. The instruction carries out a 'ones compliment' of the destination address. If you remember back to the discussion of 'Twos compliment' numbers earlier on in the series, you will remember that converting a positive number to negative involved flipping all the zeros and ones and then adding one to the result. The NOT instruction carries out the first part of flipping all the ones and zeros over.

If D0.W holds the value of $0001 then after a NOT.W D0, it will hold the value $FFFE. All the original zeros have become ones and vice versa.

NOT must not be confused with the arithmetic NEG instruction which carries out a 'twos compliment' negation of a value. (D0.W in the above example would become $FFFF which is equivalent to NOT.W D0 followed by ADDQ.W #1,D0)

NOT affects the flags in the following way:

N - set if the result becomes negative and the most significant bit becomes a 1. Cleared otherwise.

Z - set if the result is zero, cleared otherwise.

V - always cleared and you cannot create an overflow by inverting the bits.

C - always cleared & there is no carry generated by flipping bits.

X - not affected.

This OR That

Next up in the logical family is the OR instruction of which there are a few. OR is quite different from NOT in that it needs to have two operands in order to be used. The format of the OR instruction is:

OR.size source,Dn or

OR.size Dn,destination

Note that in this form of the instruction either the source or the destination must be a data register. The size can be byte, word or long.

This is the 'inclusive or' instruction and there is also an 'exclusive or' variety which we will see later on in this article. An inclusive or works according to the following 'truth table':

Source	Destination	Result
0	0	0
0	1	1
1	0	1
1	1	1

Simply imagine each individual bit in the source is being OR'd with the same bit in the destination. The result and which will be stored in the destination bit and will always be a 1 if one OR other of the two bits being processed is a 1. If both are zero then the result will also be zero.

An example

D0.W contains $AAAA and D1.W contains $6543 the instruction

```
OR.W D0,D1
```

will result in D1.W being set to $EFEB and D0 will remain unchanged. How does this work? In binary:

```
D0 = $AAAA = 1010 1010 1010 1010
D1 = $6543 = 0110 0101 0100 0011
```

So using the truth table above, the result will be

```
D1 = $EFEB = 1110 1111 1110 1011
```

The flags affected by OR are exactly the same as for NOT above.

The OR Immediate format of the OR instruction has the format:

ORI.size #data,destination

And can be byte, word or long sized. It is used when the source value in the OR is immediate data as opposed to a register or memory address. Some, but not all, assemblers will allow you to write:

OR.size #data,destination

But the actual instruction assembled will be ORI instead. Again the flags are affected as for NOT.

ORI #data,CCR is an instruction that is used to set the flags to a set of known values as supplied in the immediate data. This instruction only uses bits 0 through 4 of the data supplied as the other bits are not used in the 68008. As it is possible that future processors may introduce other flags, you are always best to make sure that bits 6 through 7 are zero when using this (and the following) instruction. That way, you won't cause any 'strange effects' on a different processor.

The flags are set as:

C - set if value in bit 0 of the data is a 1 otherwise unaffected.

V - set if value in bit 1 of the data is a 1 otherwise unaffected.

Z - set if value in bit 2 of the data is a 1 otherwise unaffected.

N - set if value in bit 3 of the data is a 1 otherwise unaffected.

X - set if value in bit 4 of the data is a 1 otherwise unaffected.

ORI #data,SR does a similar job to the above but affects the entire status register. The other difference is that the processor must be running in Supervisor mode for this instruction to be carried out. If it is not then a privilege exception will be generated and this will hang the QL (usually)

As above, the flags are set according to the data and bits 0 to 4. The rest of the status register is set as follows:

T (trace) is set if value in bit 15 of the data is a 1 otherwise unaffected.

S (supervisor) is set if value in bit 13 of the data is a 1 otherwise unaffected.

The value in bits 10, 9 and 8 can be anything from 0 through 7. This is OR'd with the current value in the interrupt level bits of the SR and the new value becomes the new interrupt level mask.

Once again, all unused bits must be zero in the data to prevent unpredictable results on different processors (it is called defensive programming.).

This instruction can be used to turn off all interrupts except level 7 these are known as non-maskable interrupts as they cannot be turned off.

```
TRAP  #0
ORI   #$0700,SR
```

This sets the QL so that only a level 7 interrupt will be actioned. The only problem here is that CTRL ALT and 7 activate a level 7 interrupt and effectively hangs your QL. After the above instructions, the supervisor mode is still in effect. (Work it out in binary!!) To exit from supervisor mode

```
ANDI  #$07FF,SR
```

would need to be done and this leads us nicely into the AND family.

This AND That.

In a similar manner to the OR instruction, the AND instruction needs two operands to work on to get a result.

The format of the AND instruction is:

AND.size source,Dn or

AND.size Dn,destination

Note that as with the OR instruction, this form of the instruction requires either the source or the destination to be a data register. The size can be byte, word or long.

AND works according to the following 'truth table':

Source	Destination	Result
0	0	0
0	1	0
1	0	0
1	1	1

Simply imagine each individual bit in the source is being ANDed with the same bit in the destination. The result and which will be stored in the destination bit and will always be a 1 if and only if both bits being processed are 1. If either are zero then the result will also be zero.

Using the same example as for OR above:

D0.W contains $AAAA and D1.W contains $6543 the instruction

```
AND.W  D0,D1
```

Will result in D1.W being set to $2002 and D0 will remain unchanged. How does this work? Once again, in binary:

```
D0 = $AAAA = 1010 1010 1010 1010
D1 = $6543 = 0110 0101 0100 0011
```

So using the truth table above, the result will be

```
D1 = $2002 = 0010 0000 0000 0010
```

The flags affected by AND are exactly the same as for NOT above.

The AND instruction has the same variations as the OR instruction. These being:

ANDI.size #data,destination

ANDI #data,CCR is an instruction that is used to reset or clear some or all of the flags. The flags are reset as follows:

C is reset if value in bit 0 of the data is a 0.
V is reset if value in bit 1 of the data is a 0.
Z is reset if value in bit 2 of the data is a 0.
N is reset if value in bit 3 of the data is a 0.
X is reset if value in bit 4 of the data is a 0.

ANDI #data,SR works upon the entire status register.

As above, the flags are reset according to the data and bits 0 to 4. The rest of the status register is reset as follows:

T (trace) is reset if value in bit 15 of the data is a 0.

S (supervisor) is reset if value in bit 13 of the data is a 0.

The value in bits 10, 9 and 8 is ANDed with the current value in the interrupt level bits of the SR and the new value becomes the new interrupt level mask.

All unused bits should be one in the data to prevent unpredictable results on different processors.

This instruction can be used to exit from supervisor mode. The instructions:

```
TRAP  #0
ANDI  #$D7FF,SR
```

Would set the QL so that supervisor mode was first switched on (by the TRAP #0) and then only the supervisor bit in the SR was cleared (bit 13) so the QL would revert to user mode. All other modes and interrupt levels and flags would remain unchanged.

Professional & Graphical Software

ProWesS

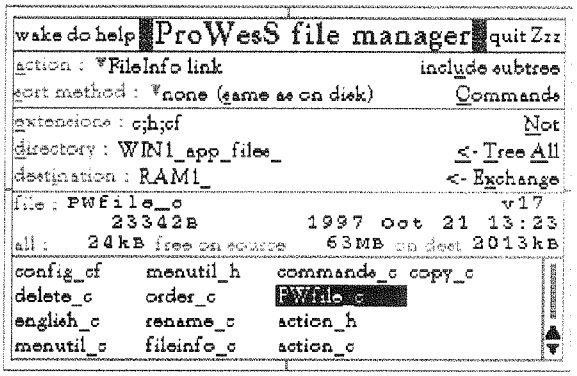
ProWesS is a new user environment for the QL. ProWesS is short for "PROGS Window Manager", but it is much more than that. Apart from a new window manager, it contains all the system extensions from PROGS, and is essential if you want to run programs which need these extensions.

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

Another important aspect of ProWesS is the possibility to allow programs to automatically install themselves on your system, and to be able to run them without resetting the system. This means that, when you get a new program, all you have to do is insert the disk and indicate "start the program in flp1_", a menu option in the "utilities" button. To install a program, you indicate "install software", and the software can be added to your system. This way, you don't need to know how to write a boot file to use the multi-tasking capabilities of your computer.

ProWesS includes many programming libraries. These include syslib, an interface to the operating system, PROforma, a vector graphics system, allowing rendering both on screen and on paper (via a printer driver). The DATAdesign engine is also part of ProWesS. It is a relational database system with a bonus, as you don't even need a key field. You get a powerful record at a time data manipulation extension to the language you already use. Of course it also includes ProWesS itself, the new resolution independent window manager.

New ProWesS application
a powerful and very user friendly file manager



LINEdesign

Create artistic drawings, technical drawings, process bitmaps (even scale and rotate them!), and any kind of vector drawings. You can use grpahics objects to create the most fabulous drawings ever seen. Because LINEdesign is a vector drawing program, any part of the picture can be moved, scaled, rotated, slanted without any loss of precision or resolution. In LINEdesign, pictures are device independant, meaning that the printout will be the same on any printer (e.g. same size and position).

LINEdesign is good at handling text. You can easily put titles and full paragraphs on the page. All the fonts can be displayed at any size, rotation, etc. All the fonts which are available to ProWesS can be used in LINEdesign.

LINEdesign is a drawing program, but it can also be used by people who are not good at drawing. LINEdesign is a great program for making leaflets, posters, and any kind of printed work. Lots of clipart and extra fonts are available from public domain libraries and BBS's. You can even import Adobe Illustrator files.

PFlist

Easy to use program to create listings on any printer (especially inkjet and laser). This ProWesS application allows you to indicate the files which have to be printed. Each column contains a footer which can include the filename and filedate. The listings always allow perforation. PFlist can create your listings in two columns and in landscape (or both).

fsearch

File search utility with many useful options, like the choice to search only files with a certain extension, and whether or not the directory tree has to be scanned. All occurences of the searchstring will be displayed with line number or offset. You can also use special matching features, like case dependent, matching a space with a stretch of whitespace, and searching for a word dilimited string.

font-utils

manage your font collection. You can preview fonts on screen, see what characters exist in a font and convert Adobe Type 1 and similar fonts for use in ProWesS.

DATAdesign

Never before has it been so easy to create, fill in and maintain your personal databases. To start a new file, just type the names of the fields. To add or delete a field, no problem, just do it. To change the name of a field, just indicate it. You can choose which fields are displayed and also which records. You can have a hidden comment for each record, look at the file in tabulated form and transfer data to the scrap or hotkey buffer. Files can be memory based (for speed) or disk based (for safety).

new address !!

Dr. Fr. Hemerijckxlaan 13 /1
2650 Edegem
Belgium

tel : +32 (0)3/ 457 84 88
fax : +32 (0)3/ 458 62 07
email : joachim@club.innet.be
www : http://www.club.innet.be/~year2827

ProWesS - BEF 2400 | DATAdesign - BEF 1200 | LINEdesign - BEF 1200 | PWfile - BEF 900 | fontutils - BEF 1200 | PFlist - BEF 600 | fsearch - BEF 600

Payment terms :

You have to run ProWesS to make LINEdesign, DATAdesign, fsearch, fontutils and PFlist work (even though DATAdesign uses wman).

All our software is normally supplied on high density (HD) disks. However they can be obtained on double density (DD) disks at an extra costs of BEF 100. To use ProWesS and any of our other packages, you need a system with at least 2MB of memory. You should have a harddisk although a two disk system will also work. The use of SMSQ/E is strongly recommended for optimal use of ProWesS.

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

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

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

copies	*priority mail* Belgium	Europe	World	*ordinary mail* Belgium	Europe	World
one	100	200	240	100	120	145
two	135	340	420	135	190	230
3 or 4	160	560	770	160	310	395
5 to 8	185	870	1250	185	550	705
more	295	1130	1610	295	800	1030

All prices are in BEF, including 21% VAT

Exclusive OR Instructions.

Having dealt with the inclusive or instructions above, it is now time for the exclusive or instructions. This has the format:

EOR.size Dn,destination

Where size can be byte, word or long. Notice this time that EOR source,Dn is not permitted? I wonder why? (I don't know and does anyone?)

This instruction also sets the flags as per the NOT instruction. In the truth table for inclusive or, there was a 1 bit set in the result when there was a 1 in either the source or destination or both. Exclusive or is different and only allows a 1 in the result when there is a single 1 in either the source or destination. As follows:

Source	Destination	Result
0	0	0
0	1	1
1	0	1
1	1	0

Using the same example as OR and AND above we now have the following:

D0.W contains $AAAA and D1.W contains $6543 the instruction

```
EOR.W  D0,D1
```

Will result in D1.W being set to $CFE9 and D0 will remain unchanged. How does this work? Once again, in binary:

```
D0 = $AAAA = 1010 1010 1010 1010
D1 = $6543 = 0110 0101 0100 0011
```

So using the truth table above, the result will be:

```
D1 = $CFE9 = 1100 1111 1110 1001
```

One feature of EOR is that if you EOR the result of a previous EOR with the same value again, you get back to the original value. Using this code:

```
MOVE.W #$AAAA,D0
MOVE.W #$6543,D1
EOR.W  D0,D1
EOR.W  D0,D1
```

Wil return us to the state we were in before the first EOR, in that D1 will once again hold the value $6543. Try to work it out for yourselves using the example above as a guideline.

This can be used in a sort of 'Pretty Bad Privacy' program where data is encrypted using EOR. The following small program demonstrates this.

```
START MOVEQ  #7,D0
      LEA    DATA_STUFF,A1
      MOVEQ  #100-1,D1
LOOP  EOR.B  D0,(A1)+
      DBRA.S D1,LOOP
      RTS
DATA_STUFF Put 100 bytes of data here!
```

The LEA instruction is a new one and will be discussed soon. Suffice to say that it simply loads the address of the label 'data_stuff' into the address register named. This must be used in QL programs as they have to be able to run at any memory address - position independant code. The LEA instruction allows this.

The above code is very simple and assumes that there is exactly 100 bytes of data stored in memory at the location labelled 'data_stuff'. To encrypt the data, simply call the routrine at label 'start' and 100 bytes will be encrypted. To decrypt it, simply call 'start' again and the data will be restored. This is easily cracked because of the use of a single byte to encrypt the data so don't go using it for anything you value!!!!

EOR has the usual variations:

EORI #data,destination

EORI #data,CCR is an instruction that is used to change some or all of the flags. The flags are changed as follows:

C - changed if the value in bit 0 of the data is a 1.
V - changed if the value in bit 1 of the data is a 1.
Z - changed if the value in bit 2 of the data is a 1.
N - changed if the value in bit 3 of the data is a 1.
X - changed if the value in bit 4 of the data is a 1.

EORI #data,SR works upon the entire status register.

As above, the flags are changed according to the data and bits 0 to 4. The rest of the status register is chgaged as follows:

T (trace) is changed if value in bit 15 of the data is a 0.

S (supervisor) is changed if value in bit 13 of the data is a 0.

The value in bits 10, 9 and 8 is EOR'd with the current value in the interrupt level bits of the SR and the new value becomes the new interrupt level mask.

Shifting And Rotating

There are 4 shift and 4 rotate instructions, 2 going left and 2 going right.

ASL and ASR are arithmetic shifts while LSL and LSR are logical shifts. What is the difference? Taking the logical shifts first we have:

LSL.size,Dx,Dy or
LSL.size #data,Dy or
LSL ‹address›

LSR has the same format. The size can be byte, word or long. What happens is that the data in the register and which must be a data register for the first two. For the last format, the size has

to be WORD and the memory at that address is shifted by a single bit. The bit that is shifted 'out' of the register is placed into the C and X flags, while the 'vacant' bit is filled with a zero.

Consider this code fragment:

```
    MOVE.B #$81,D0
; D0.B is 1000 0001
    LSL.B  #1.D0
; Now it is 0000 0010 and C and X are 1
    MOVEQ  #5,D2
    LSL.B  D2,D0
; Now D0 is 0100 0000
```

Shifting the opposite way gives this:

```
    MOVE.B #$81,D0
; D0.B is 1000 0001
    LSR.B  #1.D0
; Now it is 0100 0010 and C and X are 1
    MOVEQ  #5,D2
    LSR.B  D2,D0
; Now D0 is 0000 0010
```

LSR is a quick way of multiplying an unsigned number by 2 for each bit shifted. LSR is a quick way of dividing an unsigned number by 2 and but the fractions are lost. Another example.

```
    MOVEQ  #8,D0  ; D0.L holds 8
    LSL.L  #1,D0  ; D0.L now holds 16
    LSL.L  #2,D0  ; D0.L now holds 64
    MOVEQ  #10,D0 ; D0.L holds 10
    LSR.L  #1,D0  ; D0.L now holds 5
    LSR.L  #1,D0  ; D0.L now holds 2
and note the remainder is 'lost'
```

When specifying the number of shifts as immediate data, only values from 1 to 8 can be used. If the number of shifts required is greater than this, then a register counter has to be used. When shifting memory, the shift is always a single bit.

After a shift in either direction the flags are set as follows:

N is set if the result became negative (MSB set to 1), cleared otherwise.

Z is set if the result became zero, cleared otherwise.

V is always cleared.

C is set to the LAST bit shifted out, cleared if the shift count was zero.

X is set to the LAST bit shifted out. UNAFFECTED if the shift count was zero.

The arithmetic shifts preserve the sign of the value but duplicating the previous value of the sign bit in the new sign bit.

Arithmetic Shifts are not Logical Captain ...

These instructions have the same format as the logical shifts:

ASL or

ASR.size #data,Dn ASL or

ASR.size Dx,Dn ASL or

ASR ‹address›

The instructions operate on long, word or byte sized data. The ‹address› version, however, only acts on WORD sized data.ASL or Arithmetic Shift Left appears at first glance to do exactly the same as LSL, however, the flags are slightly different afterwards (see below).

ASR or Arithmetic Shift Right, or the otherhand, preserves the sign bit of the value being shifted by duplicating it into the 'new' sign bit. So if the sign bit was 1 before the ASR it will still be 1 after and if it was 0 before it will be 0 afterwards.

If D0.B is $F0 or -16 then whereas LSR.B #1,D0 would result in D0.B being $7F or +120, following ASR.B #1,D0, D0.B would be $F8 or -8 so the sign was preserved by the Arithmetic shift but not by the logical shift.

In these instructions, the ASL #data,Dn format is used when the number of shifts is known and is between 1 and 8. If it is required to shift more than 8 (or less than 1!) then a register counter must be used - the ASL Dx,Dn format is used. This is also used when the amount of shifts has been calculated during the running of the program and cannot be specified at programming time. To shift memory, only the ASL ‹address› is allowed and the shift is always a single bit.

For example, to shift D0.L left arithmetically by 4 bits, do this:

```
    ASL.L  #4,D0
```

To shift it 10 places right, do this:

```
    MOVEQ  #10,D1
    ASR.L  D1,D0
```

In this case, D1 has to be used as a counter. Finally, to shift memory adddress $28000 (byte sized) left 2 places, involves loading the byte sized contents into a data register, shifting the register and storing back in memory. This is because a memory shift would have shifted the byte at $28000 and the byte at $28001 as memory shifts are always WORD sized. (I didn't actually know this until I checked the manual - we can all learn something new!) So this code will NOT work correctly:

```
      MOVEQ  #2-1,D0
SHIFT ASL    $28000
      DBRA   D0,SHIFT
```

but this will:

```
MOVE.B  $28000,D0
ASL.B   #2,D0
MOVE.B  D0,$28000
```

The flag settings for **LOGICAL** shifts are as follows:

N - set is the result is negative, cleared otherwise (ie the top bit)

Z - set if the result is zero, cleared otherwise

V - always cleared

C - set if the final bit shifted was a 1, cleared otherwise

X - set to the final bit shifted out - ie 1 or 0 - but UNAFFECTED if the shift count was zero.

For **ARITHMETIC** shifts, the flags are as follows:

N - set is the result is negative, cleared otherwise (ie the top bit)

Z - set if the result is zero, cleared otherwise

V - set is the sign bit changed AT ANY TIME during the shift, cleared otherwise

C - set to the final bit shifted out, cleared if shift count was zero

X - set to the final bit shifted out - ie 1 or 0 - but UNAFFECTED if the shift count was zero.

So while the LSL and ASL shifts look remarkably similar, the V (overflow) flag is always cleared for Logical shifts but it shows if the sign changed during the shift for Arithmetic shifts.

What this means is that if D0.B (my favourite register!) holds $A5 (-91) or %10100101 (binary), and was shifted left arithmetically by 2 bits, it will become %01001010 after the first shift. This has changed the sign so the V flag will be set. After the second shift the sign will have returned to negative

%10010100 - but V will still be set to show that it changed during the shifting. Once the V flag is set, it is not unset by any further shifts in the instruction.

Rotates are similar to shifts, but the bit(s) that 'fall off' the end are replaced at the other end. So rotating left causes bits lost from the left end to be added back in at the right end.

The rotate instructions have the following syntax:

ROL.size #data,Dn

ROL.size Dx,Dn

ROL.size ‹address›

There are also ROXL and ROXR which include the X flag in the rotate. While I have used ROL to illustrate the instructions the three formats are valid for ROR, ROXL and ROXR as well. As with the shift instructions byte, word and long sized data can be rotated although the ‹address› form of the instruction only allows WORDS to be rotated.

Once again, if the count is between 1 and 8, use the #data form. For rotates of a calculated number, use the ROL Dx,Dn form.

What is the difference between a shift and a rotate? In a rotate, you never lose any of the bits, they just end up on the other end of the value.

ROL.B #1,D0 copies bit 7 of D0 into the C flag, shifts the other bits left 1 place and then copies the C flag (the old bit 7) into bit 0. So %11111110 becomes %11111101. The X flag will not be affected.

To affect the X flag, you need to use the ROXL (or ROXR) variation. ROXL.B #1,D0 copies bit 7 of D0 into the C and X flags, then does the rotate and puts the C flag into bit 0 as before. Both the C and X flag take the value of the last bit rotated out - see details below for the full picture.

The flags are affected thus:

N - set if the most significant bit of the result is 1 otherwise cleared.

Z - set if the result is zero, cleared otherwise.

V - always cleared.

For **ROR** and **ROL ONLY**:

C - set to the last bit rotated out (0 or 1) but cleared if the rotate count was zero.

X - unchanged.

For **ROXL** and **ROXR ONLY**:

C - set to the last bit rotated out (0 or 1) but if the rotate count was zero, it will be set to the value of the X flag.

X - set to the last bit rotated out (0 or 1) but if the rotate count was zero, it will be UNCHANGED and copied to the C flag.

Putting it together - for the project.

As mentioned in the last issue, we are writing a disassembler as the series progresses. You have put up with a lot of quite boring stuff over the course of the 4 articles so far, so now is the time to 'get on down' to some hard coding work!

The first part of the project follows on here.

QLTdis - Part one - The background.

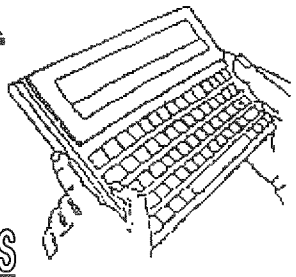
As the series progresses, I would like to leave each of the remaining articles with a piece of code that can be added to the existing code for the project. In some cases this may not be possible, but I will try to leave you with a 'working' program at the end of each issue.

sinclair Z88

OFFICE/FAX 01494-871319 (EEC) *W.N. Richardson & Co.*
MOBILE: 0850 597650
E-MAIL: WNR@COMPUSERVE.COM
6 Ravensmead
Chalfont-St-Peter
Buckinghamshire SL9 0NB

THE IDEAL PORTABLE COMPANION FOR THE QL & PC

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 WILL BUY Z88's & PARTS

* OZ4+512k Internal Ram £150 *

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)	£25	$35
Z88 TO MAC KIT	£25	$35
Z88 TO BBC KIT	£25	$35

ALL WITH INSTRUCTION BOOKS, CABLES AND EPROMS AS REQUIRED!

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

Sometimes, I will have to use code that we may not have covered. In these cases I will briefly explain what is going on, but a full explanation will follow in the series - so don't panic.

The outline of the project is this:

1. Perform 'once only' program initialisation.
2. Perform 'once per loop' initialisation.
3. Perform a disassembly of a single instruction.
4. Print the disassembled instruction.
5. Check if the user has requested that we stop - ESC pressed & goto step 8 if so.
6. Check if we are at the end of the loop - reached the end address, goto 8 if so.
7. Go to step 3 again.
8. On ESC or end, perform any 'once per loop' terminations.
9. Go to step 2, unless user quits.
10. Perform 'once only' termination & exit QLTdis.

Each step above can be expanded as follows:

1. Once only initialisation. This step involves opening the various channels required by the program, setting mode 4 if not already set. Any errors detected here must result in the program exiting to SuperBasic as it is not possible to continue.
2. Once per loop initialisation. This step involves getting the start address, end address and 'printer channel' from the user, printing the program headings and checking if the user has decided to quit or not. Errors detected here should be notified to the user and the information requested again.
3. Disassemble a single instruction. Decodes the word at the current address and updates the current address as required depending upon the size of the instruction (2 to 10 bytes). Builds a buffer holding the instruction address, hex codes and disassembled instruction. Any errors should be handles as appropriate.
4. Print the disassembled instruction. Prints the buffer filled by the step above. This always goes to the screen but may also be required to go to the printer or a disc file as well. This is all handled here.
5. Check if ESC pressed and 6. Check if end address reached. (combined test). Both tests have been combined into a single step as both carry out the same actions. We need to scan the keyboard for the ESC key being pressed or reaching the stop address as specified by the user in step 2.
8. On ESC or end, perform any 'once per loop' terminations. This involves closing any 'printer' files.
9. Go to step 2, unless user quits.
10. Perform 'once only' termination & exit QLTdis. Close all open channels, reclaim any allocated memory and quit the job. QDOS does close channels etc and deallocates memory when a job is killed - but it is best to be tidy. (Some systems (PCs and Windows for example) don't do this.)

Dissassembly - how it works

Step 3 seems quite simple. It is not. There are a number of instruction formats, addressing modes etc as defined in George Gwilt's text file which should be included on the cover disc. If you look at the bit pattern for each instrcution you will notice that they come in many different flavours. I spent weeks (in various hotels) looking through the instructions and collecting similar ones together. My first pass detected 32 different 'families' of instruction.

The next step was to write down which bits in the instruction op code were 'variable'. This simply means which bits vary with the different forms of the instruction. So now we have 32 different instruction types documented and a description of the variable bits noted for each. As an example of what I am on about:

The NOP instruction has the op code of $4E71 or %0100 1110 0111 0001 and has no variable bits. This is one of the simplest instructions to decode and works along the lines of:

```
IF PEEK(PC) = $4E71 THEN INSTRUCTION = 'NOP'
```

(in SuperBasic)

roughly equivalent to:

```
      LEA     <address to disassemble>,A4
      CMPI.W  #$4E71,(A4)
      BNE.S   NOT_NOP
;
; Process NOP instruction here
;
NOT_NOP
; process next instruction here if NOP ;
not detected
```

For reasons discussed later, we don't actually do it quite this way!

These instructions I have called TYPE 0 - they are the simplest to disassemble. Others in this group are ILLEGAL, RESET, RTE, RTR, RTS and TRAPV - some we know and most we don't (yet).

The TRAP #n instructions, where n = 0 to 15, have the op codes of $4E40 through to $4E4F. The lowest 4 bits (or nibble) hold the trap number - 0 to 15. To disassemble this instruction we need to mask out the data part (the trap number) and check the remainder for being the $4E4x format TRAP instruction. This is done using the ANDI.W instruction as follows:

```
LEA     ‹address to disassemble›,A4
MOVE.W (A4),D0
ANDI.W #$FFF0,D0
CMPI.W #$4E40,D0
BNE.S  NOT_TRAP
```

At this point, we know that the instruction at the address held in A4 is a TRAP #something, so now we have to extract the trap number form the instruction:

```
MOVE.W (A4),D0
ANDI.W #$000F,D0
```

And now we have the trap number in D0.W. The instruction TRAP #‹D0› can be built up in the buffer and later printed.

These instructions I have called TYPE 4 and includes only the TRAP #0 to TRAP #15 variations of the TRAP instruction.

I originally caclulated that there were 32 different types of instruction 'family'. Further examinations reduced this number slightly - and I suspect that even more binary examinations would help to reduce it even further. As thay say in all the best books, 'I leave this as an exercise for the reader'!!!

Once the various families have been identified, I calculated what the mask word needs to be for each family. The mask word is a 16 bit word which is ANDId with the op code word and changes all variable bits into a zero. All the fixed unchanging bits are left as they are. Using the trap code above as an example, we have variable data in bits 0 to 3 and fixed in bits 4 to 15 so our mask needs to have a 1 in bits 4 to 15 and a zero in bits 0 to 3 to achieve the desired effect.

Having built a mask for each instruction family, I now need a value to check against when the mask has been applied. This will be different for each instruction within the family. Using the type 0 family, the mask will be $FFFF as there are no variable bits. Applying this mask to NOP should give the result $4E71 while RTS will give the result $4E75.

This means that we need a table consisting of a mask word, a result word and an instruction type for each and every instruction in the 68000 instruction setThe table format is mask,value & type:

```
t_nop  DC.W $FFFF,$4E71,0
t_rts  DC.W $FFFF,$4E75,0
t_trap DC.W $FFF0,$4E40,4
t_oops DC.W $0000,$0000,33
```

DC.W is an assembler directive which tells the assembler to reserve one word of memory and place the value given into it. The labels are just there to remind you which instruction we are setting the values for.

The final entry, t_oops, is a catch all which means that any 'broken' instructions which have not been detected by the rest of the table will be caught here - ANDI.W with 0 always = 0.

The only thing missing here is a string holding the actual instruction. As the various instructions are of differing lengths, we have a slight problem. We must keep each entry in the table the same size so what we do is hold the distance from the start of the table to the string that makes up the instruction. Our new table looks like this:

```
      DC.W $FFFF,$4E71,0,t_nop-op_table
      DC.W $FFFF,$4E75,0,t_rts-op_table
      DC.W $FFF0,$4E40,4,t_trap-op_table
      DC.W $0000,$0000,33,t_oops-op_table
op_table ; Start of the table
t_nop  dc.b 3,'NOP'
t_rts  dc.b 3,'RTS'
t_trap dc.b 6,'TRAP #'
t_oops dc.b 6 'OOPS !'
```

This table is in two parts. The first has 4 words for each entry, the second has a number of instruction strings relating to the first part of the table. In the first part, the assembler directive DC.W t_nop-op_table does this:

- calculates the address that t_nop will have in the final code file
- calculates the address that op_table will have in the final code file
- subtracts the t_nop address from op_table's address to get an 'offset'
- stores the offset as a signed word in the first part of the table.

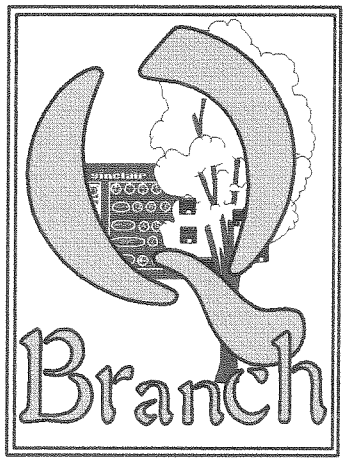
This means that the table looks like the following:

```
      DC.W $FFFF,$4E71,0,0
      DC.W $FFFF,$4E75,0,4
      DC.W $FFF0,$4E40,4,8
      DC.W $0000,$0000,33,15
```

What has happened is that the assembler has calculated the distance from the start of op_table to each individual row and placed the distance into our first table. As we will know the location of op_table in memory when the program is executing, we can add the distance or offset on to the table start address and we will get the correct place in the table to read the instruction string from.

Now that we have a table of masks, values, instruction family type and a location for the instruction string, we can use code similar to the following disassemble an instruction:

Q Branch



Q Branch News

T.F. Services and Q Branch are now making the Q40 boards and, by the time you read this we hope to have them ready for shipping. This is the most exciting development in the QDOS / SMSQ world for a long time and we fully expect to sell many of these boards. Tony Tebby recently sent us the first version of SMSQ/E for the Q 40 and work continues towards the full release version contact us now for details.

We do have two new Super Gold Cards and one QXL board left for sale. We are still looking for the missing chips to make up the last five Super Gold Card boards but they are getting rarer and rarer. If you need memory expansion and disk interface get on the phone today !

SMSQ/E

Gold Card / Atari / QXL Version

£ 76.00

Various Atari versions : call for details

QPC

£ 90.00 (£ 70.00 SMSQ/E Owners)

With CUESHELL Module :

£ 110.00 (£ 90.00 SMSQ/E Owners)

We can now supply colour laptops with QPC already installed ! The systems are secondhand and vary considerably so you will need to call for details but a typical starting price for a 486 DX 70 Toshiba with 350 Meg hard disk and 12 Ram complete with QPC and Carrying case is

£ 500.00 (+postage)

'Just Words' by Geoff Wicks

THESAURUS, STYLE CHECK, SOLVITPLUS 3

£ 15.00 ea / ANY 2 PROGRAMS £ 25.00 / ALL 3 PROGRAMS £ 35.00

THESAURUS V4.01 / SOLVIT PLUS 3 v 3.00

Pointer drivens versions of Geoff Wicks popular programs.
(needs Hot_rext, WMAN and PTR_GEN or SMSQ/E to run)

Still the same price £ 15.00

Upgrades from previous versions £ 2.50 + S.A.E.

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

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. WATCH THIS SPACE !

UTILITIES

FiFi 2	£ 22.00
QSup	£ 32.00
QSpread v1.44	£ 66.00
Cueshell 2	£ 30.00
Qload / Qref	£ 15.00
Disk Mate 5	£ 16.50
QPAC 1	£ 20.00
QPAC 2	£ 40.00
QTYP 2	£ 30.00
QLQ	£ 32.00
LDUMP	£ 26.00
Q Count	£ 25.00

New Version

PROGRAMMING

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

Shop hours :
Mon - Fri
10am - 6pm
Wed / Sat
10am - 2pm

The Bank Volt, 6 Coronation Buildings, Ham Road.
Worthing, W. Sussex. BN11 2NN U.K.
Tel: +44 (0) 1903 200212 Fax: +44 (0) 1903 208070
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.

Q Branch

Hardware

QXL II	£ 180.00
Super Gold Card	£ 160.00
Recycled Gold Card	£ 60.00 *
Aurora	£ 100.00
Qubide	£ 55.00
Qplane	£ 25.00
Aurora cables	£ 3.00
Aurora rom adaptor	£ 3.00
The 'Braquet'	£ 16.00

* when available.

On board Sound / Graphics / Keyboard Interfaces
16 Mb Ram, full 68040 processor I/O card with floppy and hard disk support.
The fastest QL Hardware yet !
Special deals for the first 25 pre-orders.
£ 330.00 - Call us for details.

Q Branch Programs

The Knight Safe 1 - standard version	£ 30.00
The Knight Safe 2 - with compression	£ 35.00
Q - Count Pointer driven home accounting	£ 25.00
Q - Route v1.07 Route finding programme	£ 25.00

ProWesS

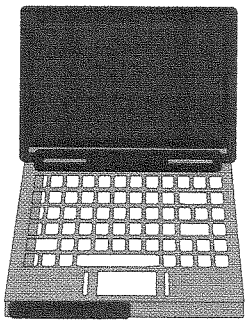
ProWesS	£ 48.00
DATAdesign	£ 24.00
Fontutils	£ 30.00
File Search	£ 12.00
PFlist	£ 12.00
Fontpack	£ 60.00
LINEdesign v 2.16	£ 24.00
PWfile	£ 18.00

Package deals

QXL II + SMSQ/E	£ 220.00
Aurora + SMSQ/E	£ 160.00
Above package + Monitor	£ 380.00

2nd user 14" SVGA Colour Monitors !
(suitable for Aurora)
Ready for use and giving various screen resolutions
prices from £ 20.00 + shipping

LAPTOPS



LAPTOPS

We have these laptop computers in stock :
with QPC - add £ 60.00

	Ram	Hard disk	Processor	CD ROM	O/S	Price
Toshiba 2130CS	8Mb	502Mb	486/DX100		W 95	£ 450.00
HP Omnibook	16Mb	1.55Gb	P150		W 95	£ 700.00
Hitachi SA 100	32Mb	3.2Gb	P133	12 x speed	W95b	£ 800.00

Prices do not include shipping

Laptop 12 x CD ROM PCMCIA card + Drive + software £ 160.00

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.

```
START   LEA     ‹address to disassemble›,A4
        LEA     MASKS,A5            ; Table of masks etc
        MOVE.W  (A4),D7           ; Op code word

AGAIN   MOVE.W  D7,D0             ; Copy of op code word
        AND.W   (A5)+,D0          ; Mask out variable bits
        CMP.W   (A5)+,D0          ; Check against value
        BEQ.S   TYPE              ; Found it, skip rest of loop
        ADDQ.L  #4,A5             ; Skip over type and offset words
        BRA.S   AGAIN             ; Try again

TYPE    MOVE.W  (A5)+,D0          ; Get the type word
        MOVE.W  (A5),D1           ; Get string offset
        LEA     OP_TABLE,A5     ; Start of string table
        EXT.L   D0              ; Convert offset from word to long
explicitly
        LEA     (A5,D0.L),A5    ; Add offset to table start and put in A5

        ; Process according to type here

MASKS   DC.W $FFFF,$4E71,0,t_nop-op_table
        DC.W $FFFF,$4E75,0,t_rts-op_table
        DC.W $FFF0,$4E40,4,t_trap-op_table
        DC.W $0000,$0000,33,t_oops-op_table

OP_TABLE
T_NOP   DC.B 3,'NOP'
T_RTS   DC.B 3,'RTS'
T_TRAP  DC.B 6,'TRAP #'
T_OOPS  DC.B 6 'OOPS !'
```

At this point (just before MASKS above), D0 holds the instruction type word, D7 holds the actual instruction word, A4 holds the address of the instruction and A5 holds the address of the string in memory. We are now ready to process the instruction acording to its type word, as held in D0.W - how best to do this?

In SuperBasic we would probably have a SELect statement to process it. In assembler we can do this using code similar to the following:

```
CMPI.W #0,D0
BRA    TYPE_0
CMPI.W #1,D0
BRA    TYPE_1
;
;
CMPI.W #32,D0
BRA    TYPE_32
BRA    TYPE_33
```

The problem is that this is a long drawn out piece of code and would be prone to typing errors. Another method is to use a jump table as follows:

```
J_TABLE DC.W TYPE_0-J_TABLE
        DC.W TYPE_1-J_TABLE
        DC.W TYPE_2-J_TABLE
        DC.W TYPE_3-J_TABLE
        ;
        ;
        DC.W TYPE_32-J_TABLE
        DC.W TYPE_33-J_TABLE
```

Which sets up a table of offsets from the start of the table to the routines in memory. The code to select the offset and jump to it would be similar to this:

```
        LEA     J_TABLE,A2      ; Table of jump offsets
        EXT.L   D0              ; Make sure D0 is long sized
        LSL.L   #1.D0           ; Multiply by 2 - table entries are 2 bytes
        MOVE.W  (A2,D0.L),D0    ; Get offset
        JSR     (A2,D0.W)       ; Perform routine
```

The table holds the difference between the address of the routine and the address of the start of the table (J_TABLE) in a signed word. D0 has the type number so needs to be doubled to convert it into an offset into this table. Type 0 starts at offset 0, type 1 at offset 2, type 2 at offset 4 and so on.

The value in D0 is then added to the tabls start address and the word at that location extracted from the table and copied into D0 by a single instruction

```
MOVE.W (A2,D0.L),D0
```

This is a small demonstration of how powerful the 68000 addressing modes can be. The instruction following is equally as powerfule. Remember A2 holds the table start address, D0 now holds the offset from A2 to the routine we want to perform, so adding them together should get us where we want to be - at the start of the routine. The single instruction:

```
JSR (A2,D0.W)
```

does just that.

If you have been paying attention, you will notice that in previous bits of code I have commented 'Convert from word to long explicitly' but here I don't. In previous example code I explicitly convert the word value in a data register to a long value using EXTL Dn this SIGN EXTENDS the word value from 16 to 32 bits allowing long sized operations to be carried out.

In the JSR (A2,D0.W) above, the 68000 does the sign extension automatically. I merely demonstrated both methods.

Now then, JSR and BSR what is the difference? BSR is Branch to subroutine and JSR is jump to subroutine. Surely these are identical? No they are not. In assembled code, a BSR is actually a PC relative jump. The data that defines the destination of the branch is a signed offset from the current PC (Program Counter). In a JSR it is an absolute address.

We normally write BSR label and the assembler works out the offset. So the following are not equivalent:

```
BSR $100
JSR $100
```

As the BSR $100 means 'jump to the subroutine $100 bytes on from here' and the JSR $100 means 'jump to the address at address $100' - a small but subtle difference.

We have placed the assembler source code on to this cover disk to save you some typing. The files are

QLTdis_asm

More details of the QLTdis in the next issue.

■

Adventures on the QL Part 1 - The Prawn

Darren Branagh

A few months ago I was sent a collection of disks to review from a (then) new software company, RWAP Software, fronted by Rich Mellor. Unfortunately, due to other commitments (namely work) I am only now getting the chance to review these - so firstly, apologies to Rich for the delay.

The Prawn is a text adventure game, which means you basically enter instructions via the keyboard and these are interpreted by the program accordingly - such as CLIMB TREE, KILL BEAST, EXAMINE ROCK, or whatever. Directions and commonly used instructions are abbreviated to single letters such as N,S,E,W to go North,South,East or West. You get the picture.

You play the part of a prawn that gets caught up in a trawler net, only to find himself on a strange land, with a weird copper band around his tail that must have been put there as he slept. You must help the prawn survive the perils of his dilemma and help him solve problems along the way.

The name comes from the fact that you play the part of, you've guessed it, a prawn. It's a play-on-words spoof of another QL adventure, called The Pawn, which was marketed by Magnetic Scrolls in the early Eighties and while it was a rather serious swords and sorcery type epic, the Prawn is quite the oppostite - funny and tongue in cheek.

First Look

On Loading the game, a rather nice graphical screen is presented of two symmetrical prawns, with the choice of three scenarios to play. These are: KINGS KEEP, ANCIENT CITY OF LAROS and THE BADLANDS. These are selected by pressing A, B or C. I chose A for KINGS KEEP as this is the first part of the adventure and the best place to start.

You are then presented with 3 further options:

- beginning a completely new game by typing BEGIN
- INTRO will give you a brief introduction to the game
- keying LOAD will load a previously saved game

All very simple and user friendly. I type BEGIN and off we go. Immediately, we are given our first description - "The Prawn trudges along a gravelly path, trailing his tail behind him..." and so on. It also tells us the path leads east to west and a forest is to the south. Now we are adventure gaming - all the text is quite colourful, with location headings in different colours to the main description.

So how good is it?

I found the user interface of the Prawn to be excellent even when you type something that the interface doesn't understand, rather than get something like "command not recognised" you get a rather funny passage along the lines that a Prawn has a brain the size of a flea, and to tell him something he'll understand. This is one of the reasons I enjoyed the Prawn so much - its humour. For instance, as you approach the kings keep and enter a stable courtyard, you are told it's empty as the cavalry are holidaying in Majorca with the horses!!

The Prawn has some really useful commands I liked, which I have never seen in any other adventures. Among these are RAMSAVE and RAMLOAD, which allows you to save a current game to RAM rather than disk so that you can return to it later, quickly and easily - a nice feature and useful if the phone rings or whatever. Obviously, if you needed to switch off the QL you would save to a floppy or hard drive, which is also available as an option.

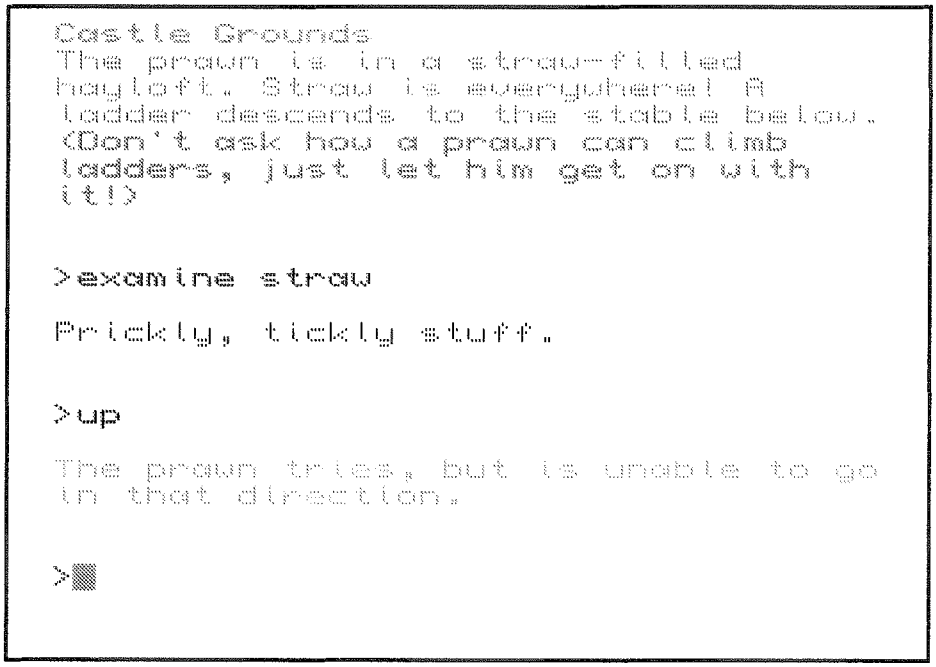
The gameplay is good too - each location description appears almost instantly when you enter a new command. The game was originally written on the Quill adventure system (a utility for adventure writing way back then) but Rich has since converted it to Superbasic and QLiberated it. It runs remarkably quickly, both on a gold card QL and my QXL II.

On top of that, despite being funny, it's a very challenging game. It is difficult to get to where you want first time as something always pops up to thwart your efforts. You then have to think on your feet (do prawns have feet?) as to how best to solve the problem. Also, despite taxing the brain, I found the Prawn an extremely good way to relax and unwind especially after a hard day at the office.

Compatibility

This is the only area of complaint from me - The Prawn refused to load on my QPC powered Pentium PC. I was trying it in 800x600 resolution, so this is probably why. In fairness to Rich, in the manual which accompanies the game, he says that as there are now so many platforms on which to run QL Software, full compatibility cannot be guaranteed, but he will endevour to cure any problems which occur if given an accurate description of the hardware and circumstances involved when the problem arose.

I tried the Prawn on my QXL II running in a Philips 286 PC. In 512x256 mode it ran perfectly



QUANTA

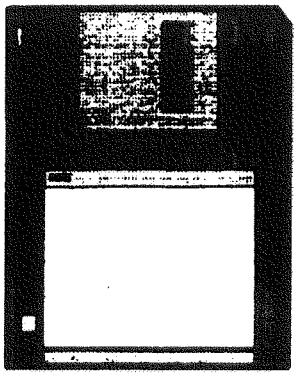
Independent QL Users Group

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

Barclaycard: Visa: Access: Mastercard

*** Now in our SIXTEENTH successful year ***

Further details from the Membership Secretary



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

and quickly too - much faster if run from hard drive than from floppy, obviously. It also ran perfectly on my original black box QL with Gold Card or Trump Card (I tried both). It ran fine with the Pointer Environment loaded too.
What about graphics? Well, I have always had a soft spot for text adventures, mainly because they leave so much to the imagination - which contains the most powerful graphics processor of all! I have loved them since I first loaded SHIP OF DOOM into my rubber keyed Spectrum in early 1984. I hope through this review some of you will try this wonderfully funny and playable game and get some fun from your QL. Now, back to the king's keep. Where was I?
P.S. Part 2 of this QL Adventure series of reviews featuring another RWAP game, called NEMESIS, follows next issue.
■

COMPETITION

Based on an idea suggested to me by Tony Firshman, the idea behind this competition is that you have to make up a vaguely meaningful sentence out of all the original QDOS SuperBASIC keywords (in CAPS in the keyword guide of the QL manual) plus all of the valid listed operators (& . : , etc)
The winner will be the one judged to use most of the keywords, but marks will also be gained by making the QL Today editorial team laugh a little, and of course it has to be legal and decent enough to be published! A little repetition is allowed, but marks will be deducted if too many keywords are repeated.
Example: **IF AT AUTO THEN BEEP AND GOTO BORDER. BLOCK OR CONTINUE BEEPING & PAUSE FOR DAY$**
The prize will be a small software voucher from JMS, so get composing!

Printing EURO with QLs

Dietrich Buder

1. Introduction

Since 1.1.1999 the share prices are shown only in EUROs and now some shops print their price-tags and advertisings in DM and € too. I expect that the price lists of our QL traders will come only with €, Pounds and SFr. And soon we have to make our tax returns in EURO. Then I or we need to write and print out with our QLs and need the EURO symbol on our printers. For details of the construction of €, see the article in QL Today english May/June 1998 page 14.
All my knowledge is only for QL CLASSICs, QXL and QPC, the word processors QD, QUILL/XCHANGE, text87 and some dot-matrix printers.

2. € on the QL screen

The first part was to search for an unnecessary special character. Tony Tebby and Jochen Merz took the big question mark CHR$(181), a character unused on normal printers. The present SMSQ/E version 2.91 has the € built-in and it looks fine. But the solution is not perfect: The € is obtained by typing the keys [CTRL][SHIFT][u] and the new German keyboards for Windows98 have the EURO symbol on the key [e]. The less-than-ideal solution is to define the hotkey

```
ERT HOT_KEY ('e',CHR$(181))
```

for [ALT][e] and loss of the previous hotkey for 'e'.

The keys [CTRL][SHIFT][e] supply the 'O with tilde' CHR$(165), also not a special character of normal EPSON printers. I prefer this solution because it is logical. The ASCII-181 is a border of EPSON printers and I want to use it.
In my following description I write about '€ on key [u]'. Later come some hints for my solution with '€ on key [e]'.
QD uses a font for all characters. The disk of QD98 has a new QL_FNT with EURO on CHR$(181). Therefore QD works fine with EURO on QDOS and SMSQ/E.
BASIC and QUILL/XCHANGE got the EURO from SMSQ/E V2.91. QDOS users with JM/MG/MGG-ROM have to accept the wrong question mark and Minerva users get Q and V one above the other. They can ask Tony Firshman for a good solution.

For Text87 users, the file 'FOUNTED89_EXE' can change all screen characters. But I know of no solution to print out the € with text87.

3. Print out the €

All my word processors transmit the ASCII code of the typed character to the printer and expect a correct print out. Each QLer knows the difference between the CHR$(x) of QLs and the ASCII code of printers. We need always a TRA table to translate the different codes for characters higher then CHR$(127).

I have three solutions for printing out the €.

3.1. The botched solution: Many printers know the command CHR$(8), the backwards step for one character. A printer test is very easy. Enter this (substitute SER1 or SER2 for PAR as required):

```
OPEN #3;'par'
PRINT #3;'('&CHR$(8)&'='
CLOSE #3
```

The printer is useable if the printed character looks similar to the €. The TRA 3 of SMSQ/E V2.91 made such an EURO as a three-step character with 'C'&CHR$(8)&'-' and it looks terrible for font types with serifs.

QDOS users have to work with a TRA table for three-step characters, SMSQ/E users only for simple printers. Such a TRA table creates my new file 'TRA_EURO_BAS' which was first published by the German newspaper Computer Kontakt issue 12/1987. My new file can create four different files 'TRA_EUxx_BIN' for € at key [e] or [u] and for point 3.1. and 3.2.

3.2. The ingenious solution: Some printers have the option 'Download' to print out user defined characters. Such a EURO looks like being a feature of future printers, see point 3.3.

The first work is to study the printer manual and see if the answer to all questions is'yes'.

- Is the printer option 'Download' built-in?
- Is download mode configured?
- Does the printer accept the normal EPSON download commands?
- Do you use SMSQ/E for the necessary command 'UPUT'?
- Is the file 'TRA_EUdu_BIN' on device 'dev1_'?

The download solution requires a special TRA table with a translation from CHR$(181) to ASCII-181. It can also be created with my 'TRA_EURO_BAS'. The TRA 3 of SMSQ/E don't work correctly for download EURO on my QLs.

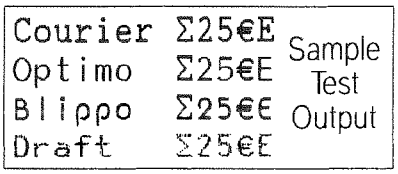
3.3. The trivial solution: Waiting for future printers with € at ASCII 164. This requires another TRA 3 or other TRA table with translations from CHR$(181) to ASCII-164.

4. Commands for download characters

The first step is normally to change the printer mode for download characters. It will be easy if the manual is written clearly and understandably! I got problems and Jochen Merz helped me.

The second step is to check the printer commands of my file 'DRU_EURO_DEMO_BAS'. It is written for my 24-pin dot matrix printer, a STAR-XB24, and EPSON commands of the operating mode 'Standard'. My file has REMarks for changing the language at line 1380, the operating mode to 'IBM' at lines 1570/1840 or for 9-dot-matrix printers. The available printer fonts often require an easy change of font codes at line 1220.

This file should print three NLQ lines with a German special character and the '€' and one DRAFT line also with the special character and '€', please see my print out (made with QUILL in double size mode). If this test is successful you have a source for all necessary files to work with QD and QUILL/XCHANGE.



The command lines 1510 to 1600 and 1790 adjust the printer and the lines 1630 to 1790 fix the printer dots of the NLQ download EURO. This graphic mode works only with TRA 0 because a TRA table would change some dots. But the print out lines 1800 and 1970 needs a TRA table for the German special character. This contradiction is resolved by the SMSQ/E command 'UPUT'. It is similar to the command 'BPUT' but it ignores the TRA table. Lines 1570/1580 and 1840/1850 copy the selected NLQ printer font from the printer ROM to the printer RAM and after lines 1790/1960 the printer uses the font in RAM.

During this download mode no printer RESET instruction (27, 64) is allowed! This may be one reason for the problems with text87.

Now I'll give some hints for all QLers who have never used printer graphic modes. Normally the printer manuals give detailed information. The following rules are for my STAR-XB24:

- maximum columns for the character: 9 for draft and 29 for normal NLQ.
- maximum columns for character and border left and right: 12 and 36
- dots of 24-dot-printers may lie on top of each other but not side by side

The last rule is the reason for the horizontal zigzag lines of draft characters like 'E'.
The 24 needles of the printer work in three groups from 1 to 8, 9 to 16 and 17 to 24. Each needle has a value and all values of one group must be added. This gives three numbers for three bytes. A draft character has a maximum of nine columns and therefore we get 9*3=27 bytes for one draft character. A non proportional NLQ character with 29 columns has 29*3=87 bytes. These bytes overwrite the original bytes in the printer RAM.
My EURO in draft is constructed:

Column	1	2	3	4	5	6	7	8	9
1. Byte	0	3	12	16	32	16	32	16	12
2. Byte	72	182	73	144	72	144	72	128	1
3. Byte	0	0	128	64	32	64	32	64	128

Nr	Value									
1	128									
2	64									
3	32					XX		XX		
4	16				XX		XX		XX	
5	8			XX						XX
6	4			XX						XX
7	2		XX							
8	1		XX							
9	128		XX		XX		XX		XX	
10	64	XX		XX		XX		XX		
11	32		XX							
12	16		XX		XX		XX			
13	8	XX		XX		XX		XX		
14	4		XX							
15	2		XX							
16	1			XX						XX
17	128			XX						XX
18	64				XX		XX		XX	
19	32					XX		XX		
20	16									
21	8									
22	4									
23	2									
24	1									

5. Hints for the BOOT

The BOOT file of our QL must have some command lines for the TRA tables added to it. I prefer to store important data in RAM3. This may cause trouble but as far as I can see, I get no problems. The minimal solution only for SMSQ/E is:

```
i$='u': x=181: REMark for EURO with
[CTRL][SHIFT][u]
b=RESPR(512): LBYTES
'dev1_TRA_EUb'&i$&'_BIN',b: REMark  bodged
EURO
d=RESPR(512): LBYTES
'dev1_TRA_EUd'&i$&'_BIN',d: REMark download
EURO
OPEN_NEW #3;'ram3_TRA_DAT': PRINT #3; i$\b\d:
CLOSE #3
ERT HOT_KEY ('e',chr$(x)): REMark for EURO
with [ALT][e]
ERT HOT_CMD ('q','ex dev1_QD_EURO_OBJ: cls
#0'): REMark starting QD
```

I think it is unfortunate that some important characters are not in the QL font. QLers with hardware ambitions miss the Ohm character for resistors eg. on the keys [CTRL][SHIFT][o]. Also, some QL characters are not in the normal font of EPSON dot-matrix printers and presumably not necessary for the important European languages. These are: CHR$ 129,133,139,161,165 (my €), 171,175(important),181(€),183. So I changed the QL_FNT of the QD disk with the file 'QLUDGE_V2.5' of the speedscreen disk and made my own fonts with Ohm, some mathematical characters, other special characters and borders of the EPSON font. I want to be on the safe side and have two files with EURO=CHR$(181) and my favourite with EURO=CHR$(165). The file names are 'QL_FNTu' with only 38 EPSON borders and QL_FNTe' with all 40 EPSON borders. The 'TRA_EURO_BAS' is made for both QL fonts. If anyone is interested in getting the new QL_FNTx to save having to do your own work: ask me or Jochen Merz.
So my BOOT file has the additional lines of the SMSQ/E manual:

```
f$='QL_FNT': f=RESPR(5*512):
LBYTES 'dev1_'f$&i$,f: CHAR_DEF f,0
FOR k=0 TO 2: CHAR_USE #k;0,0
COPY 'dev1_'&f$&i$, 'ram3_'&f$        and now
OPEN_NEW #3;'ram3_TRA_DAT':
PRINT #3; i$\b\d\f: CLOSE #3
```

6. Hints for word processors

The file 'ED_QD_EURO_BAS/OBJ' results from the change of 'DRU_EURO_DEMO_BAS' and is written for QDOS and SMSQ/E and also for German or English language, see line 1080. The first necessary input is for the printer connection and only in this place comes the printer RESET line 1390.
Under SMSQ/E follow the question to the printer mode 'download', lines 1400 to 1430. If the input is [y] for 'download configured' at line 1300 then the download commands copy the bytes of draft EURO into the printer RAM. I use only condensed characters in draft because QD has 80 characters per line. The print out looks like NLQ.
The correct TRA table line 1300 results from the inputs and the file 'TRA_DAT' line 1250. At the end comes the start of QD line 1320.
The configuration of QD must be without a RESET 27,64 and should be for the QL_FNT at RAM3. So it is possible to check other QL fonts without changing any harddisk file.

The start of QUILL or XCHANGE is similar to the QD start file. But this contains more software commands eg. for printer type and port connection and then for each printer the font type, font size, italic, line width, perhaps download EURO in NLQ and draft and suchlike. If anyone is interested he may ask me or Jochen Merz.

I wish all readers good results with download characters.

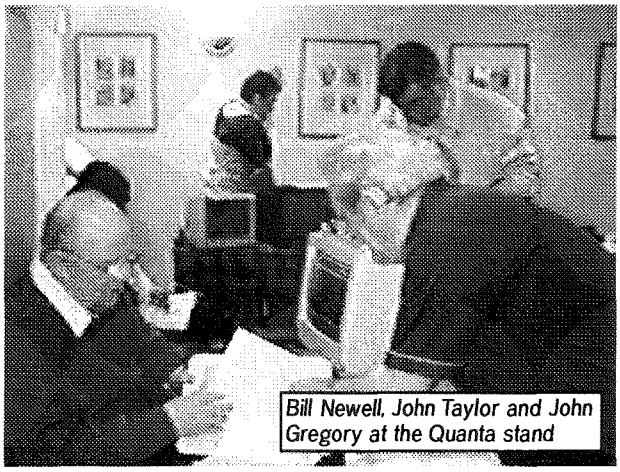
[Editor's note: the files referred to in this article are on the cover disk with this issue]

■

Hove Quanta Workshop

Dilwyn Jones / Pictures taken by Paul Hasler

On Sunday 28th February, Quanta held a one day workshop in the Excelsior Hotel, Hove, on the south coast of England. The night before, several of us gathered at a local Indian restaurant (once we'd found it!) for a get together and a chat and to exchange gossip. A difficult night's sleep followed for many of us, as the rooms in our hotel were boiling hot and the constant noise of water in pipes was interrupted only by vain efforts to turn off radiators due to the fact they either had no knobs, or they were broken.



Bill Newell, John Taylor and John Gregory at the Quanta stand

This workshop was quite well attended and the highlight of the show was the appearance of the Q40 boards from TF Services and Q-Branch. Designed by Peter and Claus Graf in Germany, this is the latest QL-compatible hardware and looks set from the outset to have 3 operating systems to choose from.

SMSQ/E will cost about 30 pounds for it, while the Q40 comes with a free version of the QDOS Classic operating system, itself adapted from the old Amiga QDOS emulator very like a JS ROM in some ways. Meanwhile, a version of Linux has also been ported to the Q40 by Richard Zidlicky. Rumour has it you can run the uQLx emulator on this. If true, you have a choice of three ways to run QL software on this brand new machine.

It's to be supplied as a board designed to fit into PC-AT style tower cases, though at some point systems may be offered ready built or custom built. The units on show at Hove were not yet fully working as the operating system versions had been arriving fast and furious and still had a few teething problems. Prototype high colour mode drivers apparently existed but were unlikely to be in the initial version.

Most of the well known QL traders were dotted around the room, including Geoff Wicks (Just Words), Qubbesoft P/D, Jochen Merz Software, Q-Celt Software, Bill Richardson, Quanta, TF Services, Miracle Systems, Enrico Tedeschi of Hove Books (see the review of his Sinclair Archive book in QL Today Vol 3 Issue 1), RWAP Software and Q-Branch. Apologies to anyone else I have forgotten.

Enrico Tedeschi had an interesting stand. He collects anything Sinclair related and publishes a small number of books on a variety of subjects. Objects on display included the Sinclair wristwatch of old, wooden models of ZX81s and other computers, MK14s and so on. A real nostalgia trip - the first time I have seen Enrico attend a QL workshop in this way.

Jochen Merz demonstrated his latest QD98 on his stand. Of particular interest to me was the printer filter facility available in this - you can write your own printer filter or driver even in SBASIC via the 'standard' codes listed in the manual.

Having come all the way from Ireland, Darren Branagh had his second outing as a QL trader and demonstrated the new database program, Easy-Base, on his stand. Darren had several disasters on the way home - such as missing his ferry, subsequent sailings cancelled by bad

David Gilham in conversation with Jochen Merz

weather, a burst suitcase, car clamped in Dun Laoghaire as he was late back due to the cancelled sailings... Not the luckiest of Irishmen. I promised him I wouldn't say anything about the sheep he took back with him.

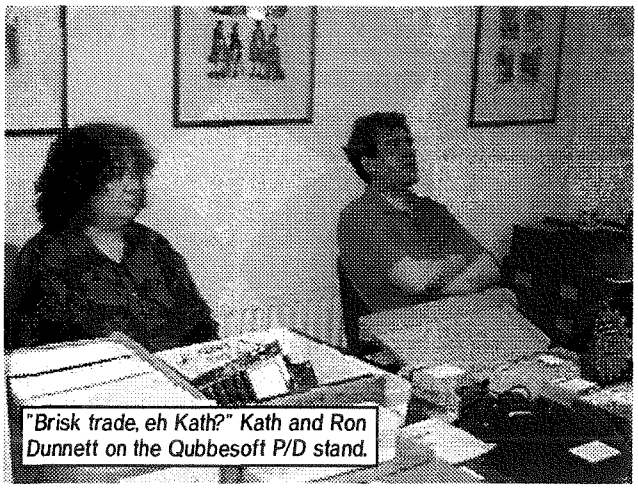
Geoff Wicks keeps on updating his range of software. His latest was the QTYP dictionary version of the Spelling Crib program (see this issue's cover disk for a free copy!) and he has plans for further improvements to his range of software.

On Qubbesoft P/D I saw a Beta test version of a new utility for converting GIF files - might be very useful if the QL is about to acquire a more graphical following with the proliferation of high resolution displays and imminent higher colour modes for Aurora, Q40 and some emulators.

Speaking of Auroras, a few could be seen in their MinisQL guises dotted around the room. Keith Mitchell, who has done much of the work on MinisQL was in attendance on the Q-Branch stand demonstrating and performing surgery on them.

RWAP Software had the latest revision of their SBASIC/SuperBASIC Reference Guide on display, along with latest maps for use with Q-Route, and an imminent release of a utility called Q-Help for adding help files to programs.

Several well known software authors were in attendance, including Jonathan Hudson who was working on a program to allow file transfer between a Palm Pilot hand held PDA and a QL. Mark Knight and David Gilham, the team producing the freeware release of Turbo Toolkit (see cover disk) and working on Perfection were also in attendance. Mark Knight gave a lecture on graphics and fractals (he is working on a major piece of software for people interested in this field).



"Brisk trade, eh Kath?" Kath and Ron Dunnett on the Qubbesoft P/D stand.

Phil Jordan of The Library, a new PD software service, was in attendance, though not actually trading there as his library was not fully ready in time for the show - he does expect to be in action soon and to start making catalogue disks available. He inherited a vast library of software from Steve Johnson and it's proving no mean task updating that quantity of software!

■

Printer Control Codes - A dreaded Subject? - Part 1

Dilwyn Jones

This article is targeted at the less experienced QL users, but it does assume you know a little bit about BASIC on the QL, such as how to use the PRINT command. We get quite a few printer help requests, so this seemed a fairly obvious article to write. Printers often cause computer users more headache than almost any other computer-related subjects.

Many inexperienced users go through that phase of wanting to throw their printers out of the window when they just can't get it to work properly with their software. It's often the unfortunate case that you have to get to grips with printers and printer drivers before you've had a chance to learn much about them, a kind of catch 22 situation.

Many software traders and authors will tell you that printers are often the biggest time wasting issue for software users, that much of the telephone help requests received centre on printers and trying to get them to work with particular software.

Now that I have probably managed to put anyone off ever buying a printer for their QL, it's time I put the record straight. Printer control codes and programming for printers is a difficult subject to master, but once mastered repays the effort handsomely. There comes a point when it all clicks in your mind, that suddenly you feel in control of the subject.

I hope that after reading this article you'll be on the way to understanding printers and printer control codes. The best way to master the subject will be to read the article, then try some simple basic routines to activate various facilities on your printer. I guarantee you'll never remember all the control codes for your printer, the command sets are just too

long and complicated, so keep your manual to hand, you'll need it to look up the details for your printer.

What is a Printer Driver?

If you have acquired any commercial or free programs, you may have noticed that the instructions refer to 'printer drivers'. On the QL, most programs have their own specific drivers; on some other computers the operating system maintains 'standard' drivers for the various printers its users are likely to encounter.

In simple words, a printer driver is basically a list of instructions which control many features of a given type of printer. For example, it may contain instructions for making the printer switch to bold text, or to turn on underlining. Of course, if it can turn something on, it will usually need the corresponding instruction to turn it off again!

It's important to note that in many cases, the driver itself is not a program which writes the information on the printer. The printer driver is often just a table or list of numbers. On the QL, in most cases the actual program (e.g. Quill) does the hard work of sending the information to the printer, first looking at the information in the printer driver file to 'learn' how to control those features of the printer.

Consider the case of the programs supplied with the QL, Quill, Abacus, and Archive. They are all basically text handling programs (Easel needs to handle graphics, so is slightly different) and share a common printer driver, a file called PRINTER_DAT. This contains a table of codes to use for such facilities as bold text, underlining, subscript (small) text and so on. In addition, it contains information such as which printer port to use (is your printer connected to SER1 or SER2 for example), the Translates (how to make the printer handle certain difficult characters such as the Pound symbol), the baud rate and so on. For an explanation of baud rate, see BAUD in your QL manual.

This means that although they are three separate and distinct programs, you only actually need one printer driver between them, which simplifies life somewhat. Once you have created a driver file which works for Quill, it also works for Abacus and Archive. In theory, anyhow!

The next question is, why does one printer driver not suffice for all programs - surely all programs could use the same driver?

A good idea in theory, but one which does not stand up in practice. If it was that simple, it would have been done a long time ago. In practice, printers change and develop, more facilites are added, until the list of facilites becomes quite large, so any printer driver table which has the entire facilites range within it would probably be prohibitively large and cumbersome. In practice, software authors create printer drivers which just hold the list of facilities required by their programs. This enables them to keep the drivers short and simple (well, by printer standards!), and also makes it easier for the user to adapt it to his/her printer, because only relevant information need be changed for a particular type of printer. For example, if the program merely prints text to a printer, there is no point

carrying a great deal of unrequired graphical control code information!

What are Control Codes?

A control code is simply a number, or set of numbers, which control specific functions on a printer. For example, it may be a couple of special numbers which make the printer switch to printing bold text, or print using a different font.

These numbers will have values from 0 to 255, whole values only. This range of values is often referred to as "byte values", so called because such numbers will fit into one byte of computer memory. In some cases, just one of these numbers by itself is enough to control a printer function. For example, on some printers, sending the number 14 to the printer will make it produce double width characters, while sending the value 15 to the printer will make it print condensed (narrow) characters. In other cases, you have to send a sequence of 2 or more numbers to the printer. For example, sending the two values 27 and 52 (in that order) will cause many printers to switch to printing in italics text. On some printers, sending the value 7 will cause the printer to beep at you!

You may have noticed that all the codes I have listed all start with values of less than 32, and there is a good reason for that. If you look in the QL manual under Concepts, Character Set, and study the list of characters given there, you'll find that one column lists the characters themselves, and another lists their "code" values. You'll find that the characters you can print start with the SPACE character, which has a value of 32. As the characters after this are used for printing, we are left with the values 0-31 for these codes used to control printer functions (some are used for screen control as well). Not all have specific functions, while some are a little obscure. But you may have realised that there is a potential problem here - with only numbers 0-31 available, we could only control up to 32 functions on a printer. Modern printers have dozens if not hundreds of functions, so we need to be able to double up somehow. What printer manufacturers have standardised on is to have a second type of control code, allowing for 2 or more values to be sent as a code. In this case, sending the value 27 first tells the printer that more information is to follow. Since both the printer and the software understand the same control codes if the correct printer driver has been used, the printer knows to treat the next set of values as a control code for the length of the command used. Some commands consist of two numbers, others consist of three numbers, while some can be longer. The value 27 is referred to as an ESC or ESCAPE character. There is another slightly less common scheme consisting of control codes which start with the value 28, which is referred to as FS. Following an ESC or FS value, we can even send printable characters, which the printer accepts as part of the control code, and does not print.

The number which corresponds to the character given is called its CODE. So the code of the SPACE character is 32, while an upper case A is 65, the number zero is 48 and so on. On a British QL, using a Sinclair ROM, printable characters have codes starting from 32, up to 191. Additionally, a 'default' character with a code of 31 prints a chequerboard character when the QL gets into difficulties and doesn't know how to print a character you've told it to print. There is a function in BASIC called CODE, whose function is to translate a character into its code value.

So to find the CODE value of the asterisk character (*) we could use a PRINT command like this:

```
PRINT CODE("*")
```

which would print the number 42 to the screen. And as you might expect, there is also a function to perform the opposite action, converting a code value back to the character it represents. This is called CHR$, and we could check our result with this command:

```
PRINT CHR$(42)
```

which would print an asterisk on the screen (or whatever character corresponds to code 42 on your operating system version).

Note that I have used upper case characters to represent the commands to be typed in. The QL will also understand lower case.

The CHR$ and CODE functions will come in very useful when we start writing a few lines of BASIC to send commands to the printer, so it is worth spending the time mastering them now.

Is there such a Thing as a 'Standard' Control Code Set?

It would be rather nice if there was! But the answer, sadly, is no. In very broad terms, the QL user is likely to come across 4

main types of printer command sets. Unfortunately, as printers have grown more sophisticated, the list of printer control functions, or the Command Set, has grown longer and longer, leading to a number of standards within a standard. In general, later standards include all or nearly all of the command set of an earlier version, so if you buy a new printer from the same manufacturer, you will usually find that it continues to work with the old printer driver, but some of the new features (e.g. a wider choice of fonts) may not be supported. Newer standards are thus usually 'supersets' of the older one.

1. EPSON - Used by printer manufacturer Epson for its range of printers, and widely used by other printer manufacturers who claim that their printers are 'Epson compatible', or include an 'Epson compatible' mode, often referred to as an Epson Emulation Mode. There are various degrees of Epson compatibility. My first Epson printer was an FX80, which included a fairly basic command set. Later printers such as the LQ range included a broader command set taking account of the new facilities, while even later printers used a standard known as ESC-P2, which is quite a comprehensive command set.

2. IBM - Used by IBM on its printers. The command set bears some resemblance to the Epson set, but is not identical.

3. CANON - Printer manufacturer Canon has its own command set. I have only used the one on their BJ range of Bubblejet printers. It bears some resemblance to the Epson command set, but with some important differences. Many Canon printers also have emulation modes for Epson command sets, for example.

4. HEWLETT PACKARD PCL (Printer Command Language). This command set is quite different to those used by the other "standards" listed above, and there are various grades of PCL - their inkjet printers may not have all the facilities of their laser printers, for example. If you have been used to an Epson or Canon command set, you may find the HP commands rather difficult to master at first.

Some other printer companies have their own command languages. There is no particular reason for omitting these other than to keep the article simple, and the fact that I have some experience of the four listed!
If you have used certain other computers, you may have realised that the operating system itself actually drives the printer - the fonts printed may not be ones that appear in the printer command set necessarily. Printers can be driven in 'graphics mode', which basically means that you can specify each and every tiny little dot printed by sending them individually to the printer. As you might expect, since printers these days commonly print at up to 300 dots per inch, you end up having to send huge amounts of information to the printer just to print a single character, whereas using our simpler system means that by sending just one value to the printer to print a single character, it makes writing our own programs to control printers a much simpler affair.
■

Last Minute News

News from Jonathan Hudson

Pilot-xfer. A comprehensive set of tools to connect the Palm Pilot PDA (all models) to SMS/QDOS systems. The archive, about 1.2MB along, which may be downloaded from
www.jrhudson.demon.co.uk
contains executables, documentation and GPL'd source code. You can backup and restore your Palm, install new programs and data, transfer data (memos, expenses, addresses etc). We hope to have an article from Jonathan in the next issue of QL Today.

Jochen Merz Software News

QPC 2 will most likely be ready when the next QL Today issue will be delivered. It is running quite stable on Windows 95/98 and even Windows NT. The serial ports are working, but not as fast yet as Marcel would like them to work. The parallel port needs to be done.
This version will - many people asked for it - stretch screens to fit resolution. It is possible, for example, to make the QL's original 512x256 pixels use a full screen in 1024x768 resolution on a PC - this way the characters are much larger, also on modern laptops.
We still need to do a lot of testing, so if you would like to try a beta version of QPC 2 on your PC, check Marcel's homepage
www.deuschle.de/QPC/
or visit the JMS homepage
www.j-m-s.com/smsq/
We hope to have QPC 2 available at the US shows.

BYTS OF WOOD

SAW POINTS OFFCUTS AND SNIPPETS

Hold the Blank Page!

Graham Bindon's letter in the last issue required an instant response if only to dispel the concept that I am a closet PC user. Although I have a laptop and a neat little light which plugs into the keyboard slot so I can see the keys in the dark I do not have a closet - well there is an under stairs cupboard but that is full of paint pots power tools (covered in dust) and a vacuum cleaner (this is a contradiction in terms because if the vacuum is dirty it must contain something and is, therefore, not a vacuum)

All of my catalogues and invoices are produced using LINEdesign and QD and the SuperBasic Reference Manual was written and printed via Text 87. It is all printed on an EPSON STYLUS COLOUR 200 which is now, unfortunately, no longer made. As Jochen says the EPSON STYLUS 740 still has the ESC/P2 drivers that are needed for many of the QL programs and these printers produce perfect colour prints from the ProWesS suite if used with the colour replacement drivers. (see my first Byts of Wood Column !)

I recently set up a copy of Text 87 with the ESC/P2 Drivers on a customer's laptop (driven by QPC). He had just bought a EPSON STYLUS 740 and I was very impressed both with the speed of the printing and the quality of the end result. These printers are not necessarily easy to find but they are well worth the effort in searching one out. What we need now is for someone to write a colour version of the ESC/P2 Driver for Text 87. Volunteers anyone? This is something which would sell because there are many people who would like to print from Text 87 in colour.

One Dot Short of a Matrix

Talking to the EPSON helpline about anything other than PC is like talking to a brick wall but even when you are discussing a PC problem they are one brick short. The problem is that most helplines are staffed by people who are willing to sit down and answer the telephone all day. The prospects for this as a career are pretty low so the people that you talk to are..... well.

When I bought my first EPSON I had only my QL so I threw the printer driver into a box and forgot about it. Later I bought a new Stylus Colour 200 and by that time I had a 486 laptop with a QXL in it and my wife had something similar. I set up a cable from her workroom to the Q Branch HQ in the loft and bought a printer switch. I actually bought two printer switches an automatic one to connect between the two laptops and a two way one to switch between the laptops and the QL because the QL does not send out the right signals to activate an automatic switch. We installed the printer drivers supplied onto our laptops (Windoze 3.11). I printed from LINEdesign - perfect result, Text 87 - perfect result QD - yes you guessed it - perfect. This joy continued through all of my QL packages no problems at all.

On to the laptop. This used to fire up directly to the QXL so I tried that first. Ah yes, just the same - perfect. On to Windoze. I had a few web pages and stuff which I had downloaded so I thought I would print from them - total rubbish, pages and pages of squiggles and symbols. OK I thought check the drivers. All the drivers were there and seemed OK. When printing starts a little box with a drawing of the printer on it appears and that should have a big red cross on it if something was wrong. (according to the 'manual')

Right, on to the EPSON helpline. "Hello can I help you ?". I explained the problems "Do you have a Bi-Directional printer cable ?" he asked. "I don't know", I said "Oh well the printer needs a bi-directional printer cable" "No it doesn't" "Yes it does" "No it doesn't", this was becoming a bit too pantomime for me, "The printer does not need a bi-directional printer cable. Maybe your software needs it because it wants to draw a picture of the printer on the screen with paper coming out of it and waste even more memory and time. I know what the printer looks like - it is sitting beside the computer!"

I was on fairly safe ground here because I knew he was wrong. My QL only had a mono-directional printer port. He then went on to talk about looking in the BIOS and I asked a few questions about that. "Oh I don't know much about that really", he said "They only trained me on the printers. This was the TECHNICAL helpline and I was paying 50p a minute to talk to someone who knew less than I did.

The answer, of course, was the automatic switch which was mono-directional but what a load of rubbish all this is. Why write software that will only work if it can draw a picture on the screen? Answers on a postcard please you will find the words 'stupid', 'PC-User', 'pointless' and 'wasteful' spring to mind. Some Cannon Printers are even worse because you cannot change the ink cartridge without clicking on an icon on the desktop.

Jumping to Conclusions - Now an Olympic Sport.

I was guilty of a bit of conclusion jumping during the past few weeks and it was only the perception of my customer which led us to the right solution to the problem. I sold a new Super Gold Card to Mr Spencer recently and he reported that it would not format disks properly.

At first we suspected the cabling which can sometimes lead to these problems but a close examination of his cables proved fine so I asked if he could send the unit back. He returned the SGC to me and we ran a few tests. Keith Mitchell, who is always my mentor in thing hardware, ran it up on his Aurora system and that worked fine. I then ran it up on a standard QL and tried it out with a range of disk drives. No problems there either.

We were puzzled by this and the theories about why the SGC would not format disk became more and more complex. Could it be the voltage? could it be the small 8473 chip on the Super Gold Card which controls the disk actions?

The real answer was supplied by Mr. Spencer himself. It seems he had a set of drives which were built to be ED but, when the ED drive failed to catch on they were sold on as HD drives. Some of these drives have part of their electronics removed but many function perfectly well as HD drives. If you get a chance open the little door on the front of you disk drive and look to see if the extra sensor is there. The problem he had was related to the FLP_JIGGLE command which is mentioned in the manual but only in reference to the ED drive - which he thought he did not have. Turn the jiggle on and it works!!

Another customer who had bought a Super Gold Card reported that it would not print. Again I went for the physical side of the problem. I sent a new cable still no printing. I sent a new Super Gold Card - still no printing. This was doubly strange because I have two disks of test software that I run on all Gold and Super Gold Cards. They run a series of test on the boards to see if everything is OK and test the clock, disk drives and finally print a line from the parallel port of the SGC.

Just as I was beginning to suffer a total meltdown I got a phone call. 'It's OK,' he said, 'It's all printing fine now' 'So what was wrong ?', I asked. 'It's OK. Everything works now', he replied This was a bit evasive and I wanted to know more. 'What did you do to fix it ?' 'Er... well... I suppose if you configure your applications to print from PAR you should put SER_USE PAR in the boot file should you ?'

'Arrrrrrrrrrrgggggggggg!'

It's a Virus!

At the start of February I fell victim to the dreaded 'flu' virus and Q Branch ground to a halt for two weeks. It was then that I realised what a drawback the shop was because people were calling the shop number and just getting the answering machine. Even worse, I was unable to attend the Radio Rally in Cambridge that Qubbesoft T.F. Services and ourselves had promised to go to. I apologise for anyone who was frustrated by our non-appearance and also to those who tried to contact me during my illness. On the first Monday of the illness I came home from the shop and tried to access some files on my main Aurora machine and found a blank subdirectory. Since I was a bit befuddled at the time I gave up and went to bed for three days.

When I emerged from bed I tried again only to find more corrupted directories and programs not working. This is the first time I have considered the concept that you might be able to pass a virus between a human and a machine.

I tried several solutions to fix the problems but, in the end I had to re-format the hard disk and restore the files. At this point I said a silent thank you to Mark Knight for 'The Knight Safe' and Syquest for the EZ Flyer. Although I backed all of the files up with The Knight Safe I made a complete mirror image of WIN1_ onto my Syquest. When I had re-formatted the hard disk all I had to do was to use the WIN_DRIVE command from Qubide to allocate the Syquest to WIN1_ and run the boot program to have my system up and running. From then on it was a simple task to use Cueshell to copy all of the files and sub-directories back onto the main hard disk and restore the system with no loss of data.

If you have a Qubide and either a Syquest 135 or EZ-Flyer or an Iomega Zip Drive it is well worth making such a mirror image for peace of mind, DATA security and a fast restore.

What do you use yours for?

I recently had a letter from Phil Thompson of Thompson's Furniture Makers and he sent me a couple of photographs of one of his QLs driving the lathe which makes several of the parts used in his furniture. He has, apparently, written a series of programs to control the movements of the lathe and each program is related to a specific part to be produced. He actually sent Tony Firshman a finely turned leg (wooden not human) as an example.

This ingenuity impresses me. Are there more people out there who have new or unusual uses for their QLs? I would really like to hear from them. Maybe they would like to write a short piece for the magazine about it I am sure that others would be interested too.

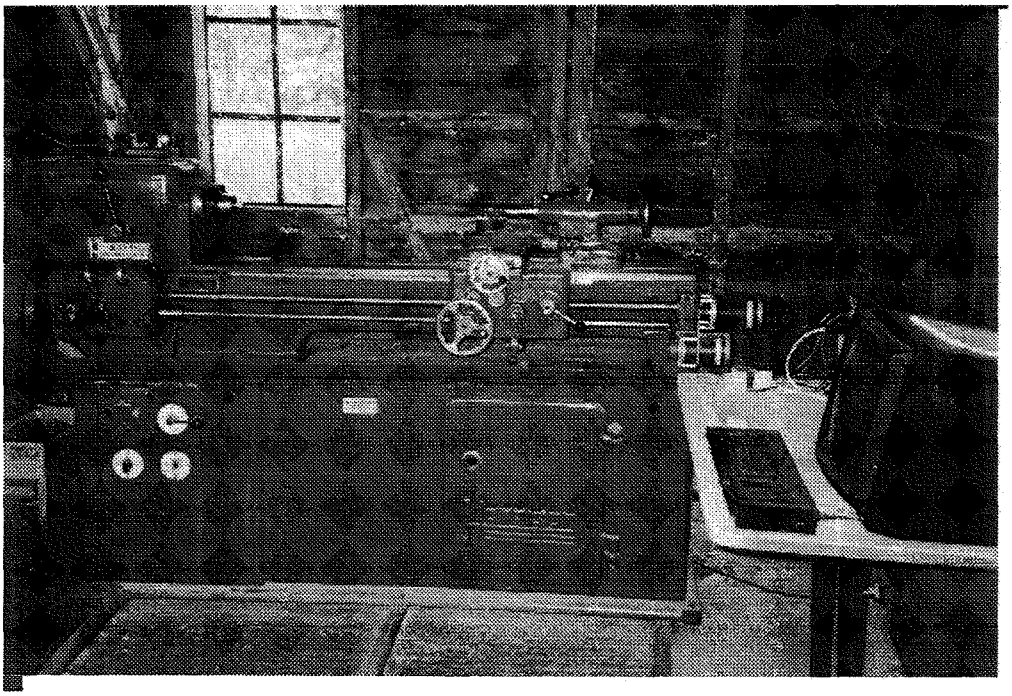
Tony's Revenge

As many of you may have read in my last column, Tony Firshman seemed to accept my near destruction of his car on the Croatian trip with a degree of equanimity. "I have expensive insurance," he said, "It covers me for any driver and for business use".

This was all very well except that, in the manner of all insurance companies of my knowledge, all money is passed through a diode. It will only flow in one direction - from the insured to the insurers. Any attempt to reverse that flow is met with a system overload and a frantic scrabbling about by the insurer to find a way to avoid parting with any money whatsoever.

True to form Tony's insurance company said yes he was covered for any driver and he was covered for business use - just not at the same time. How any human being is ever supposed to work this out from the total gobbledegook that passes for an insurance document is beyond me but the result is - no money for Tony. It seems that when you take out an insurance policy what you are doing is insuring the owners of your insurance company never get to be poor.

Inspired by this Tony managed to take his revenge on me at the last Eindhoven meeting. Just before we left the ferry (once again in Tony's car) he rummaged around in the back to get the tools to adjust the headlights. As we drove off it became evident that he had not shut the tailgate. The evidence for this was one of my cases laying on the ramp in several pieces. A woman in a BMW looked at the mess in front of her car and then calmly drove over my cashbox.



Amazingly enough my MinisQL / Syquest EZ flyer and EPSON Stylus 800 printer survived their flight intact and worked flawlessly when I set them up at the show. I am now minus one or two keys on my keyboard but apart from that and a battered cashbox all is fine.

A Word for the Master

One last thing. In the QD 98 review in the last issue Dave Westbury mentioned a bug in Master Spy. This bug has been fixed in the current version so I think an upgrade is in order Dave.

The QL Show Agenda

QUANTA ANNUAL GENERAL MEETING AND WORKSHOP

being held at Walton Park Hotel Clevedon, North Somerset near Bristol on

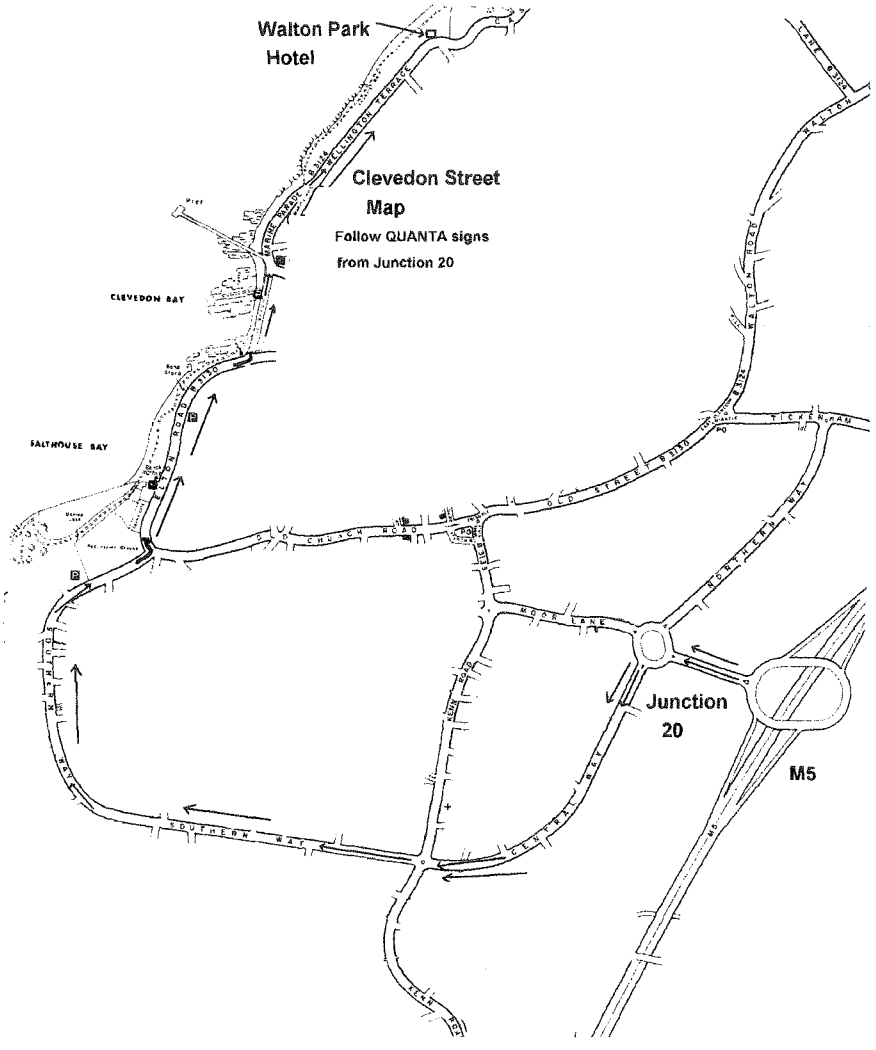
Sunday, 25th April 1999 10.00am to 4.00pm

*** separate traders' and users' areas**

*** product presentation and special interest area**

*** Bring and Buy stand**

*** Help and advice clinic**



The event is well signposted from Junction 20 of the M5, overlooking the Bristol Channel. Don't miss it!!

For details of local accomodation contact Mike Ashford, Tel 01934-415416.

The QL Show Agenda

International QL Show - Eindhoven

Saturday, 17th of April 1999! Don't miss it!

Held at its usual Venue (St. Joris College). The show starts at 8am and ends at 6pm. Most dealers will be there - we're hoping for other "famous" guests as well.

Two US-Shows coming up!

US QL-East Coast Show 29-30 May

The show will be at Wesley Hall, St. Andrews United Methodist Church, in Spring Lake, NJ, about 60 miles south of New York City. QL-East will include vendors, work shops, and some interesting tutorials by Bill Cable, Herb Schaaf, and Simon Goodwin. The show includes other Sinclair computers, TS-2068, ZX81 etc. A block of 20 rooms are available at Budget Inn, tel:732-775-7200. To obtain the show rate of $50 per night for 1 or 2 people for Friday and Saturday, mention the QL Show and make reservations by 1 March.

Spring Lake is a seaside resort area about 75 minutes from airports at Newark, Atlantic City, or Philadelphia. It's about 2 hours from JFK airport. Local information including (probably free) transportation from JFK, can be obtained from Local Host Bill McKelvey: Email **MCKELVEYW@delphi.com**

AGENDA:

FRIDAY night 6PM till ? dinner at Cobblestone Restaurant.

SATURDAY 9AM to 4:30PM Vendors, Tutorials, and Workshops.

Fee per family is $10 if paid by 1 May, $12 at door. Fee includes light lunch and coffee/tea throughout the day. SAT. EVE. 6-9 PM Banquet, panel discussion, and entertainment. $20 per person. Late nighters retreat to Bar and Grill. SUNDAY 11AM at McKelvey's home there will be additional tutorials and informal discussions. Includes light lunch and beverages free.

Pay advance fees to show treasurer: Bob Malloy, 412 PACIFIC ST.,MASSAPEQUA PARK, NY 11762. Overseas attendees may obtain advance rate by notifying Bob **bmalloy@idt.net** and paying after arrival. Questions about the agenda should be addressed to Al Boehm **boehm@ziplink.net**. Updates will be added to the NESQLUG web page **http://www.airnet.net/boehm**

US QL-West Coast Show

Tim Swenson writes: The show will be held one week after the East Coast Sinclair Show (so Europeans can hit both shows in one trip). It will be held in Union City, California. Union City is on the east side of the San Francisco Bay, about 15 miles from both Oakland and San Jose. The best airport is Oakland, then San Jose, then San Francisco.

The selected hotel is the South Hayward Motel 6 (right on the border with Union City). It has a Denny's Restaurant and a McDonalds on its property. Within walking distance is a Taco Bell. If the Motel 6 fills up, there is a Super 8 motel nearby. The Motel 6 is reachable by the Union City Transit bus (cost is $1 per person).

The meeting will be held at the Mehran Banquet Hall. It's really an empty store front used by the adjacent Mehran Pakistani restaurant. It is in one of the two major shopping areas of Union City (lots of places for food) and is within walking distance of BART (the local commuter rail). Buses from the Motel 6 go to the BART station. BART also goes close to the Oakland Airport, with a shuttle bus connector. For vendors, I have a station wagon that I can use to make a few trips. It's about 4-5 miles between the motel and the meeting site.

No plans are in place for where the dinner after the meeting will be held. I don't think many will be brave enough to try Pakistani food.

The night before the meeting, I plan on having a Bar-B-Q at my house (about 1.5 miles from the motel). The fare will be burgers, hot dogs, chips (crisps for you English types), and soda.

For any Europeans coming to the show, I am offering a visitor info service. Let me know how long you will be in the area, what you are interested in seeing (even just an area of interest), how you plan to get around (rental car, public transit, bike, etc), and I will figure out what places you might like, where they are, and how to get there. I will provide maps, public transit information, and so on. If you are coming this far to our show, this service is the least I can do.

For anyone coming, even though the show will be held in May/June, please bring a jacket or sweater. The nights do get cold in the Bay Area. Even though the daytime high may be in the 90's, the overnight lows can be in the 50's.