QL Today

Volume 10
Issue 4
January-March
2006

ISSN 1432-5454

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

Colours

The HOME Thing

Start Here! The BOOT file

Illustrating a Talk

Special QUILL printer drivers

Sound Advice

QL on the Internet Part III

62 Pages!

www.QLToday.com

Contents

Advertisers
in alphabetical order

QL Today

ISSN 1432-5454

German office & Publisher:
Jochen Merz Software
Kaiser-Wilhelm-Str. 302
47169 Duisburg
Germany
Tel. +49 203 502011
Fax +49 203 502012
email: JMerz@j-m-s.com
email: QLToday@j-m-s.com

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

Editor:
Geoff Wicks
56 Peveril Crescent
West Hallam
Derbyshire DE7 6ND
United Kingdom
Tel. +44 115 9303713
email: gwicks@beeb.net
email: QLToday@j-m-s.com

Co-Editor:
Bruce Nicholls
38 Derham Gardens
Upminster
Essex RM14 3HA
United Kingdom
Tel +44 20 71930539
Fax +44 870 0568755
email: qltoday@q-v-d.demon.co.uk
email: QLToday@j-m-s.com

QL *Today* is published five times a year, our volume begins on beginning of June. Please contact the German or English office for current subscription rates or visit our homepage www.QLTODAY.com.

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. We prefer ASCII, Quill or text87 format. Pictures may be in _SCR format, we can also handle GIF or TIF or JPG. To enhance your article you may wish to include Saved Screen dumps. PLEASE send a hardcopy of all screens to be included. Don't forget to specify where in the text you would like the screen placed.

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

The deadline for the next issue is the 1st of May 2006

Editorial

by Geoff Wicks

Norman Dunbar has confessed. It was not so much a confession as a public declaration of guilt. In the headline to his DocBook article in the last issue of QL Today he used the dreaded words "Off Topic".

As it happens we disagree with him. Jochen added a note that the article was relevant because his work was to be found on the "Insider CD" that came with the last issue. Even without the "Insider CD" I think we were still right to use the article. In the QL community we are too few in numbers to justify the expense of professionally printed documents. Norman's article was relatively short and contained valuable information on how we can improve the appearance of our documentation.

"Off Topic" also raises its head in this issue in the third part of the "Internet on the QL" series. Cynics who read this may accuse us of wanting to change the title of the magazine to "Linux Today". We did have problems with this article because of its length and technical content and, for reasons of space, decided it was unsuitable for the last issue. Nevertheless, we have readers who wish to access the internet from a Q40 or a Q60 and for them this article could be valuable.

Nor are our problems over. We have long wanted more articles on hardware and have been promised a series. However, these are long, technical and, more important, do not begin with QL hardware, but with other Sinclair machines. The author has reminded us that on our cover we claim to be "The Magazine about QL, QDOS, Sinclair Computers, SMSQ..." and therefore maintains his first articles are not "Off Topic". He may have a point. Our cover aims give us some flexibility and we have readers who are interested in modifying other Sinclair computers.

Sometimes we have to say "no". Last year we received an article from a skilled c-programmer, which, although of high quality, we felt was unsuitable for QL Today. It was just too long, too technical and had no direct relationship to the QL.

I am a person who enjoys testing boundaries, but that means I need keeping in control. The boundaries between what is and what is not "off topic" are not clear cut, and we shall not always get it right. Important factors are the length of an article, its technical nature and how far the content is of interest to QL-ers. We would welcome the views of our readers about what is and what is not "off topic".

If you have written an article or would like to write one you fear may be "off topic", please contact us for advice. We may be able to suggest changes to make it more relevant for our readers.

A trouble free Year

Marcel Kilgus has announced a update of QPC Print, which now supports landscape printing. It can be downloaded by registered users from his web site:

http://www.kilgus.net/qpcprint/downloads.html

The download is protected and users should first note their registered name and number which can be found by clicking on the ABOUT button in QPC Print. They should then follow the instructions on the web site. The download file is 1.53Mb.

QPC Print has now been available for over a year and appears to be a trouble free program. Traders report they have had few queries from purchasers and the Quanta helpline only one request for help. One program that has caused some difficulty is Archive, but usually these problems can be solved by using the spoolon and spooloff commands.

QL Today would like to run a trouble shooting article on QPC Print and would be pleased to hear about readers' experiences.

XCHANGE Improvements

Recently ***Marcel Kilgus*** has been updating Xchange almost faster than QL Today can keep up with him. Version 3.90N can be downloaded from his website:

http://www.kilgus.net/smsqe/xchange.html

You will need a minimum resolution of 512 x 512 to use this version.

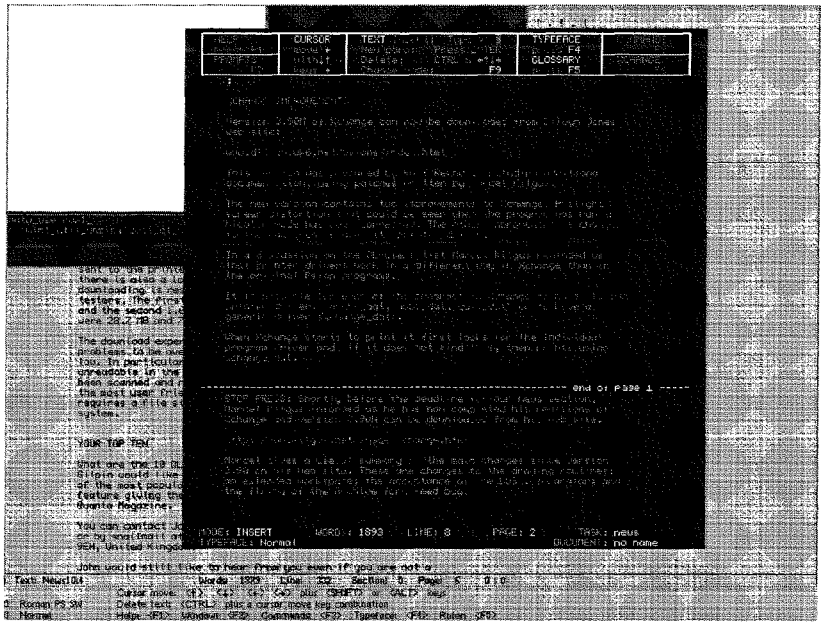
There have been four main improvements since version 3.90L.

1: Changes to the colour drawing routines to make them colour mode agnostic. Quill's status area now uses a slightly diffferent font colour in high colour mode to differentiate it from the edited text.
2: An extended working area with the main window twice as high as before.
3: All modules now accept the DOS separator. Thus "quill.doc" is recognised as well as "quill_doc".
4: An Archive form feed bug has been fixed. Unlike the original Psion Archive, the Xchange version did not ignore the "lines per page" feature in the printer driver. This gave problems to some users.

In this connection Marcel reminds us that printer drivers work in a different way in Xchange from the original Psion programs.

It is possible for each of the programs in Xchange to have its own printer driver, (archv_dat, abba_dat, quil_dat) or to have a generic driver (xchange_dat).

When Xchange starts to print it first looks for the individual program driver and, if it does not find this, then prints using xchange_dat.

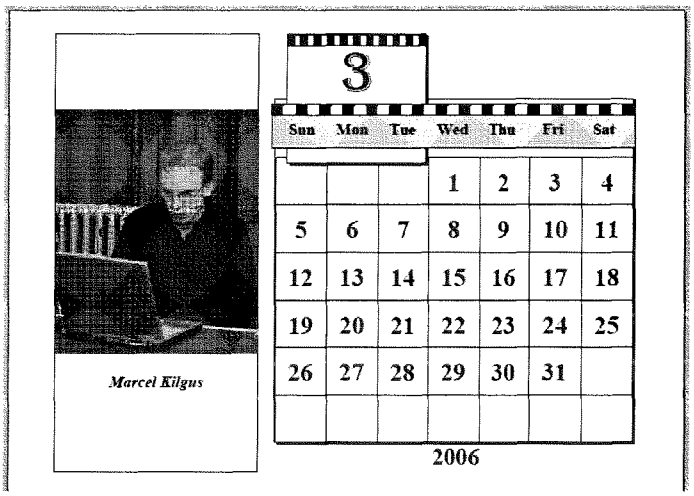


QL Pin-Ups

New on Dilwyn's site is a calendar printing program. ***Dilwyn*** writes:

"Multi-Cal is a new Calendar printing program by Dilwyn Jones. The freeware program goes a step further than most QL calendar makers in that it offers a choice of 6 layouts including 12 month per page, planners and diary layouts, is written in documented uncompiled BASIC to let you modify it to your requirements, can print calendars starting on any day of the week and also lets you save the calendar as a plain text file for import into a word processor or DTP program if you wish to tart up the calendars a bit with various fonts and text sizes, or add pictures etc. Of course, it will print the calendar too. 6 sample plain text files generated with the program are included.

Multi-Cal may be downloaded from
http://www.dilwyn.uk6.net/utils/index.html
Also available is the QaLendar 2006. This is a QL-orientated calendar you can download, in Word DOC format, and includes pictures of some well known QL personalities (but don't let that put you off!).
QaLendar 2006 is available for download from:
http://www.dilwyn.uk6.net/gen/calendar/calendar.html



REFORMAT

Dilwyn Jones writes:
"I have added a new Reformat program to the files page on my QL software download site.
Reformat is a rapid reformatter for floppy disks (DD, HD, or ED) already in QL format. It is useful for bulk erasing of disks, for example. The program is based on the non-pointered QFormat program by Norman Dunbar (with his permission), but this program is called Reformat, needs Toolkit 2, direct sector access (works on most disk interfaces) pointer environment and Window Manager 2. The program uses System Palette colours. It is usable with recent versions of SMSQ/E, or with QDOS if you have pointer environment version 2 or later.
www.dilwyn.uk6.net/files/index.html
And on the Launchpad download page you can find a minor revision of the PicView graphics viewer accessory program, with a small bug fix in the resize routine and a small change to the Quit routine - if the program was started with a filename command parameter (e.g. from File Info II) it no longer queries ESC to quit.
www.dilwyn.uk6.net/gen/launchpad/ launchpad.html

TIDY-UP Version 2 now available

Dilwyn Jones writes:
Some time ago, at Jochen's suggestion, I wrote a program called TidyUp to assist with mundane text file transfer. This little program did character set conversion, adding and removing carriage returns, removing email quote symbols and all the other little jobs that made it easier for a magazine editor to strip down emails into usable text.
I've now converted it to GD2, or more specifically Wman 2 - it uses your current system palette colour theme, so you can use it in your established colour theme for consistent appearance with your other programs.
The program is freeware and available from my web site:
www.dilwyn.uk6.net/filetran/index.html

Yet more Dilwyn

Finally *Dilwyn* announces that the Computer One Pascal manual is now on his site:
"Derek Stewart has kindly scanned the Computer One Pascal manual to go with the package available on my web site (Computer One Pascal has kindly been made freely distributable for the QL).
Derek's work is a zipped searchable PDF file. It needs a recent version of Adobe reader to load the file - some versions such as the version 6.0 installed on my computer give an error on loading, but seem to display it OK). It is approximately 1.8MB in size, so think carefully before downloading as it may take 10 minutes or more on dial-up.
www.dilwyn.uk6.net/language/index.html

QDT Online

Jim Hunkins writes:
"I have taken my QDT updates web page on line. you can now download the demo and updates.
http://jdh-stech.com/QDT/downloads.html
The current release is 1.02r. If you have 1.01r already installed, this downloaded update will work for you.
If you have an older version, please request a full install version from either your distributer or send an email with the info as described at:
http://jdh-stech.com/QDT/qdt.html#Updates
Please use the link at that location for the email request.

ZeXcel Patch

Phoebus Dokos writes:
I've completed a smallish (very ugly but works) patch (needs TK2 and Turbo Toolkit) for ZeXcel 0.33 with which you can replace the executable contained ROMs with other ones.

Currently tried and working ones:
Gosh Wonderful v.0.3 (non single keyword entry, numerous bug fixes - type REM and ENTER for the extra commands) - 48K or 128K (Replace only 48K portion)
SDOS 64+ (64 column ROM with bugfixes) - 48K only
Pentagon (TRDOS not working) - 128K only

Tried but NOT working:
Plus 2 ROM (Although it is mentioned as a ZX Spectrum +2 compatible emulator, ZeXcel actually emulates the 128K -See the ROM which states Sinclair Research instead of Amstrad for a "clue"). It will start properly, however it has trouble switching to the 48K model. My guess is that there's some patching of the ROM involved.
SE Basic ROM - It actually loads in both 48K and 128K mode but the tokenizer reappears where it shouldn't.

For prepatched executables and the patch itself, drop me a line.
The patch may be downloaded from
www.dilwyn.uk6.net/spectrum/index.html
Phoebus R. Dokos **ql.css@dokos-gr.net**

Congratulations

QL-Today congratulates John and Sarah Gilpin, treasurer and secretary of Quanta respectively, on their Ruby Wedding Anniversary (40 years) this month.

QUANTA

Members of Quanta will have noted a much improved print quality in the Quanta Magazine arising from changes in the production process. These changes have been introduced as the first step to producing the magazine in a downloadable form for those members who prefer to receive it in this way.
The magazine is now produced in two resolutions. The version sent to the printer is of higher resolution than previously and there is also a low resolution version for downloading. So far downloading is restricted to committee members and a handful of testers. The first of the download issues had a file size of 907 Kb and the second 1.64 Mb. (The sizes of the files sent to the printer were 28.7 MB and 7.2 Mb respectively.)
The download experiment has shown there are two technical problems to be overcome. Firstly the quality of the graphics is too low. In particular some of the advertisements are practically unreadable in the download version, probably because they have been scanned and not imported from the original images. Secondly the most user friendly way of printing out the downloaded image requires a file size too large for downloading on a non-broadband system.

Your Top Ten
What are the 10 QL programs you most frequently use? John Gilpin would like this information to assist him in compiling a list of the most popular QL software. He then hopes to run a regular feature giving the latest version of each of the programs in the Quanta Magazine.
You can contact John Gilpin by email at **treasurer@quanta.org.uk** or by snailmail at 181 Urmston Lane, Stetford, Manchester, M32 9EH, United Kingdom.
John would still like to hear from you even if you are not a Quanta member.

SINCLAIRQL.INFO Website

Javier Guerra has set up a new English language QL news web site:
sinclairql.info

Structured SuperBASIC

Tim Swenson has now released Structured SuperBasic 2.7. He writes:
"After much procrastination, I've finally finished and released Structured SuperBasic (SSB) 2.7. Biggest changes are:
- Now compiled with Turbo.
- Now supports three languages (English, German, Italian). Support for other languages is fairly trivial once I have the text translated.
- Includes documents on how to use SSB with Make, RCS, & MicroEmacs.

It can be found on my geocities site:
www.geocities.com/svenqhj/

SUDOKU

George Gwilt's sudoku solving programs can now be downloaded from the SQLUG website:
www.jms1.supanet.com
To run the programs you will have to have an SMSQ system capable of running GD2 colours on high resolution screens. In addition to the traditional 3 x 3 grid, George has written a program for 4 x 4 grids. The zip file includes the source code for each of the programs.
George hopes to demonstrate both programs at the Manchester show.

STOP PRESS NEWS:

More Psion Changes

Shortly before we went to press, QL-Today received a demo disk of ***Roger Godley***'s updates of the Psion programs. This includes GD2 colour, high resolution screen versions of Abacus and Archive. We shall include a full news report in our next issue.

Roger Godley hopes to attend the Hove show.

Editor's comment:

It is good to see that XChange is modified now, so that it can be used on modern systems and also uses some of the features like high-res or more colours. But is it to the benefit of the users that two authors produce two differently patched versions? Competition leads to progress as we can see, but on the other hand it may be a waste of resources. What do YOU think? Tell us for an XChange special next issue.

Oooops...

We apologise for a few errors that appeared in the last issue of QL Today.

SUDOKU

Our readers discovered that a `%` was missing after the third `grid` in line 1330.

The last line of Ian Pine's sample sudoku input file on page 27 disappeared during the production process. This should read

010407030

Some readers without a text editor asked how they could produce their own input file. They could use a few lines of SuperBasic:

```
90 OPEN_NEW #4,ram1_sudoku_dat
100 PRINT #4, "460300509"
110 PRINT #4, "000000000"
120 PRINT #4, "000709010"
130 PRINT #4, "290008000"
140 PRINT #4, "700605003"
150 PRINT #4, "000200095"
160 PRINT #4, "000000000"
170 PRINT #4, "000000000"
180 PRINT #4, "304007081"
190 CLOSE #4
```

MODE32BMP

There was a fault in the program mode32bmp, which is in the folder gd2conv on the QL Documentation Insider CD. The fault also appeared in the original article in QL Today, Vol 8 Issue 4, page 43.

Dilwyn Jones writes:

"Unfortunately we noticed a bug in one of the programs on the cover CD, a basic program used to convert QL screens/pic files to Windows .BMP graphics files. Although it was too late to change the CD, only a minor correction is needed in a program called mode32bmp_bas, located in the directory GD2CONV_GD2toBMP_ :

```
270 GET #3,tmp% : ql_bpl = tmp% : REMark bytes per line in QL file
```

This forces the GET command to fetch a 2 byte integer instead of wrongly trying to fetch a 6 byte floating point value from a QL _PIC file."

On the QL-users email list, Duncan Neithercut suggested changes to the program to convert mode 33 files (Q40/Q60) to bmp:

```
1900 DEFine PROCedure Q40colours
1905 FOR X = 0 TO pw%-1
1910   blue =  (PEEK(qlad+1)&&62) DIV 2
1920   red =  (4*(PEEK(qlad)&&7))||
       ((PEEK(qlad+1)&&192) DIV 64)
1930   green = ((PEEK(qlad)&&248) DIV 8)
1940   intensity = (PEEK(qlad+1)&&1) :
       REMark 0 or 1
1950   POKE pcad,(8*(blue))+intensity*7
1960   POKE pcad+1,(8*(green))+
       intensity*7
1970   POKE pcad+2,(8*(red))+intensity*7
1972   qlad = qlad + 2
1975   pcad = pcad +3
1978 END FOR X
1980 END DEFine Q40colours
```

ANAGRAMS

Stephen Poole writes:

"Today I just happened to read one of the long english words used in the anagrams program, in a book on linguistics. Alas I had mis-spelt it. (I had held it in memory from school some 45 years ago and hardly used it since...). It should have read: floccinaucinihilipilification, which is much harder to pronounce than my incorrect version of the word. The meaning is to qualify something as trivial. So I checked it out on the web, and came across an even longer monstrosity of a word, (even though the german language probably allows much longer constructions):

pneumonoultramicroscopicsilicovolcanoconiosis, which has the advantage of being relatively pronouncable and at the same time is THE longest word in the English language.

Perhaps Dilwyn will send us the exact spelling of the welsh word known as 'Llanfair P G'"

Gee Graphics! (on the QL ?) – Part 45

by H. L. Schaaf

Sudoku musings only, no program listing this time. Do you do Sudoku too? So do I. That's what I've been trying to get my QL to do. I've been working with a collection of 41 puzzles, collected from the local newspapers. Ian Pine's program solves all of them, taking from 1 to 6224 seconds, and does 25 of them in less than 2 minutes. My program solves 33 of them in 5 to 66 seconds and is stymmied by the other 8 of them after a minute or so.

There are many interesting web sites with hints, methods, etc to help humans solve Sudoku so I'll keep at it. Wayne Gould(1) provides Pappocom puzzles and offers 'how to do it' suggestions.

Wikipedia((2) has links to a glossary of terms and jargon and mentions something called 'Dancing Links' as the best computer algorithm. It uses 729 rows and 324 columns in a Boolean array. Perhaps I can teach it to my QL?

Wikipedia also has a section on the mathematics of Sudoku and some of my mathematical friends have wandered off into group theory to enumerate the possible 9x9 grids. Let me count the ways? One figure given for valid Sudoku 9x9 puzzles was 5,472,730,538.

I noticed that there were 8 simple equivalent symmetries, 4 mirror and 4 rotation. Some would solve faster that others in Ian's program (e.g. 5 done in less than 2 minutes, the other 3 much longer), but not in mine. Interesting?

We can also shuffle the stacks, bands, columns within stacks, rows within bands, each with 6 permutations. We could also make simple substitutions of the 9 digits in 9! or 362880 ways. Combining reflection, rotation, permutation, and substitution could lead to some 3762339840 variations of the same puzzle, and some of those variations could be easier to solve. So how many puzzles, what makes them easier, and how to index them? How would we recognize a puzzle as simply a variant of one we had dealt with before?

Another interesting site is the Daily Sudoku(3) with several variations. I wrote a little basic program to convert their 'classic' PDF file to a plain text file as used in Ian Pine's program. So my collection of puzzles grows.

And still unresolved, how to use my QL to create unique puzzles. Hirofumi Fujiwara(4) shows how to do it by hand, with reverence for beauty and symmetry.

I'd like to have my QL do something as nice as "Simple Sudoku" by Angus Johnson(5). I recommend you have a look at it, great graphics content.

I hope to return to the Lame curves, but for now, more time is being expended on the logic(?) of solving Sudoku with my QL. So please tell us what you've found, done, etc. with your QL?

Websites of interest:

(1) **Sudoku.com**
(2) **Wikipedia.org/wiki/Sudoku**
(3) **www.dailysudoku.co.uk**
(4) **www.pro.or.jp/~fuji/sudoku/index-eng.html**
(5) **www.angusj.com/sudoku**

Editor's comment:
Ian Pine's Sudoku solving program was appreciated by many readers, including Arnold Clarke, who has discovered one puzzle the program cannot solve:

100006490
000010006
000000050
000030005
075020000
008000009
300501040
200900000
810000530

Start Here - Part 4
Essential Information For 21st Century QLers

by Roy Wood

The BOOT is on the other Disk

The standard method for many people when they wanted to start a program in the original QL system was to place the disk in mdv1_ or mdv2_ and reboot. This was fine while there was only 128Kb of memory available and even OK when the available memory increased to 512Kb with ExpanderRam or similar expansion devices but today's users have large amounts of memory available to them and, more to the point, have access to hard disk drives with enough storage capacity to have all the QL programs ever written available at their fingertips.

Not only that but we now have extended resolutions, high colour drivers and subdirectories all of which were pipe dreams when the QL was first on the market. They are now all there all you have to do is reach out for them and the aim of this series is to help those who may feel a little left behind by QL development to get up to date. It is also a chance for some people to learn, or re-learn, some skills and to get a more efficient system. We have, so far, discussed the various aspects of the hardware we will have to deal with and, in last issue's section, we touched upon some of the basic software.

It is at this point that a boot file becomes an essential thing and too many people start off with the conviction that they cannot write one.

Don't Worry.

Anyone can write a boot file. If you can write down the directions to get to your house you can write a boot file because that is what a BOOT file is. It is a list of directions to tell the computer where things are and how to get to them. I would like, in this section, to tell you how to do it because without it many of the QL's capabilites will always be beyond your reach.

To start with there is no 'correct' way to write it. You can buy the most expensive editor available or you can use the one that comes free with all QDOS systems (the command line) - the choice is up to you. Almost every person that you meet will look at your boot file and say, 'Ah but you should.....' and my advice is to listen to them and then make up your own mind. I have learned a lot of useful tips from other people and I have heard a lot of rubbish at the same time.

No boot file is static. It changes as your system and your knowledge changes and that is the key to it. If it works it is a good boot file and if it does not do what you want then you can change it. You are in command.

What does a boot file do?

If you imagine that your system is a house then the boot file is the key that allows you to open the door and let yourself in. Not only that but it can also allow you to decorate and set up some of the furniture you may need. The programs and data are all rooms within that house and they all have specific functions.

When you buy a new piece of software it has its own boot file on the master disk and that is the key to that particular room but when you have a hard disk you can start building a skeleton key which will let you into any of the rooms on that disk.

There are certain things that a boot file does.

1. Opening Screens and Windows

Many programs put a 'splash screen' up as they open and these can contain logos and information about the program which is being run. There are specific commands in BASIC to load and display these screens.

If you are running SMSQ/E on an Aurora / Atari / QXL / Q40 / Q60 / QPC2 or other system with increased screen resolution you may find that the opening screen is just a small box compared with the full three coloured screens displayed on a standard QL. We discussed WINDOW commands in an earlier section of this series. Writing 'WINDOW' commands into the boot file will allow you to open the screens to the full size available on your monitor. Other commands are available to reduce or enlarge the screen resolution so you can make the text more legible or make things display in the way you want.

2. Setting Defaults

Some programs need to have specific default settings. The most common of these are:

PROG_USE - the program default.

DATA_USE - the data default.

DEST_USE - the destination default.

These will be discussed in more detail later.

3. Loading Resident Procedures

Many programs require resident procedures or extensions to be present before they will run. All Pointer Environment programs, for instance, need the three extensions WMAN, PTR_GEN and HOT_REXT and many other programs need similar extensions to run. The BOOT file provided on the master disks for these programs will do the loading for you but, when you have to write a master BOOT file for a hard disk system, you will have to list which programs are to be loaded and then make sure that the appropriate extensions are present.

One essential thing to realise here is that SMSQ/E has the three Pointer Environment files and the Toolkit II extensions built into it. If you load the three P.E. files after loading SMSQ/E you will probably be loading older and less compatible versions of the files which are already in existence in memory.

4. Setting Environment Variables

Environment variables are used by some programmers instead of Configuration Blocks. They have become increasingly popular recently and need to be set up before the programs they refer to are called.

5. Setting HOTKEYS and strings

You can use your BOOT file to set up various HOTKEYS which can be called at any point to perform set actions. The Pointer Environment allows the setting of HOTKEYS for calling programs or issuing a command to the 'command line interpreter' in BASIC. One thing to be remembered here is that the BASIC window must be available to do this. If BASIC is put to sleep then all you will get is a short 'BEEP' and no action.

One other useful thing to be set in the BOOT file is a string which can be called from a HOTKEY. Things like the current date can be set up and used in any program.

6. Running Programs

Finally, you can run programs from the BOOT file either directly or from simple HOTKEYS. This is where management systems such as TASK-MASTER and QPAC II come in. In my discussion of this I will only refer to QPAC II since TASK-MASTER is no longer available commercially.

All of the above points will be discussed in more detail in later sections but, before I launch into this I would like to say a few words about directory structure.

Sub-Directories

When you move from a floppy disk based system to a hard disk based system you immediately have a lot more space to play with. This can cause you a lot of problems if you do not give the matter some thought right from the start. On a disk you can only have a small number of files so it does not matter too much if they all exist in the root directory. It is fairly easy to page through them and find the one that you want. Once you move to a hard disk system the number of files you can store increases dramatically and it is very easy to lose files in the general chaos. This is where a sub-directory structure comes in very handy.

There is no set way to structure your files. You do have two definite possible dividing points in the form of DATA and PROGRAMS so setting up a subdirectory for each of these would be a good idea. You could also set up an extensions subdirectory so that you can store all of the resident extensions in one place. This makes sense when you have to update files because you know immediately where to find them. On my hard disks (I have three main systems and try to have the same structure on each of them) I divide DATA and PROGrams and then have smaller sub-directories within them to provide further compartments. In the PROG subdirectory for instance I divide the programs between full programs such as QD / QSPREAD / etc and utility programs such as CALCULATOR / CALENDAR etc. The DATA subdirectory is divided into DATA subjects like clip art, the Q Branch DATA, Quanta/QL Today articles etc.

There are also some programs which have their own subdirectories. Text 87, for instance, has a large number of fonts and anciliary programs. These are all stored in a directory of their own. ProWesS has a setup program of its own which automatically creates subdirectories and stores the files directly there so the ProWesS programs all have their own compartments on the hard disk.

I keep all the programs that I am writing in a subdirectory called (imaginatively) 'MYPROGS'. This is, again, a handy way to keep all of the code in one place for easy access.

One final slot on the disk is a TEMP directory. I find this invaluable to store files that you are either working on or keeping for a short time. Keeping them in a temporary subdirectory is a good way to avoid losing them in other directories and you can make it a habit to periodically flush out the TEMP directory to regain hard disk space.

Different systems store the hard disk information in different ways. If you are using a QXL or QPC system the files which appear to the system as WIN1_ are seen by the host system as a single file called 'C:/QXL.WIN'. QPC and the QXL can

have up to eight QXL.WIN files with names which can be configured from the program's config block. The QXL used to be able to only see one file on the root hard disk as a QXL.WIN file but, with later versions of SMSQ/E for the QXL it is able to have the same flexibility as QPC 2. The old rule, which is still true for SMSQ and early versions of SMSQ/E for the QXL is:
C:\QXL.WIN = WIN1_,
D:\QXL.WIN = WIN2_,
and so on. It is possible to use the DOS command SUBST to create extra 'virtual' hard disks and each of these can hold a QXL.WIN file. This method is memory intensive but does work well on DOS v6 and above.
Qubide users have a different system because the driver is completely different. You can create eight 'WIN' files on a single hard disk and link these in and out by using the WIN_DRIVE command. This is a very useful command because each 120Mb partition that is linked in will take around 100Kb of your system memory for its map. It is, therefore, a good idea to organise the data stored on these disks to the best advantage. WIN1_ must exist and is linked in on boot up so this is the best place to store all of the programs and the data which is in most common use. Other locations such as WIN2_ can be used for clip art and only linked in when they are needed. Until version 2.xx of the Qubide ROM you could not make a subdirectory if files containing the same name were already in existence. This means that if you have files called:
WIN1_TEST_first_bas
WIN1_TEST_second_bas
etc. you could not use the MAKE_DIR command to make WIN1_TEST_ as a new subdirectory. The hard disk drivers of the QXL and QPC could always do this and would then store the files in the new directory. With Qubide you had to rename or move the files to another place and then create the directory and move then into it. The later version (v2.00 onwards) of Qubide allows you to create directories in the same way as the QXL and QPC drivers do.
To create a new directory such as the one above go to the command line and type
```
MAKE_DIR WIN1_TEST_
```
this will create a subdirectory on win1_called 'TEST'. to read the directory type
```
DIR WIN1_TEST_
```

Full information on the Qubide directory structure and the use of Qubide commands can be found in the instruction manual. Information on the QXL and QPC directory structure and commands which relate to them are found in the Manual which accompanies the QXL and Super Gold Cards.
A good sub-directory structure is essential to maintaining a system and you should give the matter a lot of thought before committing yourself to it.

I. Getting Started and Choosing an Editor

So now we are about to start to write our BOOT file how exactly do we go about getting it all started and, more importantly, which editor do we choose to do it in ?
QDOS systems provide the user with a free built in editor and this one even has built in error checking so you can actually launch straight in using this. Assuming that you are using SMSQ/E or have activated the Toolkit II extensions by typing the command
```
TK2_EXT
```
at the command line you can launch the editor by typing
```
NEW
```
to clear any program lines that may be lurking in memory and then
```
ED
```
You are then presented with a flashing cursor in Window #2. Type in a line number (try 100) and start from there.
If you lean towards a more featured editor you could try using some of the other editors that are available. There are several editors that can be used to produce passable BOOT files and, if you ask different users you will get different answers as to which ones people prefer. Free editors include QED and MicroEmacs - the former supplied by some PD libraries as a reader for their files and the latter recompiled and made P. E. compliant by Theirry Godefroy. MicroEmacs is the best featured of these two with many commands available for formatting and re-structuring text files and is probably a bit overpowering for the casual or novice user.
Many people swear by (or at) the Editor. This was Digital Precision's flagship editor and again it certainly has a large array of features, many of which are again not applicable to writing BASIC programs.
My own preference is Jochen Merz's QD. This is a pointer driven editor with a lot of features but most of these are tailored towards the programmer. It does allow you to write the code first and number/re-number the lines later. It also has a built in Hyper Help system which will pop up definitions and syntax explanations for many BASIC keywords and is a great help.
In the end it all comes down to personal prefe-

rence so have a look at as many editors as you can and then make a choice based on what you actually want. For the purposes of this article all of the BOOT files that we are going to create can be written using the QDOS 'ED' editor so don't rush out to buy a new program - it will only be something else you will have to learn about and may get in the way of learning the process of writing the BOOT file.

II. Opening Screens And Windows

Most of us are used to the familiar three window setup of the QDOS Screen unless, of course, you have opted to start in TV mode. Once you fire up a QL system you are presented with a choice of pressing F1 or F2 (I will leave the Minerva options to one side and concentrate on the two QL options because they are not relevant to this discussion). I have, in a previous section of this series spent a bit of time discussing the windowing on a standard QL in order to fill in a few gaps or refresh your knowledge. Iwill, quickly, go over that ground again.

A Bit of History

The original QL was supplied with two options at startup so that users with monitors could take advantage of the wider screens available for these devices and get an 80 column display instead of the 40 columns available in TV mode. At the time this was an innovation because most of the home machines available then would only plug into a TV set. Pressing F2 opened only two windows #0 and #1 and gave a screen size of 256 x 256 pixels. Channel #0 is the console window which accepts commands and is displayed at the bottom of the screen with white 'INK' on black 'PAPER'. It is also used by some programs to display information while the program is loading. This information is usually the name of the program, version number and author / copyright message. Channel #1 is the default output channel and appears in the centre of the screen with white 'INK' on red 'PAPER'. Typing

```
PRINT "Hello"
```

at the command line will result in the word "Hello" appearing in the red window. If you change the line to

```
PRINT#0,"Hello"
```

the same word appears in the black window beneath.

If, however, you press F1 on startup the screen is very different. It is now divided into three windows one white, one red and one black. These three screens are given numbers and, as such, are labelled channel#0, channel#1, and channel#2. #0 is the console channel and it is that channel that accepts the SuperBasic commands and displays messages in the same way as when F2 is pressed. Again it has black 'PAPER' but this time the 'INK' is green. Channel#1 is, illogically, on the right hand side of the screen and this is displayed with white 'INK' on red 'PAPER'. If you have a disk in any of the drives and you call up a directory the list of files will appear in this window. This is the default display channel so typing

```
PRINT "Hello"
```

will cause the word "Hello" to appear there. Channel #2 is on the left hand side of the screen and has white 'PAPER' with red 'INK'. These two screens are rather confusingly labelled in reverse in the QL User Guide but you can test this out for yourself by typing in the lines above. If you have a Gold or Super Gold Card and you have installed Toolkit 2 (by typing 'TK2_ext') on the command line you can load a basic program and, then typing

```
ED
```

on the command line will list the program in channel#2.

The above concepts are very important when you move up to one of the more advanced systems. One mistake made by some programmers on moving from a standard QL to one of the advanced ones is to neglect the opening of windows in their own programs. Because the F2 screen has channels #0 and #1 opened and defined for you many people who started in this mode find that their own programs are displayed incorrectly on the new systems.

This problem also applies to some of the earlier commercial programs written in the QL's infancy, even some written to operate in monitor mode and for this reason the first screen opened by all current QL systems is the standard 512 x 256 pixel F1 screen as a compromise. It does not correct programs written for the F2 startup but it does help some of those which write directly to the 512 x 256 screen. At this point the next pitfall comes up. The original QL screen was at a specific address in memory so programmers could write directly to that address to display their screens. When the Gold and Super Gold Cards came along they had to patch the QL's memory onto their own boards and some clever programming was needed to make these programs work. The whole thing became impossible when the QXL, QPC and Aurora arrived so some of these programs will not display properly.

That, therefore is a good place to break this article until the next issue. In the next part we will start to build a BOOT file in earnest. Flex you typing muscles!

Illustrating a Talk

by George Gwilt

Introduction

I always like lectures to be illustrated. When the speaker's voice recedes into the background of my thoughts I find myself looking more carefully at the screen containing the latest piece of information deemed relevant. I will take note of its colour and check for any misspellings. Reluctantly I might even read and try and understand what is presented - which peculiar behaviour might lead me back to the words issuing from the speaker's mouth. You might deduce from this that I judge lectures without some distraction to be in danger of exhibiting an unwelcome degree of aridity.

In recent times all the talks I have attended have taken the same format. A lecturer stands, or sits, beside a laptop computer and fiddles with its keys from time to time while he delivers his diatribe and, lo and behold, from time to time different images appear on the large screen set at the front of the hall.

When I agreed to say something about Turbo at a recent meeting in Portsmouth I immediately purchased a laptop computer, installed the second most recent version of QPC2 (the most recent version of anything is likely to contain some errors, [though to be fair one would not expect that of Marcel Kilgus]), and set about preparing the illustrations designed to alleviate the boredom of the audience.

General Idea

What I decided to do was to arrange a set of images which would be sent from the laptop to a projector. I would scan through the images by pressing the up and down arrow keys. Since I wanted to be able to intersperse the images with the running of actual programs, some S*BASIC and some compiled executables, I realised that it would be easiest if the image-showing program were executable too and not run in S*BASIC.

I thus decided to write the program for image projection in S*BASIC and compile it (with Turbo of course). I would call the program "slides" since in the old days the projector would be fed with "slides" which are transparencies taken by a camera. My mother used to call the projection apparatus a "magic lantern".

Images

The images to be relayed to the projector were taken from the computer's screen and stored as "partial save areas" as defined in the official documentation for the Pointer Environment in the QPTR manual. The images were expanded to a size to fit the area on the projector's screen by the program "pr2win". I stored all the required images in a directory on the hard disk.

NOTE The program "pr2win" is available as part of "svscr.." from the SQLUG website:
www.jms1.supanet.com

Hot Keys

My talk required me to load some S*BASIC programs, to run these, to compile them and eventually to run the result. Experience told me that my typing skill would result in several, if not the majority, of the commands to be sent being faulty. To save time and embarrassment I decided to use the Hot Key system to cut down the number of key presses needed. The following five lines of a S*BASIC program show what I used:

```
2310 ERT HOT_PICK('t','slides')
2320 ERT HOT_PICK('p','pe_demo')
2330 ERT HOT_CMD('h','ex ram1_demo3_task')
2340 ERT HOT_CMD('i','ex ram1_test_task')
2350 ERT HOT_CMD('j','load demo3')
```

For example, I wanted to load my version of QPTR's demo_bas. I just pressed ALT/j to achieve this. In the midst of discussing my alterations to this program (so that it could be compiled by Turbo) I wanted to return to the "slides". Pressing ALT/t did this.

After compiling demo3_bas I was able to run the result by pressing ALT/h. That having been done I was able to return to the compiled demo3 by pressing ALT/p.

The command ALT/i enabled me to run the compilation of a program which had been loaded before I began the talk.

The Program "slides"

The source code for "slides" is given below and explained now.

The TURBO.. commands in lines 110 to 210

These commands obviate the need to set various conditions in the front panel which appears when Turbo is called by CHARGE.

`_windo 0`
None of the windows #0 to #2 is used
`_optim "Brief"`
A small program is requested and not a longer faster one
`_mode "‹64"`
Small branches will fit
`_struct "Struc"`
Main program lines precede procedures and functions
`_diags "i"`
Line numbers are included to help debugging
`_taskn "slides"`
This is the Job Name for the compiled program
`_repfil "ram1_err"`
The listing is sent to a file and not the screen
`_objdat 2`
A 2K dataspace is enough
`_objfil "ram1_slides"`
This is the compiled program's name
`_buffersz 200`
Ensures that Parser_task produces inter-code all in ram
`_objstk 350`
Sets the stack size of the compiled program to 350 (normal)

Setup

1. Two windows are opened, the first, #3, is for the main part of the screen. The second, #4, is for a title, which will remain throughout. To accommodate differing sizes of screen, the windows are set using SCR_XLIM and SCR_YLIM. In general programs destined for QLs with other operating systems than SMSQE I would use the TurboPTR function W_MAX which returns the x and y sizes in a long word. In the present program there was no need for this since I knew that I was operating under SMSQE.
2. nw% is set to the number of slides less 1.
3. pt% is set to point to the first slide.
4. The array fn$ is set to the nw%+1 slide names.
5. Finally the title is set in #4 by the procedure Ptf and the first slide set, also by Ptf.

Ptf

The procedure Ptf sets the image a$ in the centre of the window at #3, or #4, depending on the value of the parameter o% (0 for #4, the title window, and 1 for #3, the main slide window).
As already explained each image is in the official partial save area (psa) format. This means that each image can be placed anywhere in a window, (providing there is space for it!) by the official Trap #3 PE software, IOP.RSPW. TurboPTR contains the procedure BRSPW using that routine. It is this procedure, at line 2110, that is used to set the image in the middle of the window.
The final parameter of BRSPW is 1. This prevents the saved area being returned to the heap. The following instruction at 2115 returns the area to the heap. Why was that final parameter parameter not 0, which would have saved an instruction? The answer is that the area was claimed from the heap by ALCHP at line 2070 and not by the PE Trap#3 routine IOP.SVPW. It is therefore necessary, for safety, to return the area by RECHP and not via IOP.RSPW.

Do_It

The main part of the program is in the procedure Do_It. This is remarkably simple. Only ESC and the up and down keys have any effect. ESC quits the program. The up and down keys print the previous or next image if there is one: if not, a BEEP command is issued via Noise.

Finish

... does nothing. It was there just in case.

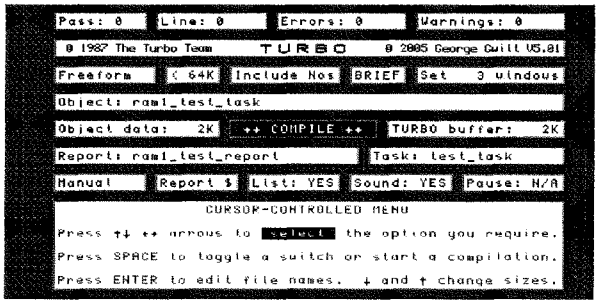
DATA lines @ 4000 onwards

The lines 4000 onwards list the images to be used. Each line gives the name of a file contining the psa information for the image to be loaded. Lines 4030 and 4050 each point to the same file. This is because I wanted to go back to an earlier image after showing the one whose address is at line 4040. I judged it better to repeat the image in the list rather than fiddle around pressing the up and down keys and going backwards and forwards.

Examples of images

To give an indication of what "slides" showed two examples appear here:

Title image, from "win2_talk_tok3_psa"



First image, from "win2_talk_sc7a_psa"

Source Code for "slides"

```
110 TURBO_windo  0
120 TURBO_optim  "Brief":REMark Rem/Fast
130 TURBO_model  "‹64":REMark ›64
140 TURBO_struct "Struc":REMark Freef
150 TURBO_diags  "i":REMark Omit/Include
160 TURBO_taskn   "slides"
170 TURBO_repfil "ram1_err"
180 TURBO_objdat  2
190 TURBO_objfil  "ram1_slides"
200 TURBO_buffersz 200
210 TURBO_objstk 350
220 :
1000 Setup:Do_It:Finish
1010 :
1020 DEFine PROCedure Setup
1025  DIM wx%(1),wy%(1)
1027  up=$D0:down=$D8:ESC=27
1030  wx%(0)=SCR_XLIM:wy%(0)=30
1040  wx%(1)=wx%(0):wy%(1)=SCR_YLIM-
      wy%(0)
1050  OPEN#3,scr:OPEN#4,con
1060  WINDOW#3,wx%(1),wy%(1),0,wy%(0)
1070  WINDOW#4,wx%(0),wy%(0),0,0
1080  nw%=16:REMark Number of items (-1)
      to show
1090  DIM fn$(nw%,40)
1100  RESTORE 4000
1110  FOR x=0 TO nw%:READ fn$(x)
1120  pt%=0:REMark points to 1st item to
      start with
1130  CLS#3:CLS#4
1140  Ptf "win2_talk_tok3_psa",0
1150  Ptf "win2_talk_sc7a_psa",1
2000 END DEFine
2010 :
2020 DEFine PROCedure Ptf(a$,o%)
2025  LOCal fln,asad,xz%,yz%
2030  REMark puts a$ in centre of window
2040  REMark o%=0 for #4, 1 for #3
2050  CLS#4-o%
2060  fln=FLEN(\a$)
2070  asad=ALCHP(fln)
2080  LBYTES a$,asad
2090  xz%=PEEK_W(asad+6)
2100  yz%=PEEK_W(asad+8)
2110  BRSPW#4-o%,asad,0,0,xz%,yz%,
      (wx%(o%)-xz%) DIV 2,(wy%(o%)-yz%)
      DIV 2,1
2115  RECHP asad
2120 END DEFine
2130 :
2140 DEFine PROCedure Do_It
2150  LOCal lp,inp
2160  REPeat lp
2170   inp=CODE(INKEY$(#4,-1))
2180   SELect ON inp
2190    =up:Do_Up
2200    =down:Do_Down
2210    =ESC:EXIT lp
2220   END SELect
2230  END REPeat lp
2240 END DEFine
2250 :
2260 DEFine PROCedure Do_Up
2270  IF pt%=0:Noise:RETurn
2280  pt%=pt%-1
2290  Ptf fn$(pt%),1
2300 END DEFine
2310 :
2320 DEFine PROCedure Do_Down
2330  IF pt%=nw%:Noise:RETurn
2340  pt%=pt%+1
2350  Ptf fn$(pt%),1
2360 END DEFine
2370 :
2380 DEFine PROCedure Noise
2390  BEEP 30000,24,151,99,2,9
2400 END DEFine
2410 :
2520 DEFine PROCedure Finish
2990 END DEFine
3030 :
4000 DATA "win2_talk_sc7a_psa"
4010 DATA "win2_talk_sd5a_psa"
4020 DATA "win2_talk_sd6a_psa"
4030 DATA "win2_talk_sd2a_psa"
4040 DATA "win2_talk_tik1b_psa"
4050 DATA "win2_talk_sd2a_psa"
4060 DATA "win2_talk_sd7a_psa"
4070 DATA "win2_talk_sd9a_psa"
4080 DATA "win2_talk_tak15a_psa"
4090 DATA "win2_talk_tak16a_psa"
4100 DATA "win2_talk_tak13a_psa"
4110 DATA "win2_talk_sd9a_psa"
4120 DATA "win2_talk_tyk1a_psa"
4130 DATA "win2_talk_sd9a_psa"
4140 DATA "win2_talk_sd8a_psa"
4160 DATA "win2_talk_tak5_psa"
4170 DATA "win2_talk_tak6_psa"
```

Eleven Lines of BASIC

by Geoff Wicks

At QL is 21 I made a proud boast. I was due to give a presentation on the new colours, and promised that in half an hour I could teach almost any one present to program in the new colours using just eleven lines of SuperBasic and no pointer environment jargon.
I heard a trader mutter under his breath, "They will be very long lines!"

Well, you can judge for yourself because I am about to repeat the promise. Almost any QL-Today reader, who works through this tutorial, can learn how to program in the new colours using a SuperBasic program of just eleven lines. And there will be no pointer environment jargon.
There is no point in reading this article unless your machine is fired up and ready to go. And, of course, your machine must be able to handle the new colours. That means you have QPC2, a Q40/60, a QXL or an Aurora card using a version of SMSQ/E above 3.00.

Can you remember the very early days of the QL, when we typed in green or white text on a black background and PC users in white text on a blue background? Some Quill users were disappointed they could not type in white text on a blue background.
Today you can change the colours Quill uses using just one line of basic:

```
PALETTE_QL 0,255,255,0,0,16777216,
  16777216,2607363,2607363
```

(Don't be frightened by these numbers. You will see why I am using them in a moment!)

After typing in this line, press enter and if you don't see a cursor press Esc.
Now start Quill. The colours are different, but it is not a pretty sight! Try running some other programs and you will see you have also changed the colours in those. Don't they look horrible? I probably should have warned you to have a sick bag ready.

There was a purpose in this little exercise. First of all it shows you how much power the new colours give you over your machine. But it also shows you what a mess you can make if you don't use the new colours wisely. And think how terrible programming would be if we continually had to enter numbers like those above.

So let's reset the machine and pause for a little bit of history and a little bit of theory. But first a promise. When we learn how to program in the new colours in the proper way, there will be no numbers higher than 65535!

The new colours first became available in 1999 for the Q40, and a year later for QPC and the QXL. There was a much longer wait before they were introduced on the Aurora Card.
For various technical reasons each QL platform had its own implementation of the new colours. For example, the Aurora card has only limited memory available and can only handle 256 colours compared with thousands for the other platforms.

Because each platform had its own colour implementation, in the early days you had to put a command in any SuperBasic program to tell it which colours to use. There were four commands:

`COLOUR_QL` were the standard QL colours. However you could modify these using the `PALETTE_QL` command we used in our Quill demonstration.

`COLOUR_PAL` was a palette of 256 colours. Again these could be modified by the user via the command `PALETTE_8`.

`COLOUR_24` were true colours or 24 bit colours. However no present QL platform can handle 24 bit colours. You were also told to enter values in HEX.

`COLOUR_NATIVE` were the colours that your QL platform used.

If you are finding this a little complicated you are in good company. Quite a few program developers and traders also found it complicated. In particular it was difficult for them to write programs that would look the same on all platforms. It was no co-incidence that in the first few years we had the new colours scarcely any software was written using them.

What the QL urgently needed was a much simpler way of using the new colours that was

common to all platforms. That happened in 2003 when Marcel Kilgus developed a new Window Manager able to display the new colours.
"Window Manager?" Isn't that something to do with the dreaded Pointer Environment?
Yes it is, and if you use the new colours you will be using part of the pointer environment. But don't panic. The pointer environment is already built into SMSQ/E and the good news is that you don't need to know anything about it to use the new colours. You can use them in a simple SuperBasic program, which is what we are now going to do.

So back to our machines. First of all we are going to be lazy and use the Toolkit 2 command to set our windows, and then we are going to use white ink on a black background:

```
10 WTV
20 INK 7 : PAPER 0 : CLS
```

Now we are going to modify Window #2 to give it a black background and a white border:

```
30 WINDOW #2, 300,154,50,50
40 PAPER #2, 0 : CLS #2
50 BORDER #2,2,7
```

Now an input command:

```
60 AT 2,3 : INPUT "Colour: ";colour
```

Now a block of colour:

```
70 BLOCK #2, 292,50,0,0,colour
```

We repeat this line twice with some slight modifications:

```
80 BLOCK #2, 292,50,0,50,colour+5
90 BLOCK #2, 292,50,0,100,colour+10
```

Now add a pause command:

```
100 PAUSE
```

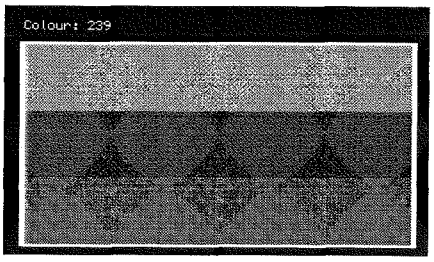
And now for something really naughty:

```
110 GO TO 60
```

Let's just print that again in one piece.

```
10 WTV
20 INK 7 : PAPER 0 : CLS
30 WINDOW #2, 300,154,50,50
40 PAPER #2, 0 : CLS #2
50 BORDER #2,2,7
60 AT 2,3 : INPUT "Colour: ";colour
70 BLOCK #2, 292,50,0,0,colour
80 BLOCK #2, 292,50,0,50,colour+5
90 BLOCK #2, 292,50,0,100,colour+10
100 PAUSE
110 GOTO 60
```

Now I count eleven lines, and not a long line in sight!
Think of a number between 1 and 245, and make a note of it. Run the program, input that number and you should have something like figure 1.



Now we are going to modify a part of this program to use the new colours. Break into the program (CTRL + SPACE) and put WM_ before each BLOCK keyword in lines 70 - 90. These lines should now appear as:

```
70 WM_BLOCK #2, 292,50,0,0,colour
80 WM_BLOCK #2, 292,50,0,50,colour+5
90 WM_BLOCK #2, 292,50,0,100,colour+10
```

WM means "Window Manager" and the new keyword instructs the QL to use the new colours. Run the program inputting the number you used before and Hey Presto!..... it looks just the same as it did using the old colours! This is because it is meant to look the same.
The Window Manager colours are not a single series of colours, but several groups of colours. The first group of 256 colours is exactly the same as the QL colours.
Now add 256 to the number you have just used and input this number into the program. This time you will notice a difference as you are now using the new colours. The second group of colours from 256 to 511 are the 256 palette colours.
We are going to skip over the 3rd group of colours as they cannot be illustrated by this program. These are the colours from 512 to 767 and it is better to avoid these altogether until you have gained confidence in using the other groups. These colours modify the appearance of most pointer environment programs.
Now add 768 to the number you have noted down and enter this. The group of colours from 768 to 1023 are various shades of grey. (Aurora users may have some difficulties with this part of the tutorial)
Let us make a summary of the groups of colours so far:

0 - 255	QL colours
256 - 511	Palette colours
512 - 767	System colours (do not use!)
768 - 1023	Grey shades

Now something different. We have to break into our basic program again and amend line 50 so that it reads:

```
50 WM_BORDER #2,2,1024
```

Run the program and you will see the border has changed to give the window a slightly raised appearance. Break into the program again and change 1024 to 1025 and when you run it this time you will see the border gives the window a slightly sunken appearance.

The group of colours 1024 to 1279 are the way in which programmers give their programs a 3D appearance. For obvious reasons you should only use these numbers with the WM_BORDER command.

Please note the colours of the border are determined by the colours your version of SMSQ/E uses to display the pointer environment. If you want to use your own colours, you have to instruct the window manager to use those colours, set the 3D border and then instruct the window manager to revert to the old colours. This is simple to do but beyond the scope of this article.

I have not done a great amount of research on 3D effects, but Wolfgang Uhlig, who has done far more, advises the use of the range 1024 to 1039 only.

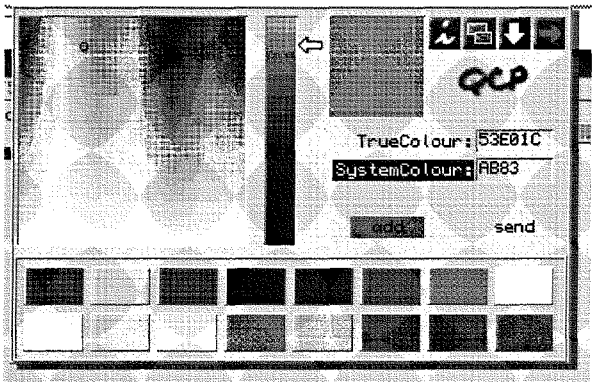
We now move into the realm of the big numbers. Try running the program with numbers in the range 16384 to 32767. These are stippled colours. At one time I wondered why we needed stipples, but George Gwilt tells me that they are often more interesting to use as a background than solid blocks of colour.

The last group of colours are the RGB colours and they are in the range 32768 to 65535. Again you can experiment with several numbers in this range. Aurora card users should remember, however, that although there are 16,383 stipples and 32767 RGB colours they cannot display the majority of these.

You now know how to use the new colours in your own programs without any knowledge of pointer environment programming.

First the easy bit. To use the new colours all you have to do is change PAPER, INK, STRIP, BLOCK and BORDER to WM_PAPER, WM_INK, WM_STRIP, WM_BLOCK and WM_BORDER respectively. When you do this, but don't change the values, the program will look just the same as before because the first 256 Window Manager colours are the same as the old QL ones. You can change the values of the colours at your leisure.

Now the difficult bit. How do you know which number to use for which colour?
There are a number of tools to help you.
Firstly the 256 palette colours. In the SMSQ/E manual you will find a short program to display these colours. To save you looking it up it is reproduced with this article (256palett_bas) in a slightly amended form to make it easier to compile. You can scroll through the colours by pressing the up and down arrow keys. Don't forget that to use these colours in your programs you will have to add 256 to the numbers shown against the colour.
Now the RGB colours. Reproduced with this article is the short basic program (WMRGB_bas) I wrote for my own use when I was learning about the new colours. What I like about this program is that it helps you to learn how the colours are made up from the Red, Green and Blue components. Press the r, g, and b keys to increase the proportion of red, green and blue respectively. To reduce the amount of each colour press shift plus r, g or b. The up arrow key increases all colours together and the down arrow key reduces it.
However if you want to do choose colours in a more professional way, the program to use is QCP which stands for QL Colour Picker, by Wolfgang Uhlig. You can download this from his website:
www.uhlich.nl/ql

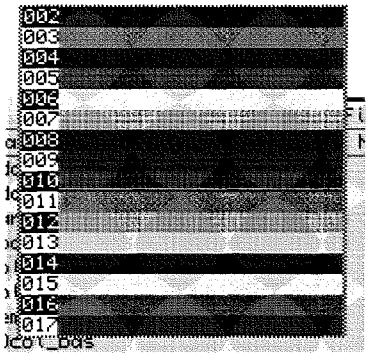


If you look at the screen shot of this program. You will see a large window on the left hand side with a cursor in the shape of a circle. You can move this cursor around to choose the colour you want, which is displayed in the square just right of centre. If necessary you can adjust the brightness of the colour by the strip next to the arrow. SystemColour gives you the number you have to input to get that colour in your program and you can put this in the stuffer buffer to be dropped in your program when you press ALT + SPACE.

You will notice the number is in HEX, but there is no need to panic as SMSQ/E has a routine for converting this to decimal:
The command `PRINT HEX("AB83")` gives the answer 43907. The window manager colours also allow you to enter hex values so you could enter that colour into your program as either:
`WM_PAPER 43907` or `WM_PAPER $AB83`
You now have all you need to know to experiment with the new colours in your own programs.

256palett_bas

```
100 OPEN #0,con : out=0
110 WINDOW #out, 16*10+2, 16*10+2,50,50
120 COLOUR_PAL : BORDER #out,1,0,1
130 bottom=-16
140 FOR i= 1 TO 16 : up
150 REPeat up_down
160 BGET #0,a
170 IF a=208 : IF bottom<255-16 : up
180 IF a=216 : IF bottom>0 : down
190  END REPeat up_down
200 :
210 DEFine PROCedure up
220 bottom=bottom + 1
230 PAPER bottom + 15 :SCROLL -10
250 i=bottom+1015 : I$=i
270 PAPER #out, i&&1 : INK #out,(i+1)&&1
280 AT #out,15,0 : PRINT #out,I$(2 TO 4)
290 END DEFine
300 :
320 DEFine PROCedure down
330 bottom=bottom - 1
340 PAPER bottom : SCROLL 10
350 i=bottom+1000 :I$=i
380 PAPER #out, i&&1 : INK #out,(i+1)&&1
390 AT #out,0,0 : PRINT #out,I$(2 TO 4)
400 END DEFine
```



WMRGB_bas

```
100 WINDOW 250,150,0,0
110 BORDER 2,248
120 PAPER 0 : INK 7 : CLS
130 red=0 : green=0 : blue=0
140 print_colours
```

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

The HOME thing

by Wolfgang Lenerz

The latest version of SMSQE has an inbuilt support for a "Home directory thing". This may be of interest to some users, so here is a small description of it.

I – What does it do?

A - Home directory

The HOME thing implements "home directories". A home directory in this context is defined as meaning the directory from which an executable file was executed. Thus, if you have a file called `fred_exe` in a directory `win1_progs_exec_` the home directory for that file will be `win1_progs_exec_`.

The usefulness of this will vary enormously, depending on whether YOU make use of it in your programs.

B - Home Filename

The home thing also supplies what, for want of a better name, I'll call the "home filename" which is the combination of the filename and the home directory, thus making up the complete SMSQ/E filename – in the example above this would be `win1_progs_exec_fred_exe`.

Both home directory and home filename are set up once and for all when the program starts, and are deleted when the program is removed. With one exception, they are immutable: once set, they may not be changed. They are just removed upon removal of the program itself.

C - Current Directory

The Home Thing also implements a "current directory". This is inherited from the job that is setting up the home directory (in most cases the parent job). If the calling job does not have a current directory, a copy of the home directory is used instead.

The current directory can only point to a valid directory. Within that limit it may be set/reset or otherwise manipulated by the job itself.

D - Default Directory for named jobs

Finally, there is also a default home directory for jobs that are executed through other means, perhaps through file managers that don't use the HOME thing, or, especially, through hotkeys. Due to the enormous variations that can exist in situations where jobs are executed from hotkeys, and while there is no problem when jobs are loaded from a file through a hotkey, sometimes the job code is already in memory, but no job with the name exists until the hotkey is actually pressed ("executable things"), sometimes the job executes immediately etc. In those circumstances, it will not always be possible to associate a job with a filename and directory.

It is, however, possible to set up a default home directory for jobs with a given name. When a job, for which a default directory was set up, executes, for example from a hotkey, and tries to get at its home directory, a home dir will be set up for it automatically.

Thus, whenever a job tries to find its home/current directory/file and they can't be found directly because they haven't been defined for that job, a check is also made in the default list. If the job's name is in the default list, then an entry for that job, with that default filename, is made in the home directory list.

II – Can it work for You? – SMSQ/E and QDOS

For the home directory scheme to work, the cooperation of the operating system or file manager(s) is needed: Indeed, whenever a job is executed, whoever is doing this executing must explicitly set up the home directory for the jobs that is being executed. Here, there is a difference between SMSQE and QDOS.

SMSQE now has the HOME Thing built in, and also support for it (see below for QDOS). Typically, on an SMSQ/E system, jobs will be started up through the EX(ec) command variants, through filemanagers such as QPAC II or through FileInfo. Finally, SBasic programs may also be loaded:

A - The EX(ec) etc command

Support for the Home Thing is built into the OS as of SMSQ/E version 3.11. Whenever you use the EX commands, the home directory for the job to be EXecuted will be set up automatically. This support does not exist on QDOS machines, since, obviously, the EXEC command itself had to be modified, which is not possible under QDOS.

B - QPAC II and other file managers

QPAC II has already been altered to take the new home thing into account. All file managers will need to be changed to support the home directory. If you are a programmer and have programmed a file manager, further information is given below, showing you the code that needs to be implemented for this. If a filemanager is in compiled basic (e.g. DiskMate) no further action will be necessary under SMSQ/E since the EX commands in SMSQ/E will do whathever is necessary.

C - FileInfo

Thierry Godefroy has modified FileInfo II to use the Home Thing. For the record, I have modified my own FileInfo (the initial FileInfo) to use the home thing, if present.

D - Basic

Under SMSQE, whenever you (q)load/(q)merge a basic program, the home directory for that basic program is set to the file just loaded. Thus basic is again an exception – it is the only job for which the Home directory may change.

E - QDOS and the HOME Thing

There is a stand alone version of the HOME thing for QDOS users (which can be downloaded at **http://www.scp-paulet-lenerz.com/14mljkl24/wolf/download/**).

Thus, QDOS systems can also profit from the home directories set up from QPAC II and FileInfo, but support for the HOME thing through the EX and LOAD commands will be non-existent, since that requires a change in these commands. The same is true for filemanagers that are compiled basic.

III – How to load the HOME Thing

The HOME thing is already present in SMSQE v. 3.11 onwards and doesn't need to be loaded.

On Qdos, use:

```
a=RESPR (file_length)
LBYTES ‹device›_home_bin,a
CALL a
```

or the LRESPR variants if your system has them.

RWAP Services NEWS!

New Websites!

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

You can find them at:

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

New Products!

Q|W|O|R|D 1.0 NOW WITH DIGITAL SOUND ON QPC2!

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

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

Prices:

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

Notes:
Q-Word DOES NOT require SMSQ/E with GD2 support -OR- SMSQ/E at all on the Aurora or Qx0 machines. It works on the highest colour depth everywhere regardless of Operating System.
The Aurora version is available on either HD or ED disk. For the latter add £1.00 to the price. ED version is uncompressed and can be run directly from the floppy. All other Floppy versions are compressed. QPC/QXL version comes on CD. Non CD versions DO NOW support digital sound on QPC2

Quantum Leap ED Drives

After many years of unavailability, here they are again! These are high quality Mitsubishi constructed/IBM badged drives with full warranty.
Unlike previously sold ED drives, these do not require FLP_JIGGLE and have no problems formatting 720K disks. However for the latter a switch is included with the cable.

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

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

TALENT COMPUTER SYSTEMS for Windows

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

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

Old Favourites!

Utilities

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

Applications

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

Leisure

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

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

RWAP Services
178 Newtown Road, Bedworth, Warwickshire, CV12 8QN, United Kingdom. Tel: +44 2476 490616 (From the UK dial: 02476 490616)
Website: http://www.rwapsoftware.co.uk
We accept:

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

IV – Using the HOME Thing

A - From SBasic

There are several new SBASIC keywords for this.

1 - Get the home directory

```
result$ = HOME_DIR$(job_id)
```

This function returns the home directory for the job given as job_id. To avoid programs stopping with an error if for some unimaginable reason the home directory cannot be found this function returns an empty string if that error happens.
The job ID is optional, in that case -1, meaning the current job, will be assumed.
E.g.:

```
...
100 define procedure init
110    mydir$ = HOME_DIR$
....
```

2 - Get the home filename

```
result$ = HOME_FILE$(job_id)
```

Same as for the home directory, but for the home filename.

3 - Get the current directory

```
result$ = HOME_CURR$(job_id)
```

Same as for the home directory, but for the current directory.

4 - Default names

```
HOME_DEF job_name$, file_name$
```

This sets a default filename for a job with the name given as first parameter. This is useful for "executable things", where no filename is readily available, or for file managers that haven't integrated calls to the home thing. Please refer to the section I-D above for more information on this.

With this keyword, you set up the default job name and filename that is to be used for the home/current file/dir.

Please note that the file_name$ parameter must indeed be a FILENAME, not a directory name.

Example:

```
HOME_DEF "Sbasic", "dev1_sbasic_test_bas"
```

5 - Get the version of the HOME thing

```
result$ = HOME_VER$
```

B - From machine code

The thing can of course also be called from machine code. It implements various extensions - it is an extension thing.

There is some ready to use wrapper code in the SMSQ/E source tree, namely in the file "util_gut_home_asm". If you still want to go the manual route, here is the documentation for the different extensions:

To use them, in short, first you USE the thing (A0 = thing name, D2 = extension). You might want to make sure you use a call that returns the pointer to the thing linkage base in A2 and a pointer to the thing in A1 as these will be expected when calling the call routine. (If you have access to the SMSQE sources, a good vector for this is gu_thjmp).
The name of the thing is, imaginatively, "HOME", and the names of the individual extensions will be given below.

After you have used the thing, you then call the thh_code routine of the thing with A2 pointing to the linkage and A1 to a parameter list.

Each extension thus has its own parameter list. They are explained below for each extension. An example for using the thing from machine code could be as follows, to SET the default name/dir (this uses the gu_thjmp routine from the SMSQE sources, which returns the correct values in A1 and A2).

It is presumed for this example that, on entry, D1 contains the job ID of the job for which the directory/file is to be set, that the job name, for which a default is to be set up can be found at label called 'jobname' and that the filename for this job can be found at a label called 'filename'.

```
homereg     reg     a0-a4/d0-d3
d1stak      equ  4
            ; where D1 is on stack
```

```
setdflt
        movem.l homereg,-(sp)           ; keep my regs
        lea     home_name,a0            ; point to name of thing
        moveq   #-1,d3                  ; wait forever
        moveq   #-1,d1                  ; I will use the thing
        move.l  #'SETD',d2              ; extension in thing to use
        moveq   #sms.uthg,d0            ; use thing
        jsr     gu_thjmp                ; on return A2= ptr thg header, a1 to thg
        tst.l   d0                      ; ok?
        bne.s   no_thg                  ; no, ignore
        move.l  a1,a0                   ; pointer to thing (!!!)
        sub.l   #16,sp                  ; get some space
        move.l  sp,a1                   ; and point to it
        move.l  #$c1000000,(a1)         ; thp.call+thp.str
        lea     jobname,a4              ; point to jobname to set
        move.l  a4,4(a1)                ; set pointer to this string
        move.l  #$c1000000,8(a1)        ; thp.call+thp.str
        lea     filename,a4             ; point to file/dirname to set
        move.l  a4,12(a1)               ; set pointer to this string
        jsr     thh_code(a0)            ; call extn thing
        add.l   #16,sp                  ; reset stack
        lea     home_name,a0            ; now free thing, ignore error on call
        moveq   #sms.fthg,d0
        moveq   #-1,d1
        jsr     gu_thjmp                ; free thing
no_thg
        movem.l (sp)+,homereg           ; ignore error
(...)

home_name
    dc.w        4,'HOME'
```

The extensions are as follows:

GETH
GETF
GETC

Respectively get the home directory, home filename and current directory.

	entry	exit
D0		0 or error
D2		buffer size needed (err.orng)
A1	pointer to parameter list	preserved
A2	pointer to thing linkage	preserved

The parameter list is as follows:

0(a1)	long word	job id of job for which info is to be gotten
4(a1)	word	$A100 (corresponding to thp.str+thp.ret)
6(a1)	word	length of buffer for return string
8(a1)	long word	pointer to buffer for return string

The routine will return 0 if no error occurred, err.itnf if the job with the given ID doesn't have a home directory and err.orng if the buffer is too small for the entire directory/filename. In this latter case, the routine will not touch the given buffer but just return the needed size in D2.

SETD

set the default directory for a given name.

	entry	exit
D0		0 or error
A1	pointer to parameter list	preserved
A2	pointer to thing linkage	preserved

The parameter list is as follows:

0(a1)	long word	$C1000000 (corresponding to thp.str+thp.call)
4(a1)	long word	pointer to string for jobname
8(a1)	long word	$C1000000 (corresponding to thp.str+thp.call)
12(a1)	long word	pointer to string for filename

V – Setting up a HOME Directory

Normally, jobs should not try to set up a home directory for themselves. This should be left to the system/filemanager. When a job is started with the SMSQ/E EX, EW or any of the similar commands, this is done automatically. However, filemanager writers may be interested in this info.

A - From Sbasic

```
HOME_SET job_id, device_and_file_name$
```

Set the home directory, home filename and current directory. You pass the thing the job ID of the job for which this is to be set up and the entire filename, including the device and directory. The thing extracts the home directory from the filename. If you want to set up the home directory for the current job, you may pass -1 as parameter.

Since there can only be one home directory for a job and since that can only be defined once, the keyword will give an error if the home directory is already set for this job. Otherwise, this keyword will set the home directory, the home file and the current directory.
This keyword exists mainly for testing purposes.

B - From Machine Code

Please read the general rules on how to use the thing from machine code (section IV, above). The extension to use here is SETH. to SET the home name/dir (this uses the gu_thjmp routine from the SMSQE sources, which returns the correct values in A1 and A2).

It is presumed here that, on entry D1 contains the job ID of the job for which the directory/file is to be set and that the filename for this job can be found at a label 'filename'.

SETH

This sets the home directory.

	entry	exit
D0		0 or error
A1	pointer to parameter list	preserved
A2	pointer to thing linkage	preserved

The parameter list is as follows:

0(a1)	long word	job id of the job for which this is set
4(a1)	long word	$C1000000 (corresponding to thp.str+thp.call)
8(a1)	long word	pointer to string for entire filename

The routine will return 0 if everything went OK else any error from the memory allocation routine or err.fdiu if the job with this job ID already has a home directory set up.

The following is an example of a routine that can use this to SET the home name/dir (this uses the gu_thjmp routine from the SMSQE sources, which returns the correct values in A1 and A2).

It is presumed here that, on entry D1 contains the job ID of the job for which the directory/file is to be set and that the filename for this job can be found at a label 'filename'.

```
homereg
        reg     a0-a4/d0-d3
        d1stak  equ  4                  ; where D1 is on stack

sethome
        movem.l homereg,-(sp)           ; keep my regs
        lea     home_name,a0            ; point to name of thing
        moveq   #-1,d3                  ; wait forever
        moveq   #-1,d1                  ; I will use the thing
        move.l  #'SETH',d2              ; extension in thing to use
        moveq   #sms.uthg,d0            ; use thing
        jsr     gu_thjmp                ; on return A2= ptr thg header, a1 to thg
        tst.l   d0                      ; ok?
        bne.s   no_thg                  ; no, ignore
        move.l  a1,a0                   ; pointer to thing (!!!)
        move.l  d1stak(sp),d1           ; get job ID back
        sub.l   #12,sp                  ; get some space
        move.l  sp,a1                   ; and point to it
        move.l  d1,(a1)                 ; insert ID of job
        move.l  #$c1000000,4(a1)        ; thp.call+thp.str
        lea     filename,a4             ; point to file/dirname to set
        move.l  a4,8(a1)                ; set pointer to this string
        jsr     thh_code(a0)            ; call extn thing
        add.l   #12,sp                  ; reset stack
        lea     home_name,a0            ; now free thing, ignore error on call
        moveq   #sms.fthg,d0
        moveq   #-1,d1
        jsr     gu_thjmp                ; free thing
no_thg
        movem.l (sp)+,homereg           ; ignore error
(...)

home_name
    dc.w        4,'HOME'
```

Finally, I'd like to point out that, whilst I have divised the home thing initially, Marcel Kilgus later recoded much of the code and also implemented some better ideas of his.

Special QUILL printer drivers

by Dilwyn Jones

Much maligned, Quill still has its uses. Fancy using it to transfer text to a PC? How about generating some HTML code? Or exporting Quill DOCs as simple plain text?

This is a set of five printer_dat files for Quill or Xchange intended to allow Quill to perform 5 special functions:

1. Print to HP Deskjet printers (also works on some Laserjets) ***(HPDJ_printer_dat)***
2. Generate simple plain text files with QL-style linefeed end of lines ***(QLplaintext_printer_dat)***
3. Print to a plain text file with carriage returns added to end of lines for transfer to PCs, and translation of character codes below 128. ***(QLtoPCtext_printer_dat)***
4. Print the text file as a simple HTML file ***(HTML_printer_dat)*** although this driver only works on the Xchange version of Quill (code sequences too long for standard QL Quill and cause an error due to failure to read the printer_dat) and may need to be renamed to xchange_dat.

5. Assume the loaded document is a text file transferred from a PC, so print it as a QL text file with translation of commonest character set differences. *(PCtoQLtext_printer_dat)*

While preparing files to be printed as plain text for example, ensure you set the correct upper and lower margins and the correct header and footer settings. With headers and footers off and top and bottom margins set to 0 and page length set to 255 lines you stand the best chance of generating true plain text files. If you set page length to 0 as in this driver, and leave page length at 66 in the document, what you'll get is every 66 lines two lines joined together, as Quill does not issue end of line characters after the last line in a page.

With all of these drivers, you should print the output from Quill to a file (using the Print commands in Quill) and ensure that the correct printer_dat from the list above is copied to have a filename of printer_dat. Then execute Quill from the same disk and PRINT the file. Note that Quill does not allow you to change printer_dat during a session, so if you wish to use a different driver, you must quit from Quill and start it again.

A short BASIC program select_printerdat_bas is also supplied. This simply asks you to select one of the 5 printer_dats supplied, deletes the existing one and copies the selected one in its place. 4 of the 5 printer_dats are in the supplied install_dat file, the master file for use with install_obj. I could not put the HTML driver in as it contains longer code sequences than allowed by install_obj or Psion Quill. It can only be used with Xchange.

Here is a list of how the printer_dats have been built. These can be entered using Quill's INSTALL_BAS program to create the individual drivers.
Or if you have internet access, you can download ready made files along with a few other useful files from my website:
www.dilwyn.uk6.net/psions/index.html

1. HPDJ_PRINTER_DAT

Assumes a serial port (PAR) is being used. The preamble issues a carriage return to ensure that the first line is printed at the left margin in case use of a previous program left the print position somewhere in the middle of a line then sets the line termination to translate a QL single linefeed to carriage return+linefeed. Postamble is set to a form feed to ensure the last page is ejected from the printer, important in the case of a Laser printer for example. The character translations are a miscellaneous set I happened to find useful. Subscript is actually half height text, while superscript becomes italic, which I found more useful.

```
DRIVER NAME          : HP Deskjet
PRINTER PORT         : SER1
LINES PER PAGE       : 64
CHARACTERS PER LINE  : 80
FORMS TYPE           : Continuous
END OF LINE CODE       : 10
PREAMBLE CODES         : 13,27,38,107,50,71
POSTAMBLE CODES        : 12
BOLD ON CODES          : 27,40,115,51,66
BOLD OFF CODES         : 27,40,115,48,66
UNDERLINE ON CODES     : 27,38,100,49,68
UNDERLINE OFF CODES    : 27,38,100,64
SUBSCRIPT ON CODES     : 27,40,115,54,86
SUBSCRIPT OFF CODES    : 27,40,115,49,50,86
SUPERSCRIPT ON CODES   : 27,40,115,49,83
SUPERSCRIPT OFF CODES  : 27,40,115,48,83
TRANSLATE 1: FROM 96 (') TO 175
TRANSLATE 2: FROM 127 ( ) TO 99,8,79
TRANSLATE 3: NONE
TRANSLATE 4: NONE
TRANSLATE 5: NONE
TRANSLATE 6: NONE
TRANSLATE 7: NONE
TRANSLATE 8: NONE
TRANSLATE 9: NONE
TRANSLATE 10: NONE
```

2. QLPLAINTEXT_PRINTER_dat

Assumes printing to SER1 and QL-style linefeed only at the end of lines.
Generates the simplest of plain text files from Quill DOC files.

```
DRIVER NAME          : QL PLAIN T
PRINTER PORT         : SER1
PARITY               : NONE
BAUDRATE             : 9600
LINES PER PAGE       : 0
CHARACTERS PER LINE  : 80
FORMS TYPE           : Continuous
END OF LINE CODE       : 10
PREAMBLE CODES         : NONE
POSTAMBLE CODES        : NONE
BOLD ON CODES          : NONE
BOLD OFF CODES         : NONE
UNDERLINE ON CODES     : NONE
UNDERLINE OFF CODES    : NONE
SUBSCRIPT ON CODES     : NONE
SUBSCRIPT OFF CODES    : NONE
SUPERSCRIPT ON CODES   : NONE
SUPERSCRIPT OFF CODES  : NONE
TRANSLATE 1: NONE
TRANSLATE 2: NONE
TRANSLATE 3: NONE
TRANSLATE 4: NONE
TRANSLATE 5: NONE
TRANSLATE 6: NONE
TRANSLATE 7: NONE
TRANSLATE 8: NONE
TRANSLATE 9: NONE
TRANSLATE 10: NONE
```

3. QLtoPCtext_printer_dat

Similar to the plain text driver, but issues carriage return and linefeed at the end of lines ready for transfer of text files to a PC. Converts bold text to *bold text* as seems to be the convention with email and other texts. Normally, '_' would indicate italics, but Quill does not support italics, so I have made underlined text be preceded by '_' and followed by '_'. The Pounds is converted to CHR$ 156 to help match the PC character set and Euro symbol is translated to a simple capital E, as not all PC character sets include a Euro symbol. The Copyright symbol is converted to (C), which will alter line lengths slightly.

```
DRIVER NAME          : QLtoPCtext
PRINTER PORT         : SER1
PARITY               : NONE
BAUDRATE             : 9600
LINES PER PAGE       : 0
CHARACTERS PER LINE  : 80
FORMS TYPE           : Continuous
END OF LINE CODE       : 13,10
PREAMBLE CODES         : NONE
POSTAMBLE CODES        : NONE
BOLD ON CODES          : 42
BOLD OFF CODES         : 42
UNDERLINE ON CODES     : 95
UNDERLINE OFF CODES    : 95
SUBSCRIPT ON CODES     : NONE
SUBSCRIPT OFF CODES    : NONE
SUPERSCRIPT ON CODES   : NONE
SUPERSCRIPT OFF CODES  : NONE
TRANSLATE 1: FROM 96 (') TO 156
TRANSLATE 2: FROM 181 (Œ) TO 69
TRANSLATE 3: FROM 127 ( ) TO 40,67,41
TRANSLATE 4: NONE
TRANSLATE 5: NONE
TRANSLATE 6: NONE
TRANSLATE 7: NONE
TRANSLATE 8: NONE
TRANSLATE 9: NONE
TRANSLATE 10: NONE
```

4. HTML_printer_dat

Prints the file as a fairly simple equivalent format HTML page.

```
DRIVER NAME          : HTML
PRINTER PORT         : none
LINES PER PAGE       : 66
CHARACTERS PER LINE  : 80
FORMS TYPE           : Continuous
END OF LINE CODE       : 10
PREAMBLE CODES         : 60,72,84,77,76,62,60,
                         72,69,65,68,62,60,84,
                         73,84,76,69,62,81,117,
                         105,108,108,32,116,111
                         ,32,72,84,77,76,60,47,
                         84,73,84,76,69,62,60,
                         47,72,69,65,68,62,60,
                         66,79,68,89,62,10,60,
                         80,82,69,62
POSTAMBLE CODES        : 60,47,80,82,69,62,60,
                         47,66,79,68,89,62,60,
                         47,72,84,77,76,62
BOLD ON CODES      : 60,83,84,82,79,78,71,62
BOLD OFF CODES     : 60,47,83,84,82,79,78,71,62
UNDERLINE ON CODES     : 60,85,62
UNDERLINE OFF CODES    : 60,47,85,62
SUBSCRIPT ON CODES     : 60,83,85,66,62
SUBSCRIPT OFF CODES    : 60,47,83,85,66,62
SUPERSCRIPT ON CODES   : 60,83,85,80,62
SUPERSCRIPT OFF CODES  : 60,47,83,85,80,62
TRANSLATE 1: NONE
TRANSLATE 2: NONE
TRANSLATE 3: NONE
TRANSLATE 4: NONE
TRANSLATE 5: NONE
TRANSLATE 6: NONE
TRANSLATE 7: NONE
TRANSLATE 8: NONE
TRANSLATE 9: NONE
TRANSLATE 10: NONE
```

5. PCtoQLtext_printer_dat

Assumes that the file is an imported PC text file, so does some character translation (Pound symbol, back tick and copyright symbol) and prints with linefeed only at the end of each line.

```
DRIVER NAME          : PCtoQLtext
PRINTER PORT         : SER1
PARITY               : NONE
BAUDRATE             : 9600
LINES PER PAGE       : 0
CHARACTERS PER LINE  : 80
FORMS TYPE           : Continuous
END OF LINE CODE       : 10
PREAMBLE CODES         : NONE
POSTAMBLE CODES        : NONE
BOLD ON CODES          : 42
BOLD OFF CODES         : 42
UNDERLINE ON CODES     : 95
UNDERLINE OFF CODES    : 95
SUBSCRIPT ON CODES     : NONE
SUBSCRIPT OFF CODES    : NONE
SUPERSCRIPT ON CODES   : NONE
SUPERSCRIPT OFF CODES  : NONE
TRANSLATE 1: FROM 156 (£) TO 96
TRANSLATE 2: FROM 96 (') TO 159
TRANSLATE 3: FROM 184 (õ) TO 127
TRANSLATE 4: NONE
TRANSLATE 5: NONE
TRANSLATE 6: NONE
TRANSLATE 7: NONE
TRANSLATE 8: NONE
TRANSLATE 9: NONE
TRANSLATE 10: NONE
```

Selecting The Required Driver

The little SuperBASIC program on the next page will help you copy the driver for the Quill operation you need. Since Quill only looks once for a printer_dat, copy the driver required first, then start Quill. If you need to change driver, you have to quit from Quill, install the new driver and restart Quill. Note that the program assumes it's running from FLP1_ (see line 110)

```
100 REMark select printer_dat
110 device$ = 'FLP1_'
120 RESTORE
130 DIM names$(4,40)
140 FOR a = 0 TO 4 : READ names$(a)
150 DATA 'HPDJ_printer_dat'
160 DATA 'QLPlainText_printer_dat'
170 DATA 'QLtoPCtext_printer_dat'
180 DATA 'Html_printer_dat'
190 DATA 'PCtoQLtext_printer_dat'
200 :
210 CLS
220 FOR a = 0 TO 4 : PRINT a+1;' - ';names$(a)
230 INPUT\'Select which printer_dat file:';ip%
240 IF ip% = 0 THEN STOP
250 DELETE device$&'printer_dat'
260 COPY device$&names$(ip%-1) TO device$&'printer_dat'
```

Helpline

Robert Hartung writes:

My thanks to all who made the current Vol. 10 Issue 3 one of the best in years! I especially appreciated Dilwyn Jones' Insider CD containing QL/QDOS/SMSQ documentation from day one to the present. I missed out on buying the printed Reference work on this while it was available, so now I have those original documentations plus updates in one source. Even though I haven't kept up with buying all the latest and greatest HW and SW as it became available, as a Sinclair aficionado and sometimes programmer since 1980 I still enjoy reading about it all.

Having said that, I am also much interested in QPCPrint in case my last two remaining parallel port ESC/P2 printers go down. I downloaded the demo and manual for it, but after repeated attempts, somehow my 81-year + brain doesn't seem to grasp how to set it up in my QPC2 v.3.02 so I can output to my USB Epson CX4600 printer. I mostly want to print out files from Quill and Abacus as they are set up presently to print with Epson 510 and Canon BJ 210 configuration.

Is there someone out there who would be so kind as to take me by the hand and, taking nothing for granted, lead me step by step through the necessary procedure? If I seem to be the only one who needs this, I would greatly appreciate an email to my address below. If there seem to be others, no doubt a short article in QL TODAY would be much appreciated by all.

Editor's reply:

Thank you very much for the compliments. Issue 3 was an issue which was a pleasure to produce - no rush, no chase for articles as enough material was waiting for us.

We tried to find out if there was any demand for QPC Print trouble shooting. QPC Print seems to be a trouble-free product. Easy to install, easy to set up - just a few easy steps. We tried to think of problematic areas. Here is what we came up with:

1. Check that your printer is working fine under Windows - otherwise QPC Print obviously cannot work.
2. Ensure that you have installed QPC Print into the same directory in which you have installed QPC.
3. In the QPC SER/PAR dialogue, the "filter" box next to the selected printer needs to be ticked (if you forget it, the printer will only print garbage or nothing, as QPC Print does not come into play without this box.

Well, and that's about it. As you seem to be printing to an EPSON compatible printer or an EPSON itself, no need to change anything inside QPC. You can type a BASIC line to see if the printer prints something, like

```
JOBS \PAR1
```

to print to the printer defined to be PAR1.

Directory Names

by David Denham

This is a small set of superbasic functions to handle filenames.
When writing programs to handle files on a system which supports directories, it is hard to resolve the directory and filename parts. Since Qdos doesn't really make a distinction between directories and filenames, it's hard to write code which copes well with moving files between directories and so on.
So I wrote these functions to help with this task.
I suppose that these functions could be seen as a kind of filename parser.
Supposing we have a program called `MYPROG_EXE` which has been saved into a directory called PROGS on drive win1_. The full filename would be:

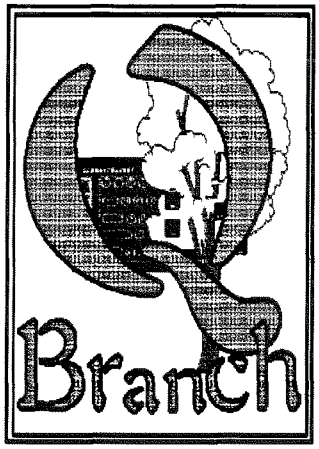
`win1_PROGS_MYPROG_EXE`

These functions will help to break down (or parse) the filename given to help us handle and manipulate files. Where the user explicitly gives filenames, this is not usually necessary, but suppose we need to write a program which looks at all the files in a directory and copies or moves these files to another directory. We need code which can break down the filename into all its various elements and handle and adjust filenames as a result.

A list of the functions to extract drive name, directory name, pure filename and extension name individually:

```
10000 DEFine FuNction Directory_Name$ (filename$)
10010   LOCal ch,d$
10020   ch = FOP_DIR(filename$)
10030   IF ch < 0 THEN
10040     d$ = ''
10050   ELSE
10060     d$ = FNAME$(#ch)
10070     IF d$ > '' THEN
10080       IF d$(LEN(d$)) <> '_' THEN d$ = d$&'_'
10090     END IF
10100   END IF
10110   RETurn d$
10120 END DEFine Directory_Name$
10130 :
10140 DEFine FuNction Drive_Name$ (filename$)
10150   LOCal d$,ch
10160   d$ = ''
10170   ch = '_' INSTR filename$
10180   IF ch > 0 THEN d$ = filename$(1 TO ch)
10190   RETurn d$
10200 END DEFine Drive_Name$
10210 :
10220 DEFine FuNction Pure_Filename$ (filename$)
10230   LOCal ch,d$
10240   ch =  LEN(Drive_Name$(filename$))+LEN(Directory_Name$(filename$))
10250   d$ = ''
10260   IF LEN(filename$) > ch THEN d$ = filename$(ch+1 TO LEN(filename$))
10270   RETurn d$
10280 END DEFine Pure_Filename$
10290 :
10300 DEFine FuNction Extension$ (filename$, separator$, maxlen)
10310   LOCal ch,c,ext$,f$
10320   REMark maxlen is the maximum length of extension permitted
10330   REMark usually '_' or '.' plus three characters
10340   REMark separator$ is '_' (QDOS) or '.' (PC)
10350   f$ = Pure_Filename$(filename$) : REMark remove drive+directory
10360   ext$ = '' : REMark extension string
10370   ch = LEN(f$)-maxlen+1
```

Q Branch



Text 87

£ 79.00

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

Text 87 High Colour Patch Available Now

£9.00

PROGRAMMING

QD 2003	£ 49.00
QD + QBasic	£ 63.00
QD + Qliberator + QBasic	£ 104.00
Qliberator	£ 50.00
Master Spy v 3.3	£ 30.00
QPTR	£ 32.00
Easyptr pts 1 & 2 v4	£ 41.50
Easyptr pt 3 (C library)	£ 14.00
QMake	£ 18.00
QMon / JMon	£ 22.00
Basic Linker	£ 22.00
DISA 3	£ 34.00
QMenu	£ 16.00

HERE'S WHAT'S COMING

v1.02

£ 41.50

+ P&P
£2.00 (UK) £2.50 (EU) £3.00 (ROR)
Requires SMSQ/E to run

QDT is now version 1.02 and updates can be obtained from Q Branch.
QPC2 has also been updated to v 3.12
SMSQ/E is now v 3.32 with many new enhancements.
We are now supplying QPC2 on a CD ROM disk with the manual in PDF format.
QPAC2 is now version 1.44 and is compliant with the new HOME directory commands.
Spring is coming and there are more new things coming along with it.

EasyPTR v4

High colour support, Major re-write!
Full Program Pts 1&2

£ 41.50

Upgrade from any Version £27.50

Needs SMSQ/E v 3.10

QL Keyboard Membranes

Available Now !

£ 17.50 ea + P&P

QPC 2 v3.32

Now Only £ 42.00

now on CD ROM

SMSQ/E

Gold Card / Atari / QXL Version
Aurora High Colour Included

£ 27.00

Version 3.12 out now!
Aurora colour upgrade
£ 13.00

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

Print to any printer from QPC2

QPCPrint £ 27.50

```
10380   IF ch < 1 THEN ch = 1
10390   FOR c = LEN(f$) TO ch STEP -1
10400     IF f$(c) = separator$ THEN
10410       ext$ = f$(c TO LEN(f$))
10420       EXIT c
10430     END IF
10440   END FOR c
10450   RETurn ext$
10460 END DEFine Extension$
```

The four functions are as follows:

DRIVE_NAME$ returns the drive name part of a filename (defined as the part up to and including the first '_' character in the filename).

DIRECTORY_NAME$ uses the technique of opening a directory channel with FOP_DIR and using the Toolkit 2 FNAME$ function to return the name of the directory part of the filename. FNAME$ does not return the '_' character, but our function does, as it usually makes it easier.

PURE_FILENAME$ strips off the directory and drive name, returning just the filename part. In our example above, WIN1_PROGS_MYPROG_EXE has a pure filename of MYPROG_EXE.

EXTENSION$ returns the filename extension, being defined as the part of the filename after the last separator.
For example, WIN1_PROGS_MYPROG_EXE has an extension of '_EXE'. This function lets us specify whether extension separator characters are QDOS-style '_' characters or PC-style '.' characters. We can also specify how long extensions should be, to reduce the risk of confusion in programs which have underscores as part of their filenames, such as a program called my_first_program store in win1_progs_. In this case, that program does not have a filename extension as such, QDOS has no real knowledge of filename extensions as such.

Examples:
PRINT Pure_Filename$('win1_progs_myprog_exe')
prints myprog_exe
PRINT Drive_Name$('win1_progs_myprog_exe')
prints win1_
PRINT Directory_Name$('win1_progs_myprog_exe')
prints progs
PRINT Extension$('win1_progs_myprog_exe','_',4)
prints _exe

Here is a short test program showing you how the routines should be used:

```
100 CLS : CLS #0
110 INPUT #0,'Full filename ';full$
120 PRINT 'Full filename  = ';full$
130 PRINT 'Drive name     = ';
    Drive_Name$(full$)
140 PRINT 'Directory name = ';
    Directory_Name$(full$)
150 PRINT 'Pure filename  = ';
    Pure_Filename$(full$)
160 PRINT 'Extension name = ';
    Extension$(full$,'_',4)
170 STOP
```

Now, let us progress to a short program to show the use of this little bit of code. Below is a simple copier program, able to copy files from one directory to another.

```
100 CLS : CLS #0
110 INPUT #0,'Copy from ';from$
120 INPUT #0,'Copy to   ';dest$
130 OPEN_NEW #3,ram1_temp
140 DIR #3,from$
150 CLOSE #3
160 drive$ = Drive_Name$(from$)
170 OPEN_IN #3,ram1_temp
180 INPUT #3,t$ : REMark skip medium name
190 INPUT #3,t$ : REMark medium capacity
200 REPeat copying
210   IF EOF(#3) = 1 THEN EXIT copying
220   INPUT #3,f$
230   PRINT'Copying ';drive$&f$;' to ';
      dest$&Pure_Filename$(drive$&f$)
240   COPY drive$&f$ TO dest$&
      Pure_Filename$(drive$&f$)
250 END REPeat copying
260 CLOSE #3
270 DELETE ram1_temp
280 STOP
290 :
```

There are probably simpler ways of writing such a program, but it should serve to illustrate how to use the routines. The locations where files are copied from and copied to are entered in lines 180 and 190 (ensure they end with a '_' character of add lines just after them to check if the string ends with '_' and if not add the character).
Lines 130 to 150 sends a list of files to a temporary file called 'ram1_temp'These lines could really be shortened to just 130 DIR \ ram1_temp, from$ on systems where DIR allows this form.
Line 160 extracts the drive name, in case the 'from' entry includes a directory name.
Lines 180 and 190 ensure that the first two lines of a DIR listing are skipped - these contain the medium name and capacity.
The loop then reads back all filenames from the list.

Line 240 sorts out the names and does the copying.
Let us suppose that the filenames are to be copied from WIN1_PROGS_ to WIN1_PROGRAMS_.

```
from$ = 'WIN1_PROGS_'
dest$ = 'WIN1_PROGRAMS_'
```

The listing output by DIR includes the directory names but not the drive name.

```
240    COPY drive$&f$ TO dest$&
       Pure_Filename$(drive$&f$)
```

drive$ was extracted in line 160 so that we have just the root drive name to go with the filenames from the DIR listing.
We combine the drive name and the filename from the DIR listing (the part of line 240 before the TO) to copy from, then we use the Pure_Filename$ function to strip the source drive and directory name to get the simple filename, which is then added to the destination drive and directory to work out where to copy to.
Somewhat mind boggling until you get used to the concepts involved!
This program could be extended to only copy files with given extensions by adding lines such as:

```
125 INPUT #0,'Copy files with extension
    ';extn$
225 IF Extension$(f$,'_',LEN(extn$)) =
    extn$ THEN
245 END IF
```

These routines are ones you never realised you needed until you've started using them!

Sound Advice

by Per Witte

Thanks to the hard work done by a number of people, its never been so easy to add decent sound to your programs as now. If you have a Qx0 or QPC2 you can use the SMSQ/E Sample Sound System (SSSS – shortened here to SSS, to encompass Qdos Classic) to play sound samples and even real music. This can make a big difference to the fun and usability of your programs.

Here I present a few simple routines that you can merge into your own programs directly, or just take a look at, to get an idea of how easy it can be to produce your own. These routines take into account that you may want your programs also to be able to run on platforms that don't have the supporting hardware, in which case they fall back on the familiar BEEP.

All you need in addition to the platforms mentioned (QPC2 must be V3.20 or later) are Wolfgang Lenerz'es Soundfile toolkit, which you can download from the Internet (contact details below) and some sound samples.
For distributable programs I prefer the Soundfile toolkit as it can be compiled in with the code, whether you use Turbo or Qlib. Simon Goodwin's sound device is more versatile, but needs to be loaded residentially at boot time or at least before starting your program.

Sound samples are a different matter. Until someone creates an online sound sample library, you will have to pinch them from other programs (for example from the game D-Miner, available off my website) or record and/or convert your own. (How? Download Jonathan Hudson's Qdos port of SOX to convert standard .wav files to the ub format, or search for and download a Windoze compatible version of SOX on the Internet and use that.)

Ok, so lets get our hands dirty. My comments normally describe the code block immediately above, unless otherwise stated:

```
1 REMark $$asmb=win1_snd_sound_bin,0,10
2 :
```

If you want to compile the program with Qlib and include the soundfile toolkit with the code, you need this line. The corresponding Turbo directive is:

```
1 REMark %%win1_snd_sound_bin,0,10
2 :
```

You don't need to compile a separate version for non-SSS platforms as, provided you don't call any of the procedures in the toolkit, it won't error. The routines shown here take care of that for you.

```
3 rem Test for presence of Sampled Sound
  System
4 TstSSS
5 :
```

One of the first things to do is to determine whether the SSS is present and set a switch accordingly. I use two global program switches relating to sound:

1) ***ssspresent*** can be true of false depending on whether the SSS was detected or not. This switch is set inside the TstSSS procedure.

2) ***sound*** is used by the sound subroutines to determine whether the user wants sound or not.

```
6 sx$ = '_ub'       : rem Sound file extension
7 sd$ = 'ram1_snd_': rem Sound file folder
8 :
```

Initialise some variables used in the sound routines. Better have them up front where they can be easily seen and altered, rather than buried somewhere deep inside your program.

```
9 game = 0: win = 1: rem Some dummy game
  variables
10 randomise date
11 :
```

I've devised a very silly "game" to help illustrate usage.

```
12 rem User input or config file determines
   sound on/off
13 snd = YesNo('Do you want sound')
14 SetSound snd: rem Switch sound on/off
15 print 'Ready to play'
16 Ping 'ok': pause 50: rem allow time to
   display sound and text
17 :
```

Its always a good idea to give the user the choice of having sound or not. ***SetSound*** sets the global program ***sound*** switch to on or off. If sound is wanted, it also sets the switch to the best available sound.
Finally ***Ping*** gives a confirming noise if sound is on (see below)

```
18 rem Your program goes here. Eg..
19 :
20 cls
21 rep main
22  PlayGame
23  if game = win then
24   print 'You win :o)'
25   Ping 'win'
26  else
27   print 'You loose :-('
28   Burp 'loose'
29  endif
30  if not YesNo("Another game"): exit main
31 endrep main
32 Ping 'bye'
33 :
```

The main program loop simulates a game. If you "win" it plays the "win_ub" sound sample, otherwise it plays the "loose_ub" sound sample. That's really all there is to it! The rest is just detail!

Ping and Burp take a filename parameter which means you can re-use them to play different sound samples, but they fall back on only two kinds of beep: 'Nice beep' and 'Bad beep'. But this is a simplification for the purposes of this demonstration. If you know how to be creative with BEEP you could do things differently and very much better. The point is that you can follow any action with a sound, as shown. Every click on a button could produce a different sample sound and fall back on a different beep if you wish.

Note: When storing sound samples in a library it is a good idea to group them and give them evocative names such as 'quack' or 'gunshot1'. However, when you use them in a program it is better to rename copies of them according to their *function* in the program. This makes the program logic clearer and also allows the user to replace the default sounds with his own selection.

```
34 defproc PlayGame
35 rem Dummy game routine
36 game = rnd(0 to 1)
37 enddef PlayGame
38 :
```

Well, that was your free game ;o)

```
39 deffn YesNo(txt$)
40 print txt$; ' ‹Y/n›? ';
41 if inkey$(-1) instr 'y' & chr$(10): print
   'Y': ret 1
42 print 'N': ret 0
43 enddef YesNo
44 :
```

Here follow the sound routines:

```
45 rem Sound routines
46 :
47 defproc SetSound(s)
48 if s then
49  if ssspresent: sound = 2: else: sound = 1
50 else
51  sound = 0
52 endif
53 enddef SetSound
54 :
```

SetSound 0 turns the program sound off, while SetSound 1 turns sound on. It also determines the level of sound available to the program; either BEEP or SSS. All it is is a logical switch in a wrapper for convenience, as the option of changing the sound status should be available to the user at any time.

```
55 defproc Ping(nm$)
56 rem Nice sound
57 if sound = 2 then
58  if ftest(sd$ & nm$ & sx$) = 0 then
59   killsound: soundfile sd$ & nm$ & sx$
60  else
61   beep 2, 2
62  endif
63 else
64  if sound: beep 2, 2
65 endif
66 enddef Ping
67 :
68 defproc Burp(nm$)
69 rem Bad sound
70 if sound = 2 then
71  if ftest(sd$ & nm$ & sx$) = 0 then
72   killsound: soundfile sd$ & nm$ & sx$
73  else
74   beep 999, 255
75  endif
76 else
77  if sound: beep 999, 255
78 endif
79 enddef Burp
80 :
```

The logic of Ping and Burp is:

If sound is off: do nothing
If sound is on then
 If SSS available then
  If the file exists: play sample sound
In all other cases just beep

```
81 defproc TstSSS
82 loc adr
83 rem GLOBal ssspresent
84 rem Test for presence of Sampled Sound System
85 rem V0.01 pjwitte 2oo5
86 :
87 adr = alchp(26)
88 poke_l adr + 00, hex("43fa0016"):
   rem lea.l result,a1
89 poke_l adr + 04, hex("26780070"):
   rem move.l exv_i4,a3
90 poke_l adr + 08, hex("0cab5353"):
   rem cmp.l #sss.flag,-8(a3)
91 poke_l adr + 12, hex("5353fff8")
92 poke_l adr + 16, hex("57e90001"):
   rem seq 1(a1)
93 poke_w adr + 20, hex("7000")     :
   rem moveq#0,d0
94 poke_w adr + 22, hex("4e75")     :
   rem rts
95 poke_w adr + 24, 0               : rem
   ds.w 1
96 call adr
97 ssspresent = peek_w(adr + 24)
98 rechp adr
99 enddef TstSSS
100 :
```

TstSSS is the only little awkwardness that remains, and hopefully it will one day be redundant: There is no easy and documented way offered by the OS of determining whether the Sampled Sound System is available or not, so we have to resort to a bit of jiggery-pokery. All this routine does is check whether it can find the SSS flag at the documented location just above the interrupt server, and set the *ssspresent* switch accordingly.

Have fun!

D-Miner can be found on my website, at Knoware:
http://knoware.mysite.freeserve.com/index.html

The soundfile toolkit can be found at Wolfgan Lenerz'es website:
http://www.scp-paulet-lenerz.com/14mljkl24/wolf/download/

The Qdos version of Sox can be found at Jonathan Hudson's website,
Daria: http://www.daria.co.uk/

Simon Goodwin's sound device and a collection of sound tools - including samples - can be found at Dilwyn Jones'es massive site:
http://www.dilwyn.uk6.net/index.html

Editor's comment:
We will put the sound files and the listing on one of the next cover disks, assuming it is not a "motto" disk. Or maybe we can even produce a "Sound cover disk" - who knows? If you have contributions for it, please send them to us and we will see what we can do!
To make it easier for you to obtain the sound files, they will also be available for download on the website QLTODAY.com *- thank you very much, Bruce!*

Internet on the QL ? – Part III
(Shoestring Linux)

by Phoebus R. Dokos B.Sc MIS

5. Shoestring Linux Access (continued from Vol. 10 Iss.1 p.48)

5c. Installation

In the last part we were left staring at the "`Command (m for help):`" prompt of `atari-fdisk`. In the meantime, Derek Stewart's article in the previous QL Today (Vol.10.Iss.2.p.34) explained some things, much to my happiness as I can now reduce the size of the article! (Slacker! – Ed.).
Assuming you already have pressed "`p`" your screen should look something like figure 1.

```
bash#atari-fdisk
Using /dev/hda as the default device.
Checking that no-one is using this disk right now... hda: AHDI hda1 hda2 hda3 XGM< hda4 hda5 hda6 >
OK
Using /dev/hda as default device

Command (m for help): p

Disk /dev/hda: 12594960 * 512 bytes, 6297480 kBytes, 6149+ Mbytes
   Device  Boot Val Rootsec.     Begin       End    Blocks  Id   System
/dev/hda1        *         0         1   1048576   524288  QWA  SMSQ native
/dev/hda2        *         0   1048577   2097152   524288  QWA  SMSQ native
/dev/hda3        *         0   2097153   3145728   524288  QWA  SMSQ native
/dev/hda4   LNX  *   3145729   3145730  11132928  3993599+ LNX  Linux native
/dev/hda5        *  11337729  11337730  12594959   628615  SWP  Linux swap

Command (m for help): n
```

Figure 1 - atari-fdisk partition list

You see in that screen that I already have Linux partitions. Assume that there is none for your purposes and give "`n`" and ENTER to the prompt. You will be presented with information on the free space on the disk as well as a prompt for the first sector of the new partition. Press ENTER there as it assumes the first available.
Then either give the last sector of the partition, the size in Megabytes or Kilobytes (a number preceded by an addition sign "`+`" and followed by a capital K or a capital M ie `+100M` -or- `+102400K`) or by pressing ENTER the default size given by `atari-fdisk` (figure 2). In my case the newest partition was number 5 (if you examine the previous partitions 5 and 4 from figure 1 you will see a gap in the sector numbers. That gap corresponds to empty space which is what partition 5 filled up.

```
Command (m for help): n
Free areas on disk are:
    Begin       End     Blocks      MB
 11132929  11337728     102400    100
First sector (11132929-[11132929]-11337728):
Using default value 11132929
Last sector or +size[MK] ([11132929]-11337728, dflt 11337728): +100M
Using default value 11337728
New partition will be number 5

Command (m for help): p
```

Figure 2 - New partition creation

If you now type `p` again your screen will look something like figure 3.

```
Command (m for help): p

Disk /dev/hda: 12594960 * 512 bytes, 6297480 kBytes, 6149+ Mbytes
   Device  Boot Val Rootsec.     Begin       End    Blocks  Id   System
/dev/hda1        *         0         1   1048576   524288  QWA  SMSQ native
/dev/hda2        *         0   1048577   2097152   524288  QWA  SMSQ native
/dev/hda3        *         0   2097153   3145728   524288  QWA  SMSQ native
/dev/hda4   LNX  *   3145729   3145730  11132928  3993599+ LNX  Linux native
/dev/hda5        -  11132929  11132930  11337728   102399+ LNX  Linux native
/dev/hda6        *  11337729  11337730  12594959   628615  SWP  Linux swap

Command (m for help):
```

Figure 3 - changed partition list

Since you are making a brand new installation, you will obviously have only SMSQ/e partitions (denoted by the type QWA), so you will need to make one or more Linux partitions – Linux doesn't mind too much about the size; but make sure you make a smaller one that will be used as a SWAP. This is where Linux "spills out" memory contents that are not being used if the physical memory gets full. The larger that is, the more programs Linux can load without crashing. Make sure this is set to at least 128 Mb. Mind you though that in small footprint systems (like the Q40 -not the Q40i) where memory is not abundant, the more applications you allow Linux to run, the slower the machine will be response-wise. To create the swap partition, create the partition as previously, then on the prompt, type t to change its type from the default LNX and enter SWP (for Swap). In any case, when you are done, enter w and press ENTER to exit and save your changes.

TF Services

Compswitch

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

Cost £24

superHermes

A major hardware upgrade for the QL

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

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

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

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

QL REPAIRS (UK only)

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

£27 incl 6 month guarantee

Minerva

The ORIGINAL system operating system upgrade

OTHER FEATURES COMMON TO ALL VERSIONS

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

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

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

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

QL RomDisq

Up to 8 mbyte of flash memory for the QL

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

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

MPLANE

A low profile powered backplane with ROM port

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

Cost .....**£34** (£35/£36)

I2C INTERFACES

Connects to Minerva MKII and any Philips I²C bus

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

QL SPARES

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

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

22 Feb 03

29 Longfield Road, TRING, Herts, HP23 4DG
Tel: +44 (0) 1442-828254 Fax/BBS: +44 (0) 1442-828255
tony@firshman.co.uk http://www.firshman.co.uk

Then type shutdown -r now and press ENTER. At the prompt, just press ENTER and wait until Linux signals you that it is time to reset.

Next, insert your Linux CD and start Linux by giving:

```
EX win1_lxx; '-k win1_vmlinux -m 32 -- root=/dev/hdX parport=0x278,0x378'
```

Where `-m 32` could be `-m 64` etc. depending on your memory size and `/dev/hdX` is the drive descriptor corresponding to your CD (ie in my machine it is `/dev/hdb`). As previously mentioned the parport option should be set to only one or two ports depending on your Multi I/O card.

Once, the Penguin shows and Linux rants about the machine etc. you will be prompted to login. Login as root and then type `xinstall` and press ENTER.
The installation program will appear and you should follow the simple but easy to use install program. Once it gets to the packages installation, get up from your desk, make a coffee (although a four course meal is more accurate!) as it will take a long time (1 hour + on a Q60 / 3+ hours on a Q40) before that step is finished.
After the packages have been installed you will be presented with the final screen where you can make configuration changes. Make SURE you enable the network, otherwise the editing involved will make the bravest, shy!
Once you exit and shutdown Linux, restart your QxO and then start Linux again, with the same command as before but this time substituting X for your Hard and partition that Linux was installed on. In my case this is:

```
EX win1_lxx; '-k win1_vmlinux -m 32 -- root=/dev/hda4 parport=0x278,0x378'
```

Once Linux has started, log in as root and get ready for the configuration.

5d. Setup for a LAN

1. Introduction

There are two main types of LAN connections; Static and Dynamic, each of which can be setup in two ways: Permanently and Per Session: As we want to simplify everything, we will look into the per session way as although it has to be repeated, every time that Linux boots, it is a lot easier to set up. Additionally, you wouldn't want everything on a silver platter now would you? (YES WE WOULD - Ed.). Okay, okay! You convinced me! Without further ado let's go to:

2. Ethernet Card Setup

Before we continue, we will need to make sure that your Ethernet card is properly set up and that Linux recognises it. If your machine was setup by D&D that should be no problem - The card should be at 0x300, IRQ: 5- however if you got one off of some bargain bin, computer store or, even, eBay, chances are that it is setup for I/O address 0x300, IRQ: 3 which is the NE2000 standard. Now, normally there's nothing wrong with IRQ 3 as it is a shareable interrupt (shareable with Ser2 or COM2), however, I have found many cases that that won't work under SMSQ/e properly for reasons that I didn't even bother to investigate (we know you are lazy - Ed.). It is best, therefore to set it up for IRQ 5 and that should create NO problems whatsoever. In order to set it up, you will need to modify a configuration file, however Linux/Unix text editors tend to have a myriad of not-so-user friendly key combinations.
Now, thanks to the foresight of Richard Zidlicky, there's an easier solution: Midnight Commander
Midnight Commander is a clone of Norton Commander on DOS. For DOS novices or people who have never used a PC, your QL equivalent is Dilwyn Jones' Q-Trans, which is also a quasi clone of Norton Commander. Midnight Commander (figure 4), provides not only for fast navigation in the typical vast Unix tree, but also a convenient file editor and viewer among other things.

To start Midnight Commander, type mc and press ENTER.

You navigate using the cursor keys, switch panels with TAB, select/deselect with INS and perform actions with ENTER (to change directories or launch a file). To perform all other actions, use the function keys listed on the bottom (The F is ommited). F9 calls the pull down menu.

The first directory that you will see listed is the root user's home directory (no need to elaborate on that as this isn't a Linux/Unix tutorial). Find the entry '`..`' (Two periods signify up one directory in the structure), press ENTER and you will be in the root directory. Next find the `/etc` directory entry, move the highlighted bar above it and press ENTER.
Next, locate the modules.conf.q40 file. Highlight it and press F4. That will take you to the editor.
Locate the entry reading:

```
options ne io=0x300 irq=3
```

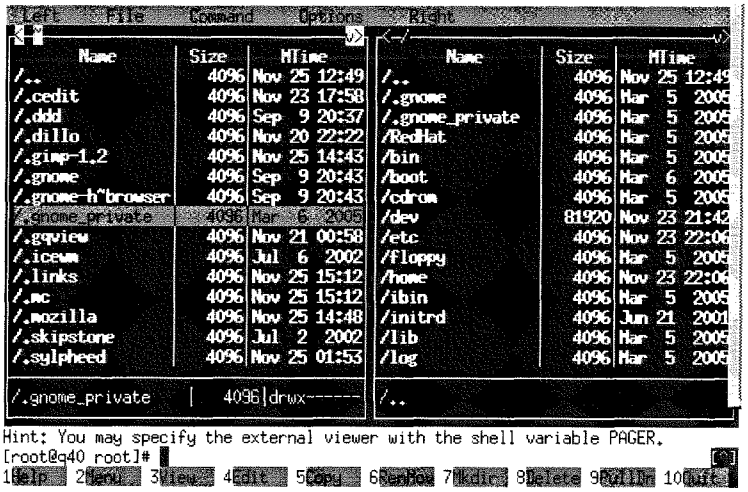


Figure 4 - Midnight commander

If the line reads instead:

```
options ne io=0x300 irq=5
```

then you are fine and you can quit both the editor and Midnight Commander and proceed to the next step, otherwise change the `irq=3` to `irq=5` (provided that that is the IRQ you set your card to). Press F10 and select `Yes` when asked. Quit Midnight Commander with F10.
Although it is possible to make Linux to read the change in configuration immediately, we will go the easiest route and just reboot the machine (What only Windows should do that?). Reboot by pressing CTRL-ALT-DEL (That effectively issues a `shutdown -r now` command, thanks to Richard and shuts Linux down cleanly).
Restart Linux as before and you are ready to...

3. Setup for DCHP

DHCP stands for "Dynamic Host Configuration Protocol", a protocol for assigning dynamic IP addresses to devices on a network. With dynamic addressing, a device can have a different IP address every time it connects to the network. DHCP also supports a mix of static and dynamic IP addresses. In short, you do not have to either assign an IP address -or- to manually change it if the case arises. The network software, probes the network for DHCP broadcasts and responds to them. In turn a DHCP server will assign an IP address from the "pool" of available addresses to it.

To setup DHCP do the following:

a. Shutdown the network by issuing `/etc/rc.d/init.d/network stop`
b. Using Midnight Commander (see above), edit the file
`/etc/sysconfig/network-scripts/ifcfg-eth0` to be as follows:

```
DEVICE=eth0
USERCTL=no
ONBOOT=yes
BOOTPROTO=dhcp
BROADCAST=
NETWORK=
NETMASK=
IPADDR=
```

c. Save this and next edit the file `/etc/sysconfig/network` to use DHCP.
This should read:

```
NETWORKING=yes
HOSTNAME=
FORWARD_IPV4=yes
GATEWAYDEV=
GATEWAY=
```

d. Restart the network to probe the DHCP server for your network settings with the command `/etc/rc.d/init.d/network restart`

e. Verify your network settings with the commands `/sbin/ifconfig` (figure 5) to make sure you have received an IP address from the DHCP server and `/sbin/route` (figure 6) to see if your gateway is properly setup.

```
[root@q40 /]# /sbin/ifconfig
eth0      Link encap:Ethernet  HWaddr 00:00:B4:33:30:54
          inet addr:192.168.254.115  Bcast:255.255.255.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:126 errors:0 dropped:0 overruns:0 frame:0
          TX packets:112 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:58995 (57.6 Kb)  TX bytes:11755 (11.4 Kb)
          Interrupt:5 Base address:0x300

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

[root@q40 /]#
```

Figure 5 (left) - ifconfig output

```
[root@q40 /]# /sbin/route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.254.0   *               255.255.255.0   U     0      0        0 eth0
127.0.0.0       *               255.0.0.0       U     0      0        0 lo
default         192.168.254.254 0.0.0.0         UG    0      0        0 eth0
[root@q40 /]#
```

Figure 6 (above) - route output

f. Ping the gateway and a few other computers on the network to verify your connection by issuing `ping 192.168.254.254` (replace the IP address with your gateway's IP address). Note that pinging will not stop until Ctrl-C is pressed.

4. Setup for a Static Address

A static address stays the same all the time. This is especially good if you want for example to run a Web Server on your Q40 and you want to redirect HTTP requests to the Q40 and say FTP requests for your Amiga (See I don't even mention Windows!). That is a lot easier to perform if the machine is on a fixed address. The downside to that is that you will have to provide Linux with more information than just the address; you will need to also provide the Netmask and Gateway for your network. This is usually either your router or a more advanced ADSL or Cable modem. Q40 Linux is perfectly capable of acting as one as well, however you will need at least two network cards (which also means that you will need some kind of ISA extender to be able to plug more cards). So we skip this information instead and go directly to the configuration.
There are two ways to configure for a static address:

a. By session

To configure by session do the following:

- Set the IP address and network mask: `/sbin/ifconfig -a eth0 192.168.254.233 netmask 255.255.255.0`
  (This example gives the machine the IP address 192.168.254.233, but you can use any combination of IP/netmask that will work with your network. This information will be available usually from your router's manual)
- Verify the settings with `/sbin/ifconfig eth0`
- Add the default gateway (in my case): `/sbin/route add default gw 192.168.254.254` (Replace 192.168.254.254 with your gateway address)
- Verify the gateway setting by issuing the following command: `/sbin/route`. The line beginning with `default` should have your gateway under the gateway column.

b. Permanently

To configure it once and for all do the following:

- Edit the file `/etc/sysconfig/network-scripts/ifcfg-eth0` to look like the following (replace with your network numbers).

```
DEVICE=eth0
USERCTL=no
ONBOOT=yes
BOOTPROTO=none
BROADCAST=192.168.254.254
NETWORK=192.168.254.0
NETMASK=255.255.255.0
IPADDR=192.168.254.253
```

- and the file `/etc/sysconfig/network` to look like the following (replace with your network numbers and hostname)

```
NETWORKING=yes
HOSTNAME=q40.home.dokos-gr.net
FORWARD_IPV4=yes
GATEWAYDEV=
GATEWAY=192.168.254.254
```

c. To verify both types of configuration, ping your gateway and a few other computers on the network to verify your settings are correct by issuing `ping 192.168.254.254` (replace the IP address with your gateway's IP address). Note that pinging will not stop until Ctrl-C is pressed.

5e. Setup for Cable/DSL

1. Introduction

Cable/DSL connections should be generally considered LAN connections as mentioned above, however there are some differences as not all Cable and DSL modems act as straight routers/ bridges. Some of them utilise a protocol known as PPPoE (Point-To-Point-Protocol over Ethernet), which essentially is the same as your run-of-the-mill dial up connection but utilising Ethernet as the transmission channel.

2. Regular connections

If you have a "regular" Cable or DSL modem and assuming that you have no separate Broadband router, then you should see 5d.3 above and particularly the DHCP setup, as that is the protocol that Cable and DSL modems utilise to connect you to the Internet. Once DHCP is setup, then the modem through its built-in DHCP server will setup your machine and you will be ready to go. If you on the other hand you have a broadband router/switch, then you should read its documentation and set up your connection either as DHCP -or- Static according to the router manufacturer's instructions.

3. PPPoE connections

PPPoE is trickier as it requires a special program run on the machine akin to a dial-up dialer (as we will see soon), however, most of modern broadband routers and modems, do provide their own inbuilt one. In such a case you do not have to run a program on your Qx0 but you WILL have to setup your modem.

Below, we will examine all cases (although not ALL possibilities as they depend on each manufacturer)

a. PPPoE on modem or broadband router

First you will have to read your modem's or router's manual. This will give you the fixed IP address where your modem's or broadband router's configuration page is usually located (say 192.168.1.2 as it is the case for my Westell ADSL modem or 192.168.254.254 where my Siemens Speedstream has its own)

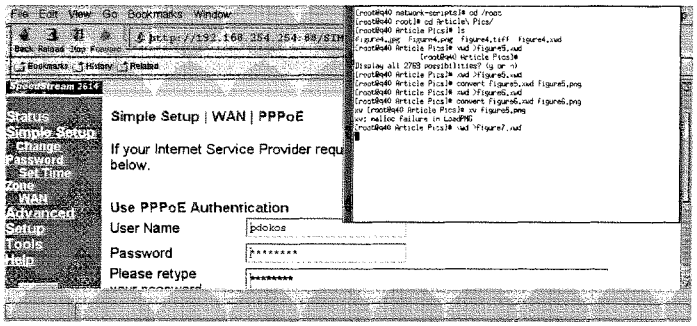


Figure 7 - Configuration of Router via Browser

Setup your machine as per the instructions for LAN with static IP address in the same network as the router/modem and directly connect to it via a browser -figure 7; see internet access below- (or telnet if your modem's web page won't work with the browsers provided with Shoestring Linux -or- if it only provides you with a telnet interface). Then, enter your account's username and password, as provided by your ISP and you are all set.

b. PPPoE software on your Qx0

In case, that your modem is older, you will need to run software on your Qx0 to connect to it. There are two ways to do that: Completely manual and semi-automatic.

1. Manual Setup

In order to do that, fire up Midnight Commander again and go to `/etc/ppp`. There, edit the file `pppoe.conf` to conform to the settings provided by your ISP (figure 8). Make sure you add the username required to connect to your adsl/cable account.

Then edit the file pap-secrets and add there as well the username (and this time also password) for your account (figure 9).
Finally type `adsl-start` and your adsl connection will start. Verify your settings with `ifconfig` and route as before. `adsl-stop` will sever the connection and `adsl-status` will show you the status.

2. Semi-automatic setup

There is an additional script which can simplify your life when it comes to PPPoE connections. That is `adsl-setup` which will ask you a few questions and then modify the appropriate configuration files for you. As with 1 above, you will need to issue an `adsl-start` command to start the connection and `adsl-stop` to stop it.

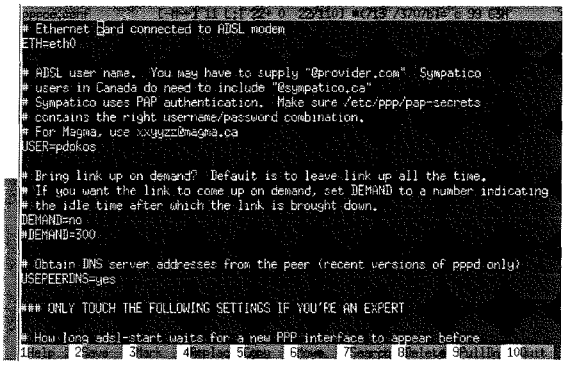


Figure 8 - pppoe.conf file

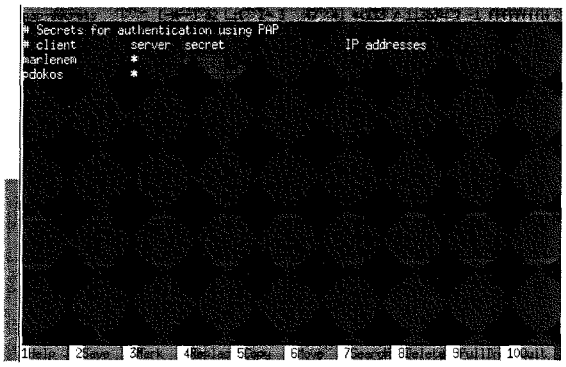


Figure 9 - pap-secrets file (pasword erased)

5f. Setup for Dialup

1. Introduction

Dialup connections are by far the most common, although they are quickly being pushed aside from ever-diminishing prices of Cable and xDSL. Most dial-up connections these days are of the PPP type, which is the type that we will concentrate on.

2. Verify your modem's port and device

Before you start, you will need to know your modem port and device. For serial ports, Linux starts counting from 0, so the port that your modem is connected should be in the range of /dev/ttysx where x=0 to 4 (There are many more serial ports supported under Linux, however due to the particularities of the Qx0 hardware, it is unlikely that you will ever have more than 4 installed. Additionally there's another device called /dev/modem. This is a symbolic link to one of the serial ports on Shoestring Linux and it is there for your benefit, as it is always easier to recognise the word modem `;-)` (Plus programs may use it directly.)
I recommend, using port 0 (ser1_ on QDOS) as it is always easier unless you install an internal ISA modem. If that is the case, first make sure that:

- Your modem is NOT a Winmodem (Winmodems require Windows to run)
- If your modem is Plug and Play, that it also has a 'jumpered' mode where you can set up its settings manually. Although pnp support is rather good in Shoestring Linux, not everything I have tried worked
- Setup your modem as ser4_ (COM4 on the PC) I/O port 2E8, IRQ: 3. Other settings are possible but that will work for sure.

Assuming now that you have an external modem on ser1_ (ttyS0), you can set up your connection and subsequently connect to the Internet, manually, semi-manually using `wvdial` or using the more elaborate and X-based `netcfg`. We however due to its simplicity, we'll use `netcfg`

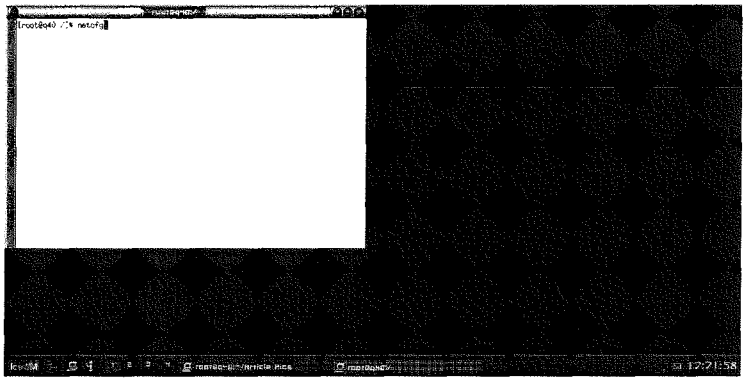
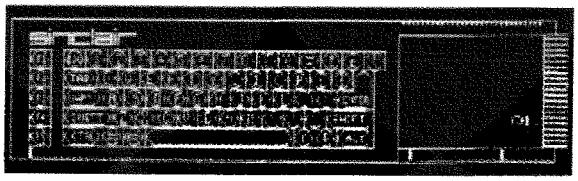


Figure 10 - IceWM Window Manager ready to run netcfg

3. Setting up

You will need to start the X-Windows GUI (Although that is not completely accurate as it is NOT the GUI, but rather the framework upon a GUI is based) with `startx` and ENTER.

QUANTA



Independent QL Users Group

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

Barclaycard: Visa: Access: MasterCard: Accepted

Now in our Twenty Second Year

Further details from the Membership Secretary

John Gilpin, 181, Urmston Lane
Stretford, Manchester, M32 9EH (UK).
Tel. +44 (0) 161 865 2872
Or
Visit the Quanta Web Site
http://www.quanta.org.uk
E-mail: membership@quanta.org.uk

Next QUANTA Sponsored Event

NEMQLUG
MANCHESTER WORKSHOP & QUANTA AGM
3rd Davyhulme Scout Headquarters,
"The Endeavour", Conway Road,
DAVYHULME,
MANCHESTER.
Saturday & Sunday 8th /9th April 2006
From 10.00 am to 4 p.m. daily. AGM at 2.00 p.m. Sunday.
Ring Sarah Gilpin on
0161 – 865 2872 for full details.

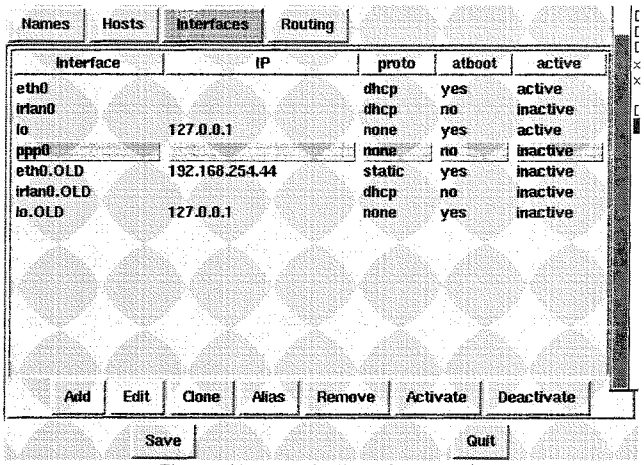


Figure 11 - netcfg 'interfaces' tab

Once the icewm (that's the default GUI) is loaded (figure 10) is loaded, go to the open terminal window and type `netcfg` and ENTER

Once this is loaded, go to "Interfaces", (figure 11) click on the ppp0 to select it and click on the Edit button below

Next the 'Create a new Internet Connection' dialog will pop-up (figure 12). Clicking 'Next', if you haven't setup a modem yet (which you haven't), will take you through the steps of finding and configuring your modem (figures 13 and 14)

Once the modem has been setup, you will be asked to enter the name and phone number of your account (figure 15). Refer to your ISP's instructions for the appropriate data. When you are done, click on "Next".

Consecutively, enter your username and password (figure 16) and click "Next"

Once, this too is done, you will be shown the new account that you created (figure 17).

Clicking "Finish" will commit the account information to the configuration files, thus making the account a part of your setup. You will then be presented with a list of the available dial up connections (figure 18). Clicking 'Close' will take you back to the first netcfg screen.

To connect to the Internet, you can either bring up netcfg, select the ppp0 connection and click 'Activate' (figure 19), or from the terminal, type ifup ppp0 (ifdown ppp0 will hang up the modem and close the connection)

Figure 13 - Automatic Search for Modem

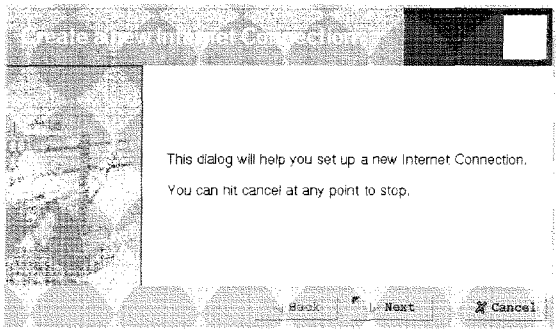


Figure 12 - 'Create a new Internet Connection' dialog

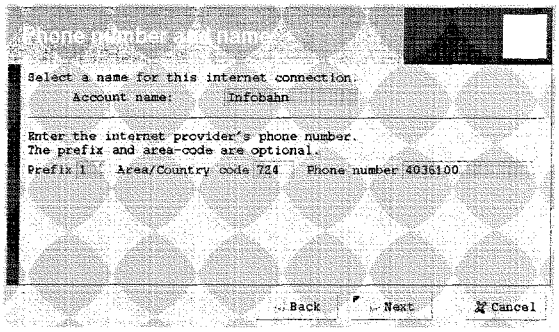


Figure 15 - 'ISP Phone number and Name' dialog

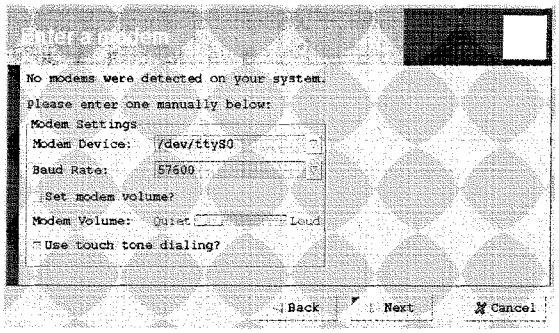


Figure 14 - Modem Setup Dialog (Manual)

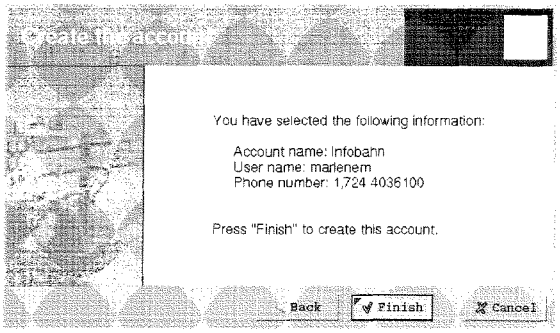


Figure 17 - 'Create the account' dialog

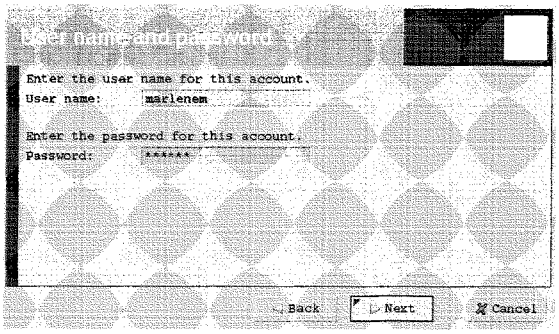


Figure 16 - 'User name and password' dialog

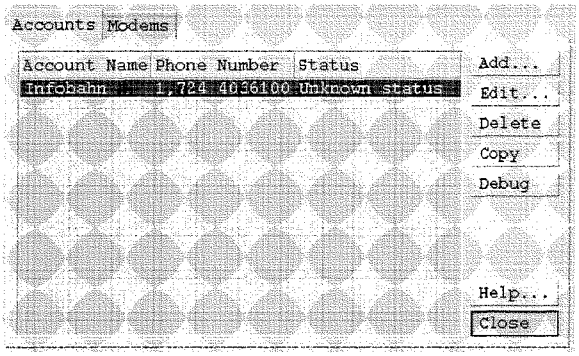


Figure 18 (above) - 'Dial up Connections' dialog

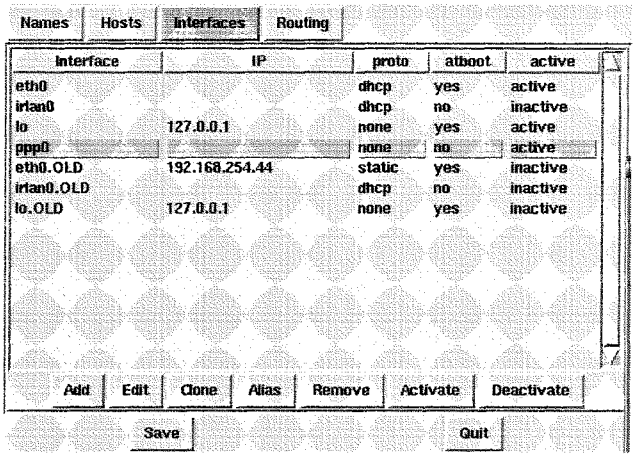


Figure 19 (right) - netcfg 'Interfaces' screen with Active ppp connection

5g. Using the connection

Unlike the previous parts of the series due to the quantity of programs available to Linux for Internet access I will not show you configuration options etc. Instead I will point you to a few of these programs and how to start them. Once you become comfortable with these you can explore the documentation and installed packages of Qx0 Linux to see what else is available to you.
Let's assume that your connection is up and running. Then it's time for...

1. Browsing

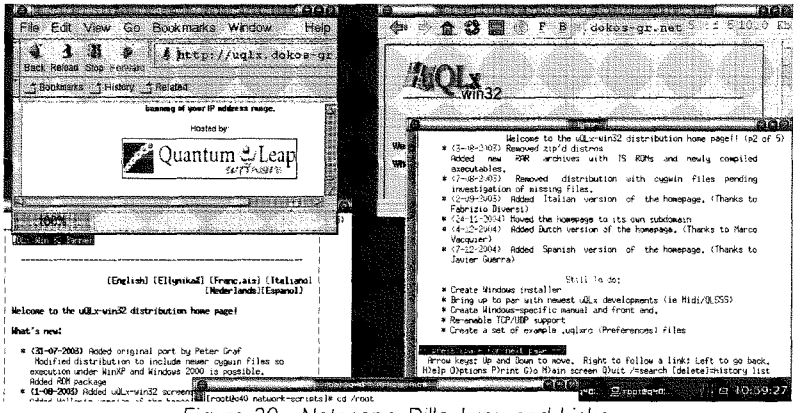


Figure 20 - Netscape, Dillo, Lynx and Links

As with almost everything in Linux, browsing can be done in text or graphic mode. By default there are four browsers available on Shoestring Linux (g is for graphical, t is for text-mode): Netscape (g), Dillo (g), Lynx (t) and Links (t) (a more user-friendlier Lynx clone).
To start Netscape or Dillo, you will need to be in an X based GUI or Desktop (Gnome -one of the best desktops in my humble opinion- is included with Shoestring Linux). Fire it up with `startx` (see previously on dial-up configuration). Then for Netscape, either click on the Netscape "Lighthouse" icon next to the IceWM start button (bottom left of the screen), via the "IceWM › Netscape" entry, the "IceWM › Applications › WWW Browsers › Netscape" entry, or, finally by typing `netscape` and ENTER on your X-Terminal.
To start Dillo, you should go to an X-Terminal type `dillo` and press ENTER
For Lynx (and Links) you can either be in a GUI, or your console. Type `lynx` and press ENTER (rest of instructions are as per the previous two article parts). Links can be started by typing `links` and pressing ENTER. Esc will bring up the menu (it is pretty self-explanatory from then on)
Figure 20 shows IceWM running all four programs at once.

2. Email

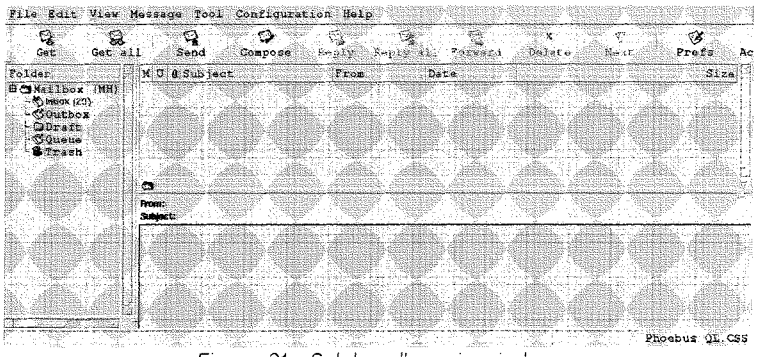


Figure 21 - Sylpheed's main window

Similarly to the browsers, email clients are also abundant in Linux. For that reason I will only mention Sylpheed - a full featured graphical email and news client that works very well. There are two ways of starting Sylpheed: If you are in IceWM click on the little envelope article at the bottom right of your screen, or, go to a terminal, type `sylpheed` and press ENTER (fig. 21).

3. FTP

One of the easiest to use FTP clients for Q40 Linux is gFTP (as a matter of fact, this article is uploaded using gFTP!). As with Sylpheed, it too requires X. Run it, by going to an X-terminal, typing `gftp` and pressing ENTER (figure 22). Another (and the de-facto standard) of course is ftp (t) which you can run by typing `ftp` at the console

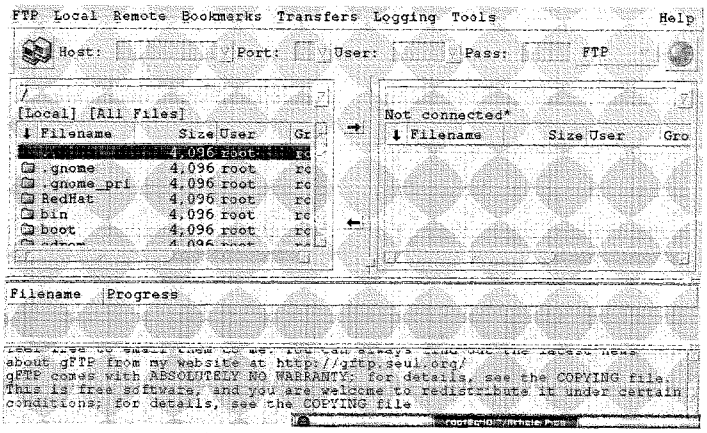


Figure 22 - gFTP's main window

4. News

For news, there are two main packages: Sylpheed (mentioned above) and Pan. To start Pan, type pan and press ENTER at your X-terminal. (Figure 23). Additionally you can use pine (t) and x-emacs for news.

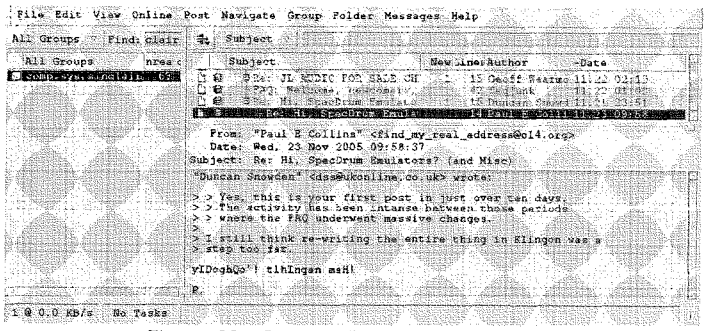


Figure 23 - Pan reading comp.sys.sinclair

5. Telnet

To start telnet, go to an X-Terminal or the console, type telnet and ENTER. You will be presented with the telnet› prompt. Type help for help on telnet's commands

6. Conclusion & Author's Disclaimer

This part of the series is by no means a Linux tutorial. What it is, is a small attempt to help Qx0 users navigate the complex world of Shoestring Linux, as to facilitate the exploitation of their machine's many abilities. Moreover, the configuration settings and commands are what worked for me. They may not work for you in exactly the same way, as you may have done something different with your installation or if you may have a different version of Linux (ie Q40 Linux). As you may have noticed (sic!) the programs did not get much attention by means of providing details on how to operate them. The reason for that is that most of them (at least the GUI based ones) provide extensive help files. I can go back and revisit specific ones in later articles if readers so wish. Just let the Editor know and we'll see what we can do!

This article concludes the series as it stands currently. Hopefully in the next issue, I will be able to bring some new and exciting new developments on the Net on the QL front. Keep your fingers crossed!

Oh and one more thing: I have received messages, letting me know that readers could not find the software (For the second time) on my website. This sadly was true but it has now been corrected. For some reason my Windoze ftp program did not transfer them properly. gFTP did that on my Q40. (The moral? Always trust QL hardware!)

7. Sources & Bibliography

1. SuSE Linux 6.3 Handbook, SuSE GmbH, 1999, Chapter 7, Section 4 (pp. 159-166)
2. **http://www.geocities.com/SiliconValley/Bay/2602/q40.html** (Old Q40 Linux website by Richard Zidlicky, contains pointers to the loader and precompiled kernels etc.)
3. **https://sourceforge.net/projects/linux-q40/** (Shoestring Linux website)
4. **http://www.yolinux.com/TUTORIALS/LinuxTutorialPPP.html**
5. **http://xwinman.org/icewm.php** (X-Windows Information site)
6. **http://www.linuxheadquarters.com/howto/networking/netcfg_modem.shtml**
7. **http://www.tuxfiles.org/linuxhelp/**

SVSCR and more

by Geoff Wicks

Elsewhere in this issue of QL Today George Gwilt writes about using the QL to demonstrate a lecture. Those readers who were fortunate enough to be at QL is 21 saw this in practice, when George demonstrated his screen snatcher programs.

These programs, the SVSCR suite, have been around for over three years, but are relatively unknown. As far as I know this is the first review of them. The suite consists of four programs written in Turbo PTR. To run the programs you will have to have a recent version of the Turbo Toolkit (V. 3.32 or later) installed on your machine.

GSV_task is the mother program that sits discreetly in the background waiting for a signal that you want to save part of the screen.

SVSCR_task works in conjunction with GSV_task and does the actual screen saving.

PR2WIN_task allows you to view the saved partial screens and to change their size or distort them.

PSA2PAT_task converts a partially saved screen into a sprite or pattern.

GSV_task and SVSCR_task

GSV_task and SVSCR_task work in conjunction with one another, and thus GSV_task needs to know where to find SVSCR_task. It is the only one of the four programs to have a configuration block. This has four items - the key combination to save a screen; the key combination to exit the program; the output file name of a saved screen; and the directory for the program SVSCR_task. You can also modify the default items at run time by starting the program with parameters.

GSV_task sits in the background until you press the screen save key combination, which, by default is SHIFT/TAB. When you do this the program SVSCR_task is loaded and a rectangle appears on the screen. You can move this using the cursor keys or, if you are a speed fanatic, CTRL + cursor. To modify the size of the rectangle you use ALT + cursor or if you are impatient ALT + CTRL + cursor. When you have the position and size you want, you press ENTER to save it. By default the saved file is ram1_sc1_psa. If you save a second partial screen this will be ram1_sc2_psa.

Some years ago Phil Jones wrote a program for saving partial screens called graBIT and, of the two, I find George's program the easier to use because it gives a better view of the portion of the screen you wish to save. However the advantage of graBIT is that its screens are saved in _pic format, giving greater compatibility with other programs, whereas SVSCR screens are saved in _psa format.

Until George sent us some screens saved in _psa format for publication in QL-Today, I knew nothing of the difference between _psa and _pic screens, although both are detailed in the QPTR manual.

The _pic format has become the standard format for saving screens that are not the standard 512 x 256 size. Prior to the screen information there is a 10 byte header. This consists of a flag identifying it as a _pic file, x and y sizes in pixels, line length in bytes and the mode flag.

A _psa file is the same as a _pic file with the exception that the header is 12 bytes instead of 10 bytes. The extra two bytes, at the start of the file, are spare for the programmer's own use.

It is fairly easy to write a SuperBasic program to strip off the first two bytes of a _psa file to convert it into _pic.

Although SVSCR_task usually works in conjunction with GSV_task it can be used independently if you already have some saved screens on file.

PR2WIN_task

The SVSCR programs are not just screen snatchers, but also screen manipulators. PR2WIN_task looks for _psa files in ram1_ and displays them one by one. The displayed partial screen can be resized and distorted horizontally, vertically or spherically.

Partial screens can be moved and resized using the cursor and CTRL + cursor keys as in SVSCR_task. The resized picture can be restored to the same proportions as the original on pressing F2.

If you press F3 a window is opened allowing distortion.

When you have finished manipulating an image you can press ENTER, and, if the image has been modified, it is saved to ram1_ with a slightly different file name from the original. When you have finished modifying and viewing all the files they are printed to the screen. In this way you could design a montage made of several different partial screens.

PR2WIN is not fully functional on all systems and

QPC2 is one of the least functional. All you can do in QPC2 is move a partial screen. There is a reason for this. Until recently George has done his QL development on a Q60, which has a 68020 processor whereas QPC2 emulates the 68000 processor.

In the manual George provides a useful summary of what can be done on which machine. Moving can be done on most machines. Resizing requires a 68020 processor and is possible with a Gold Card, QXL, Q40 or Q60. Distorting requires an FPU and should work with Q40, Q60 and some QXLs.

As my main QL platform is QPC2 I was unable to test this program extensively. My attempts to run the SVSCR programs on my JM and Minerva black boxes ended in failure, although all ran perfectly on a Gold Card version of SMSQ/E. I should add that I had to take my black boxes out of a lengthy retirement and both are not only elderly but also infirm. Both refused to load the latest version of the Turbo Toolkit with an error message "not implemented".

PSA2PAT_task

The final program in this suite is PSA2PAT which allows you to convert a partial screen into a solid sprite, a transparent sprite or a pattern. The sprite has to be rectangular, but if the transparent option is chosen all black portions of the image will be invisible. There is also a restriction on the sprite size. X size multiplied by Y size must be no larger than 4092.

The program is simple to use. First you are asked for the input file; then the output file and finally if you want to save the sprite as solid, transparent or a pattern.

Those readers who were at QL is 21 will remember how George illustrated this program by making a sprite from Tony Firshman's head that he had saved from a colour photo.

Most of us who source sprites from PC material probably use Wolfgang Lenerz's program BMP2-SPRT, but PSA2PAT could be a useful alternative. You could for example display a digital camera image in photon, save the part of the photo you want as a sprite and then convert using PSA2PAT.

Fig. 1

I was hoping to illustrate the SVSCR suite by examples of resized partial screens, but for the reasons given above this was not possible. Instead I displayed a jpg image using photon. Fig. 1 shows this image with the save area rectangle around the inn sign. Fig. 2 is a montage of partially saved areas of the main photo.

Fig. 2

The SVSCR suite is probably of more interest to the specialists than the average QL user, but it is worth looking at. The programs can be downloaded from the SQLUG site:

www.jms1.supanet.com

As an added bonus you can also download the source code.

BIKE

by Steve Poole

BIKE is a listing which Steve sent together with CONIC - published in an earlier issue of the last volume. It is a vector graphics demonstration. I must admit that I lost track of Steve's disk when I moved the office from the old address to the new one. Sorry about that - Editor.

```
110 REMark BIKE_2_bas by S.Poole v30nov92, v18nov2004
120 CLEAR: main: STOP
130 :
140 DEFine PROCedure main
150  INIT: OVER 0
160  FOR f=1 TO 25: PAPER RND(255): SCROLL RND(f)*-1
170 FOR loop2=1
180     INK 2,0,1: FILL 1: CIRCLE 84,31,31 : FILL 0
190     INK 0    : FILL 1: CIRCLE 84,31,25 : FILL 0
200     INK 2,0,1: FILL 1: CIRCLE 179,31,31: FILL 0
210     INK 0    : FILL 1: CIRCLE 179,31,25: FILL 0
220     :
230     FOR LOOP=1,4,8,9,2,3,5,6,7
240         GET_array: AZIMUT: THRUTURN: TYPEM: FILL 1: DRAW: FILL 0
250         IF LOOP='1': INK 0,2,1: FILL 1: CIRCLE 121,25,10: FILL 0
260     END FOR LOOP
270     PAN RND(1 TO 9)*-1: OVER -1
280 END FOR loop2: i$=INKEY$(#1,-1)
290 END DEFine
300 :
310 DEFine PROCedure TYPEM
320 FOR f=1 TO dots+3
330     s$=sig$(sgn(t(f,trn))+2)&sig$(sgn(t(f+1,trn))+2)&' '
340     tp=s$ INSTR sg$: t(f,typ)=((tp+2)/3)-1
350 END FOR f: t(dots+4,typ)=t(4,typ)
360 END DEFine TYPEM
370 :
380 DEFine PROCedure THRUTURN
390 FOR f=2 TO dots+3
400     th=t(f,azt)-t(f-1,azt)
410     IF th<-PI: th=th+rd360
420     IF th> PI: th=th-rd360
430     t(f,trn)=th
440 END FOR f: t(1,trn)=t(dots+1,trn)
450 END DEFine THRUTURN
460 :
470 DEFine PROCedure AZIMUT
480  POINT t(1,x),t(1,y)
490 FOR f=1 TO dots+3
500     j=f+1: xx=t(j,x)-t(f,x): yy=t(j,y)-t(f,y)
510     dd=SQRT(xx*xx+yy*yy): sx=sgn(xx): sy=sgn(yy)
520     az=0: IF dd: az=ASIN(yy/dd)
530     :
540     SELect ON sx
550        =1: IF sy=-1: az=rd360+az
560        =REMAINDER : az=PI-az
570     END SELect : t(f,azt)=az
580     REMark LINE TO t(f,x),t(f,y)
590 END FOR f
600 END DEFine AZIMUT
610 :
620 DEFine FuNction sgn(sg)
630  RETurn (sg>0)-(sg<0)
640 END DEFine
650 :
660 DEFine PROCedure DRAW
670 FOR f=3 TO dots+2
680     a=f-1: b=a+1: c=b+1: d=c+1: tipes=t(f,typ)
690     ax=t(a,x): ay=t(a,y): bx=t(b,x): by=t(b,y)
700     cx=t(c,x): cy=t(c,y): DX=t(d,x): DY=t(d,y)
```

```
710     BCx=bx-cx: BCy=by-cy: MCx=BCx/2: MCy=BCy/2
720     Mx=bx-MCx: My=by-MCy
730     azb=t(b,azt): az1=(t(a,azt)+azb)/2
740     az2=(azb+t(c,azt))/2: az3=azb+RAD(90): ef=1
750     :
760     SELect ON tipes
770        =0,8: L(1,1,x)=bx+COS(az1): L(1,1,y)=by+SIN(az1)
780              L(1,4,x)=cx+COS(az2): L(1,4,y)=cy+SIN(az2)
790              L(1,2,x)=bx: L(1,2,y)=by
800              L(1,3,x)=cx: L(1,3,y)=cy
810    END SELect
820    :
830    SELect ON tipes
840       =1,2,6,7: bx2=bx: by2=by: cx2=Mx: cy2=My
850                 L(2,1,x)=bx2+COS(az1): L(2,1,y)=by2+SIN(az1)
860                 L(2,4,x)=Mx+COS(az3) : L(2,4,y)=My+SIN(az3)
870                 L(2,2,x)=bx2: L(2,2,y)=by2
880                 L(2,3,x)=cx2: L(2,3,y)=cy2
890    END SELect
900    :
910    SELect ON tipes
920       =2,3,5,6: bx3=Mx: by3=My: cx3=cx: cy3=cy
930                 L(3,1,x)=Mx+COS(az3): L(3,1,y)=My+SIN(az3)
940                 L(3,4,x)=cx3+COS(az2): L(3,4,y)=cy3+SIN(az2)
950                 L(3,2,x)=bx3: L(3,2,y)=by3
960                 L(3,3,x)=cx3: L(3,3,y)=cy3
970    END SELect
980    :
990    SELect ON tipes
1000       =0: TRI 1,0: =1: TRI 2,0: =2: TRI 2,0: TRI 3,1
1010       =3: TRI 3,0: =5: TRI 3,1: =6: TRI 2,1: TRI 3,0
1020       =7: TRI 2,1: =8: TRI 1,1
1030    END SELect
1040 END FOR f
1050 END DEFine DRAW
1060 :
1070 DEFine PROCedure TRI(TR,CW)
1080  sax=L(TR,1,x): say=L(TR,1,y)
1090  sbx=L(TR,2,x): sby=L(TR,2,y)
1100  scx=L(TR,3,x): scy=L(TR,3,y)
1110  sdx=L(TR,4,x): sdy=L(TR,4,y)
1120  spx=sax-sbx: spy=say-sby: sg1=0
1130  IF spx: sg1=spy/spx: END IF : sy1=sax*sg1
1140  sqx=scx-sdx: sqy=scy-sdy: sg2=0
1150  IF sqx: sg2=sqy/sqx: END IF : sy2=scx*sg2
1160  sk1=say-sy1: sk2=scy-sy2: skk=sk2-sk1: sgg=sg1-sg2
1170  ix=0: IF sgg: ix=skk/sgg
1180  sgx=sg1*ix: iy=sgx+sk1
1190 IF spx=0: ix=sax: iy=scy+(sg2*(sax-scx))
1200 IF sqx=0: ix=scx: iy=say+(sg1*(scx-sax))
1210 IF TR=2: ix=(ix+bx)/2: iy=(iy+by)/2
1220 IF TR=3: ix=(ix+cx)/2: iy=(iy+cy)/2
1230   REMark INK 2: LINE scx,scy TO ix,iy TO sbx,sby: INK 7
1240  BCx=sbx-scx: BCy=sby-scy: MCx=BCx/2: MCy=BCy/2
1250  Mx=sbx-MCx: My=sby-MCy
1260 mix=ix-Mx: miy=iy-My: im=SQRT(mix*mix+miy*miy)
1270  Maj=SQRT(MCx*MCx+MCy*MCy): bm=SQRT(BCx*BCx+BCy*BCy)/2
1280  ibx=ix-sbx: iby=iy-sby: ib=SQRT(ibx*ibx+iby*iby)
1290 im2=im*im: ib2=ib*ib: bm2=bm*bm
1300  dv=(bm2+ib2-im2)/(2*ib*bm): hi=ib*SIN(ACOS(dv)): sl=TR
1310 :
1320 SELect ON sl
1330    =1: xi=Mx: yi=My: =2: xi=bx: yi=by: =3: xi=cx: yi=cy
1340 END SELect
1350 :
1360 REPeat rp
1370    IF hi>(Maj*2.7) THEN
1380       ix=(ix+xi)/2: iy=(iy+yi)/2: hi=hi/2
1390       ELSE EXIT rp
1400    END IF
1410 END REPeat rp
```

```
1420  Min=hi*.44: ecc=Min/Maj
1430  ELIPTICS: PROJECT
1440 END DEFine TRI
1450 :
1460 DEFine PROCedure ELIPTICS
1470  LOCal f,j,ct
1480  ro=0: IF scx<sbx: ro=180
1490  h=20: REMark SELect ON ecc: =.2 TO .5: h=40: =.5 TO 1: h=60: =REMAINDER : h=90
1500  q=h+1: st=360/h: rot=0
1510  IF MCx: rot=DEG(ATAN(MCy/MCx))
1520  rot=rot+ro: h=h-4
1530  hh=h/2: ct=0: SS=-st: AA=180-st: ZZ=st
1540  IF CW=0: SS=st: AA=-AA: ZZ=-ZZ: ct=hh
1550  DIM u(pp,h+1,y)
1560  :
1570 FOR f=AA TO ZZ STEP SS
1580     IF f=90 OR f=-90 THEN
1590        NEXT f
1600        ELSE
1610        ct=ct+1: j=RAD(f)
1620        Crx=Maj*COS(j): cry=Maj*SIN(j)*ecc
1630     END IF
1640     ROTATE Crx,cry: u(ee,ct,x)=Mx+xx: u(ee,ct,y)=My+yy
1650     ROTATE Crx,0  : u(pp,ct,x)=Mx+xx: u(pp,ct,y)=My+yy
1660 END FOR f
1670 END DEFine ELIPTICS
1680 :
1690 DEFine PROCedure PROJECT
1700  LOCal f,j,ct: ccx=scx: ccy=scy
1710  IF TR=3: ELSE POINT sbx,sby
1720  IF CW: j=0: ct=hh: ELSE j=hh: ct=h
1730  nd=j+hh: xa=ix: ya=iy
1740  :
1750 FOR f=j+1 TO nd
1760     xb=u(2,f,x): yb=u(2,f,y): xc=u(1,f,x): yc=u(1,f,y)
1770     xd=u(1,ct,x): yd=u(1,ct,y)
1780     xp=xa-xb: yp=ya-yb: xq=xc-xd: yq=yc-yd
1790     g1=0: IF xp: g1=yp/xp
1800     g2=0: IF xq: g2=yq/xq
1810     k1=ya-xa*g1: k2=yc-xc*g2: kk=k2-k1: gg=g1-g2
1820     xx=0: IF gg: xx=kk/gg: END IF : yy=g1*xx+k1
1830     IF xp=0: xx=xa: yy=yc+(g2*(xa-xc))
1840     IF xq=0: xx=xc: yy=ya+(g1*(xc-xa))
1850     LINE TO xx,yy: ct=ct-1
1860 END FOR f
1870  IF TR=2: ELSE LINE TO ccx,ccy
1880 END DEFine PROJECT
1890 :
1900 DEFine PROCedure ROTATE(rx,ry)
1910  LOCal rh,sx,sy,ar,rb
1920  sx=(rx>0)-(rx<0): sy=(ry>0)-(ry<0)
1930  rh=SQRT(rx*rx+ry*ry): ar=DEG(ABS(ASIN(ry/rh)))
1940  :
1950 SELect ON sx
1960    =-1: SELect ON sy: =-1: ar=180+ar: =0: ar=180: =+1: ar=180-ar
1970    =0 : SELect ON sy: =-1: ar=270   : =0: ar=0  : =+1: ar=90
1980    =+1: SELect ON sy: =-1: ar=360-ar: =0: ar=0  : =+1: ar=ar
1990 END SELect : rb=ar+rot
2000 :
2010 IF rb<0   : rb=rb+360: GO TO 2010
2020 IF rb>=360: rb=rb-360: GO TO 2020
2030 IF rb>270: rb=360-rb: sx=+1: sy=-1: GO TO 2110
2040 IF rb=270: rb=90    : sx= 0: sy=-1: GO TO 2110
2050 IF rb>180: rb=rb-180: sx=-1: sy=-1: GO TO 2110
2060 IF rb=180: rb=0     : sx=-1: sy= 0: GO TO 2110
2070 IF rb>90 : rb=180-rb: sx=-1: sy=+1: GO TO 2110
2080 IF rb=90 : rb=90    : sx= 0: sy=+1: GO TO 2110
2090 IF rb> 0 : rb=rb    : sx=+1: sy=+1: GO TO 2110
2100 IF rb= 0 : rb=0     : sx=+1: sy= 0
2110  rb=RAD(rb): xx=rh*COS(rb)*sx: yy=rh*SIN(rb)*sy
2120 END DEFine ROTATE
```

```
2130 :
2140 DEFine PROCedure INIT
2150  OPEN#1,con_128: WINDOW 512,256,0,0
2160  SCALE 148,44,-8: PAPER 0: INK 4: CLS
2170  sg$='++ +0 +- 0+ 00 0- -+ -0 -- ': sig$='-0+'
2180  ee=1: pp=2: h=20: hh=10
2190  rd180=PI: rd360=2*PI: pi2=2*PI
2200  x=1: y=2: DX=3: DY=4: dist=5: azt=6: trn=7: typ=8
2210 END DEFine INIT
2220 :
2230 DEFine PROCedure GET_array
2240 IF LOOP='1': RESTORE 2480 : READ dots:   INK 7,2,1: REMark rearleg
2250 IF LOOP=2  : RESTORE 2510 : READ dots:   INK 7,4,1: REMark foreleg
2260 IF LOOP=3  : RESTORE 2540 : READ dots:   INK 7,4,1: REMark arm
2270 IF LOOP=4  : RESTORE 2570 : READ dots:   INK 7,4,1: REMark head
2280 IF LOOP=5  : RESTORE 2600 : READ dots:   INK 2,0,1: REMark helmet
2290 IF LOOP=6  : RESTORE 2630 : READ dots:   INK 4,2,3: REMark shirt
2300 IF LOOP=7  : RESTORE 2450 : READ dots:   INK 2,4,1: REMark short
2310 IF LOOP=8  : RESTORE 2420 : READ dots:   INK 4,0,1: REMark frame
2320 IF LOOP=9  : RESTORE 2390 : READ dots:   INK 4,0,1: REMark bars
2330 :
2340 DIM t(dots+4,typ),L(3,4,2)
2350 FOR f=1 TO dots: READ t(f,x)
2360 FOR f=1 TO dots: READ t(f,y)
2370 FOR f=1 TO 4: t(f+dots,x)=t(f,x): t(f+dots,y)=t(f,y)
2380 :
2390 DATA 21, 135,142,153,174,190,194,192,181,172,171,176,182,185,179,174,172,166,159
2400 DATA 155,140,134,66,68,71,76,79,77,75,70,65,60,48,37,30,25,30,37,50,62,63,64,65
2410 :
2420 DATA 21, 91,98,105,115,129,149,155,154,149,133,124,114,101,85,82,85,96,106,103
2430 DATA 97,90,  96,95,84,68,67,66,65,64,63,53,39,35,30,27,30,33,39,56,75,88,91
2440 :
2450 DATA 15, 114,116,119,122,125,138,138,129,123,120,113,104,98.5,100,110.5
2460 DATA 119.5,116,111,101,98,91,87,78.5,74,74,79,87,95,109,118
2470 :
2480 DATA 13,  128,138,135,129,136,134.5,127.5,119.5,119,124,125,121.5,120
2490 DATA      82.5,72,57,36.5,24.5,17.5,21.5,31.5,38.5,58,69.5,73,77
2500 :
2510 DATA 15,126,139,147.5,146,136,124.5,114.5,119,117.5,107,106.5,114,125,132.5,128
2520 DATA 90,89.5,82,72.5,64.5,58.5,49,38.5,33.5,46.5,55.5,60,69,77,78.5
2530 :
2540 DATA 17,  161,164,169,181,190,198,202,203,200,197,190,176,165,161,158,154.5,155
2550 DATA      94,90.5,84.5,82,81,80,78,74,73,71,73,73,73.5,74,81,87.5,92
2560 :
2570 DATA 11,  171,181,179.5,178,176,172,171,168,162,161,164
2580 DATA 122,111,107,103,103.1,100.5,98.5,99.5,102,112.5,119
2590 :
2600 DATA 11,    149,154,167,181,184,179.5,171.5,164,160.5,152.5,149
2610 DATA 125,125.5,127,123,112,106.5,105,108.5,113.5,121.5,123
2620 :
2630 DATA 18,116,126,138,149,160,161,163,164,160,154,153,144,133,125,122,121,119,114
2640 DATA 121,125,127,123,114,108,98.5,91,88,88,94,98.5,99,101,101.5,104,111,119
2650 END DEFine GET_array
2660 ::
```

Do you remember...? – Part 2

by Ralf Reköndt

In the last issue, we promised to publish more information about the SPK in the next issue, i.e. the one you are reading right now.
Ralf informed us that he was unable to send us the additional information due to health problems. So, you need to be patient - it will be in the next issue. Meanwhile, a few more "do you remember..." pictures for you.

Ralf writes:

QL-Cash-Trader

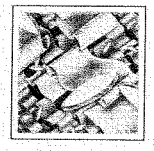
Written in BASIC using the SPK. A rather complex program. The original program filled parts in the database with zeros, which often led to problems using microdrive cartridges.

QL Intergated Accounts

Written in C. Semi-professional program for book-keeping. The cartridges contain an opcode of the program which can be started using the SuperBASIC command CRUN.

Although professionally programmed, there is one major drawback: the VAT-rate was not user-changeable, only by SAGESoft themself. If some body has the SAGESoft patch program, please report to QL Today.

QL Cash Trader



The accounting program specifically designed for the needs of any small business that is based mainly on cash rather than long-term credit.

Most small businesses are run by people with experience in things like marketing, engineering or management – not by accountants. But keeping financial records and working out the management accounts is often far from easy, particularly when most goods and services have to be paid for as they are supplied.

■ QL Cash Trader makes financial record-keeping simple for people who have many other demands on their time. It also provides speedy and accurate management information, and lets you see the effect of each transaction as it happens.

■ Because financial transactions may occur in a random order, you simply update sales, purchases, cash payments, standing orders, VAT, expenses, and so on, whenever you want. Without you needing to pre-sort by date or type, QL Cash Trader instantly processes the entries and displays a variety of reports – including a Profit and Loss account, bank balance, VAT liability, sales and purchase analysis.

■ Entries are made simple with a minimum of typing, and clearly presented report screens keep you informed. For a view of QL Cash Trader at work, see the reverse of this sheet.

Price:
£69.95
(includes comprehensive manual)

Software written by:
Accounting Software Ltd.
A Quest Group Company

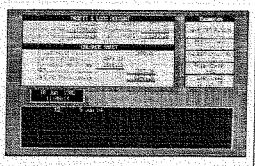
5503

sinclair

Using QL-Cash Trader

QL Cash Trader has been designed so that it can be tailored to suit the needs of a wide variety of different businesses operating in virtually every type of industry.

To make it simple for you to set up QL Cash Trader the way you want, the program contains a variety of example businesses from which you choose the appropriate options. A comprehensive manual is included, and this starts by teaching you how to enter typical transactions. It also shows you detailed accounts based on a number of different kinds of trading. Should you need additional help in using QL Cash Trader, there is a full support service with a free trial period.

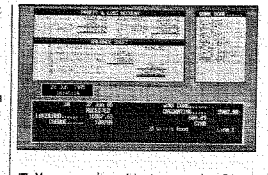


■ Before using QL Cash Trader for your own accounts, the program offers you the chance to run through the examples described in the manual. These are worked out on the main screen, which looks like this. The screen is divided into a number of windows, and this format is used for everything which you do with QL Cash Trader.

The window at top left is the Reports window. A choice of 18 different management reports can be displayed here on request. In this example, it shows a Profit and Loss account and Balance Sheet.

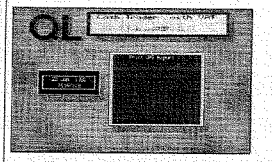
The Status window at top right displays various items of helpful information. At present these are VAT and financial year end dates, which you set to suit yourself.

At the bottom is the Entry window. This is where you fill in your transactions. You enter a description of each one, the amount, and any additional comments.



■ You can make entries in any order. QL Cash Trader already knows the date, and when you set it up for the first time, you will give it information such as the name of your bank account, and the types of sales and purchases you make.

Because the QL now has these details, you don't need to type them in every time you make an entry. You just hold down a key and it will roll through the options to reach the one you want. Then all you do is to fill in the appropriate amount.



■ When you enter the transaction, its effect instantly appears in the Report window. Any of the reports can be printed out if you wish – by arrangement with Customs & Excise, you can even use QL Cash Trader to produce your VAT returns.

After each session with QL Cash Trader you then save the data (up to 900 transactions on one cartridge) until you next want to update your records.

QL Integrated Accounts

A complete package for any business which wants to set up a fully integrated ledger-based accounts system. The program also provides a comprehensive range of reports.

Proper book-keeping is essential to any business. But, for a small business, having it done professionally can be a major expense. QL Integrated Accounts makes it possible for anyone to organise a properly-managed accounting system – saving time and money.

■ QL Integrated Accounts uses a tried and proven method based on the renowned Sage Accounting System – one of the UK's top selling accounting programs.

■ A comprehensive manual and tutorial program explain the principles of book-keeping in such a way that you don't need to be an accountant to follow them. In a very short time, you will be able to generate fully-integrated sales, purchase, and nominal ledgers. You can also request any of a suite of management reports, including trial balance, profit and loss, VAT, aged debtors and aged creditors, and a balance sheet.

■ Handles up to 300 customer accounts, 300 supplier accounts, and 1000 nominal accounts.

■ Clear graphic screens and simple entries make it easy to use this high-quality, professional program, and three months free 'Hotline' support is provided to new users. See how the system works overleaf.

Price:
£89.95
(includes comprehensive manual)

Software written by:
SAGESOFT

5508

sinclair

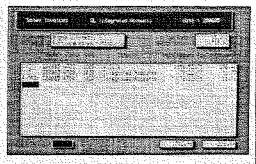
Using QL-Integrated Accounts

The QL Integrated Accounts package is designed for simplicity, and a detailed manual provides all the book-keeping knowledge you will need.

Even if you have never tackled accounting before, you will find that QL Integrated Accounts works in a way which is easy to understand. But that doesn't mean that it cuts any corners in quality. The program has been extensively developed with the aid of accountants and Customs & Excise to ensure it meets exacting professional standards.

QL Integrated Accounts is simple to use because it splits up book-keeping into easy stages. Each of these is clearly explained with the aid of examples to work through. All the screen information is presented so it is easy to understand, and simple to enter your data.

The program has two main sections; 'Postings', where you record invoices, credit notes, bank and cash transactions; and 'Reports', which contains all the routines needed to produce sales, purchase, and management reports from the posted data. You start by running the Postings program.

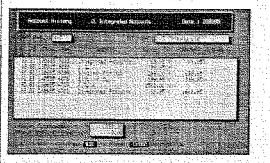


■ A menu will offer you a choice between the sales, purchase, or nominal ledgers. These contain options for creating new accounts, and recording details of invoices, credit notes, payments and receipts as appropriate. This picture shows an example of the screen used to enter an invoice in the sales ledger.

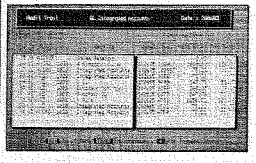
You will already have set up accounts for each of your customers. Now, entering the invoice details is simply a matter of filling in the lines by matching the details to the invoices themselves. Standard entries for things like VAT are made easy with a single key press, and it's also possible to duplicate entries if they are repeated.

When you have completed your entry, you are asked to confirm it. The screen data is then automatically filed in the correct places.

All the other entries in the ledgers are equally straightforward, so you won't find it a chore to keep your ledgers up to date.



■ Now you can compile reports from these details. The screen above shows just one of the huge range of reports which is available – a complete analysis of an account history showing all the sales invoices and credit note details.



■ Reports analyses are all based on an audit trail – the most important routine from your auditor's point of view. As well as providing vital management information, this can also be printed out in a form suitable for transferring to your VAT return – yet another of the many ways in which QL Integrated Accounts will save you time and money.

Time to Renew - No Price Increase!

Your Subscription to QL Today will expire with the next issue.

You will find a renewal form enclosed with this copy and it would help the administration of the magazine if you could complete this as soon as possible and return it with your subscription. (Subscribers with automatic renewal do not have to return it.)

Once again costs of both postage and printing have gone up since the last renewal but we have decided to hold the price down and not increase our cover charges to you so the good news is that you will pay the same this year as you did before.

Please help us to keep costs to a minimum by renewing your subscription early.

The next issue will contain an interesting Sudoku article from Norman, a report about the Quanta AGM, the latest Psion XChange developments, a hardware article and much more! We are also waiting for YOUR submission - please see deadline on page 2.

Byts of Wood

by Roy Wood

First column of a New Year so it is time to look back over the last twelve months and see what it brought us. Well the truth is quite a lot really. New programs and a few updates to old ones with one or two little bonuses left dangling at the end of the year to tempt us forward.

Usually I am one of those tardy people who leave their UK Tax Return to the last moment but this year I managed to move that last moment a couple of weeks forward so I actually had some time to look into the figures for both the 2003-2004 and 2004-2005 tax years before submitting the documents.

Sad to say, sales at the QBranch emporium were down on both years but the figures for the items we did sell were quite interesting. The most telling of the figures was for the 2004-2005 tax year. In spite of the conclusions based on the results for the survey that was conducted last year my overwhelming best seller was QPC2, closely followed by QPCPrint. I did have some second user QL hardware items and some QL Membranes for sale but these only moved very slowly at first indicating that those who do still use native hardware may have all they need.

Among the other programs on sale the T87 patch program did quite well and QDT made a strong showing. Sales for upgrades for Easyptr were not so good - certainly not deserving of the hard work Marcel put into it but that was always to be expected since, only a handful of current QL Users, really like to get their hands dirty with code.

This seems to say to me that many QL Users now have QPC2 on a PC in some form and, certainly amongst those who attend the workshops, a large number of these use a laptop for their QPC2 platform. I began to wonder what they actually used it for since there have been few really new utility programs for SMSQ/E in the last few years.

What Makes A Program Good?

In part, some of this curiosity was satisfied by a user who came over to me so I could solve a few hardware problems he had with his QL. He was using Abacus to run a series of spead-sheets and said that he had a PC at home but that Abacus 'beat Excell hands down.' Now that is clearly a very subjective viewpoint and not one that many people will hold - myself included. I have always found that Excell is one of the best programs in the Microsoft stable (probably because they did not write it - they bought it in from another software house). There is no doubt that it is more powerful and has more facilities than anything that we have to offer.

The key to this, however, lies in the simplicity of use. If you know how to set Abacus up then it is quicker to use that than to try to get to grips with a new program - especially when the new features in that new program will not be needed for the things you need to do. There was a move, some years ago now, to release some 'simple' versions of the Microsoft Office suite at a reduced price for people like this but, apart from 'Works' (and even this has a full version of Word in it) I have seen little evidence of it being pushed forward.

There is a tendency on the part of programmers to assume that the more features, bells and whistles they can build in the better the finished product will be. This is the Swiss Army Knife approach to programming that has seen so many of the commercial PC programs bloat up like a bunch of weight watchers on 'Sod it all, why not?' day.

When we started QBranch, over 10 years ago now, we used to boast that all of our leaflets, adverts and other material were produced using QL software. That, alas, is no longer the case. Some of this is due to issues outside my control, some people need files in more standard formats for instance, but I do still use QD to write this column and I do use LINEdesign to make the QL Today adverts in preference to Adobe Illustrator which I also own and use for the Quanta ads.

Help - What Help?

The situation with non-QL programs is not helped by their complete lack of manuals. (I did, much to my surprise, get a manual with Dreamweaver when I bought it last year.) I have, for instance, just spent an entire afternoon searching the online help for Adobe Acrobat Professional trying to find how to make an editable .pdf file which will calculate the values as you enter them. The online help was next to useless. Most QL programs come with good readable manuals and even those like QDT and QPCPrint and now QPC2 which has just been released in CD form for users with no floppy drive, which have manuals on disk are easy enough to follow.

Of course, we have grown up with the QL programs so we understand the concepts and the terminology. A user who switches to a PC finds something which has been evolving behind his back and he is in alien territory from the start. This goes a long way towards explaining why many of us stick with older programs when the newer ones sometimes offer us more. Even those within the QL scene.

Having written the above, the engine of Serendipity swung into action and the QL Users list on the Internet prompted Ian Pizer to write that he wanted to sort a list alphabetically but could not do it on his PC so he dropped it into QL mode and did it there with no problem. Hurrah for the QL you may be tempted to say until someone pops up and says 'This is how you could have done it in DOS'. You see it is all a question of where you came from. If you started on a QL then QL ways always seem to be the best.

Marcel recently did some hacks to improve Xchange and Roger Godley has been doing similar things with the original PSION suite to make them nicer to look at and more functional. Whilst I can only applaud these improvements it must be admitted that Quill is no more useful than the Wordpad program that comes with Windows - less so because it lacks a spell checker or different fonts. It is, however, infinitely smaller. It all depends on what you want.

Crossover.

Of course, one of the aspects that most befuddles users when changing from a QL in a black box with two hard drives is the fact that they now have to actually learn something new. This is, in part, one of the factors that led me to write the 'Start Here' series. A case in point is a customer who bought QPC2 from me in February. For a long time he has used his QL in tandem with his PC with both machines doing different jobs but he now wanted to run both things on one machine. Although, when he started out with his QL, he felt confident enough in his knowledge of SuperBasic to submit a program to a magazine for publication, his use of the QL for the last few years has been to run Quill and Abacus so he has forgotten all of the programming skills he had.

His drives, which worked well on his QL, produced disks which would not work on his PC and

he had no idea how to move the data. Even when the data and the programs had been moved it was clear that he would be struggling to write a coherent BOOT file and incorporate all of the things he needed to make the system run smoothly. Teaching him how to do this would be a long, and ultimately unprofitable, job but this is what he needed. He was not a QL Today or Quanta reader because he had 'little use for the things it was talking about' (except, of course, learning how to use the new technologies, writing a BOOT file and alll the other things that we cram into these pages. Oh Well) but he clearly needed help. I spent a few hours with him trying to get him to the stage that it would all work correctly, including changing one drive so he could produce disks that his laptop would recognise and trying to nudge him into looking at newer QL programs to use. What alternative was there? I could just leave him to it and let him carry using his high powered laptop like a QL and inserting disks in the drive, booting from them, and running as before but then his experience of the powerful QPC2 program would be inadequate and it would be selling Marcel's work short.

So, I put in the time teaching him stuff, loading Xchange for him, showing him how to set things up before he left. He still has to jump the printer hurdle so I suppose I will hear from him again but he will, at least, be getting a little more from his QPC2 purchase. That is, of course, what we are about isn't it?

This, of course, brought up that old incompatibility issue again, only written very large and it is a barrier that many people who have been running old black boxes come across when they try to get their old programs and data onto an emulator. This applies to all emulators, not just QPC2. I was left wondering about the possibility of a piece of hardware to make things easier. Of course that is mainly pie in the sky now that Nasta does not seem to be making any progress with his projects but, just for a hypothetical exercise what could we do?

Thought Experiment

In the early days of Quantum Physics many of the leading scientists such as Neils Bohr, conducted what they called 'Thought Experiments' (a phrase thought up by Ernst Mach in the 19th Century) in which they tried to conjure up the results they would get if they were able to have the equipment they needed or were physically able to measure sub-atomic particles. In our case of transferring data from a black box with a faulty disk drive or, perhaps even a microdrive, what could we do?

Direct cable transfer is one option but Sernet has to be ruled out because it requires SMSQ/E and that will not run on a QL unless it has a minimum of a Gold Card. Simon Goodwin could probably help here because I do believe there was something in his D.I.Y toolkit which would allow serial transfer. One other answer would be QTPI connected to the Windows HyperTerminal program. This is a route worth considering but the user would have to make up the cables to do it and that is not so easy - especially since the QL style serial plugs are hard to get. Maybe SuperHermes and a special cable from Tony Firshman or a special DATA Transfer version of SuperHermes Lite?

Disk transfer would be OK if the disks are readable by the PC but, in the above case that is not possible and, in the case of a microdrive-only machine, the user would need to get a disk interface and set of drives that could be read - not easy these days.

Other hardware solutions are even more unlikely but, since this is a thought experiment we can launch into them. We already have the ROMDisq but that is still in QL format and cannot be directly read by a PC but how about a ROM to Compact flash adaptor? Compact Flash is the best way to do this because it is, at it's heart, an IDE device. the device would need an interface that would accept Input and Output, to and from the QL's ROM port and output to an IDE device. Almost the same idea that Miracle had many years ago except that they were dealing with MFM and RLL drives in those days and not IDE. We would also need some software that would create a QXL.WIN file on the Compact Flash or, if the user was able to format and create the appropriate file on his PC, just to write to it and read from it.

One other, even more unlikely idea, is a RomDisq to PC interface. I suppose this would have to be some kind of USB device and would need a lot of PC software and firmware to be written to allow the PC to see the device and make it available to the underlying emulator.

Finally, the most outlandish idea of all. A stand alone device that would accept a Compact Flash card and a ROMDisq. Push a button on the device to transfer DATA from one to the other.

Well, they are all thought experiments, and I have no idea of the feasibility of designing and building them. It would be nice if someone could take these subjects up and produce an article on getting data from a QL to an emulator. Maybe several people could write something and we could lump it all together as a part of the Start Here series.

Now, how about it Tony and Nasta? Get you designer's hats on and implement one of the hardware devices.

Microsoft Land

Digressing into Microsoft Land for a brief paragraph or so, the QL users list was recently invited to look at Tarquin Mill's website, **www.oft.me.uk**, in which he lampoons the Office of Fair Trading in the UK for not attacking Microsoft for being unfair. Now it does seem to me that there are issues to be advanced in this direction but the major problem is that, like the giant supermarkets, Microsoft has grown so huge that it does not matter how many Don Quixote wannabes launch themselves at its flanks, all it will do is belch and turn the other way slightly. That other great Behemoth, the EU took it on over the last few years like some Harryhausan Clash of the Titans epic. They wasted a colossal amount of taxpayers money forcing the company to provide other versions of its software which did not have the media player built in. I work in a company that sells Microsoft products and no one has ever asked for this copy of Windows. As a result of this legislative overkill the new version of its O/S, to be called Vista, will come in more than 8 different versions but will anyone make any great distinction between them? People still buy XP Professional because it has the word 'Professional' in its title but they are unlikely to use or, in some cases, even understand, why they forked out the extra cash over the 'Home' version.

More to the point here as the EU forces Microsoft into opening up its code to its rivals, in a different part of the building tries to foist its Intellectual Properties copyrights law onto the books which will almost make it illegal to think about writing a program with a similar concept to one already written. This kind of Carlos the Jackal and Mr Hyde, double standards baffles me. After all why is there only one European Parliament? should we not have a few different ones so the public can choose which laws they want and maybe buy those. Now there is an interesting political system to think about. True market forces at work.

Tarquin implies that Microsoft was somehow instrumental in the demise of the small computer systems of the 80's but they had a pretty good death wish themselves. Back then the systems were proprietory and inward looking. There were few attempts to get a coherent system of communication between rival systems, partly, I suspect, because they could not see the need for it and partly because they did not want anyone straying from the system they were using. Sinclair itself, was going through its oddball inventor stage, producing strange vehicles that ran on hair dryer motors and paying no attention to its core electronics market. It botched the launch of the QL as we all know and pretty much signed the systems death warrant at an early stage. Cuddly Alan Sugar, only slightly less appealing than a room full of wasps, did the rest. He bought out Sinclair to get his hands on the successful Spectrum and to bury the QL which he saw as a rival to his Amstrad business machines. As it was they both swirled down the toilet because IBM and then Microsoft had hit on the idea that offices all over the world would like to be able to share files and the, now global, dominance of Windows based systems began,

Tarquin also points the finger at Microsoft for holding LINUX back but LINUX did a pretty good job of that itself by making itself the geek's pin-up. It is not the great skill of the programming or the accessibility of the code that makes people rush out and buy the system. It is how pretty it looks, how well it is marketed and how successfully you can convince the people that they really need to enough bells and whistles to drown out a Football stadium in full voice. Oh yes, and where is 'Grand Theft Auto' for LINUX then?

It would seem to me to be pointless to rail, rant and tear your clothes in frustration over the wrongs, real or imagined, that you may think Microsoft has done to you. Why not put that energy into making your own system better? Put enough termites in the cellar and even the strongest foundation will collapse.

Fex and Friction

I think that Per Witte may have got the wrong end of the stick when I mentioned his use of the FEX keyword. I was not complaining about him but about the way in which the system for monitoring the progress and enhancements for SMSQ/E are implemented. The whole reason I fought tooth and nail to get the licence for SMSQ/E adopted was to keep it regulated so that conflicts did not happen. I also wanted to make sure that things that were written for one platform would either work on another platform or at least not cause a fatal error. This is a Herculean task and one not to be undertaken lightly and Wolfgang Lenerz has done a good job in trying to keep it in check and to make it work for the majority of cases. On this one occasion it fell down because no-one knew that FEX was used in other programs but, even then, it still worked until a new version of FileInfo 2 was

launched - correcting the FEX keyword. In my opinion this was the wrong approach. Once a keyword has been introduced and accepted it should never be changed for any reason. You never know who might have used that keyword or where it may cause a program to fall over. If this had just been an oversight it would have been excusable but it was not. It was recognised that FEX had been used before and it should have been rejected as a new keyword and mentioned in the SMSQ/E documentation.

It is not Per's fault but it is a 'heads up' for the way in which SMSQ/E is regulated. You never can tell where these things are used so avoid duplication in the future.

It was used in Disk Mate 5, an underestimated file handling program which sees a lot of use here at QBranch HQ. It was because I use Disk Mate 5 a lot that I noticed it when Thierry changed things in the new version of FileIfo 2 and dropped FEX from the extension that is loaded at the start. You can actually re-insert the FEX keyword by a change in the config block of FileInfo 2 but that, of course, will mean that future programs using the new version of FEX will fall over. Can someone hack DM5 to change the command? That would solve the problem this time but a lesson should be learned here. Compatibility is more important than innovation.

It is interesting to note that after Per put his hands up and said he was the guilty party both Dilwyn and Francois van Emelen started discussing lists of keywords on the QL Users lists. I have often wondered who reads the stuff I write and, if all I have managed is to get this discussion started, it is worth all the hours I put into it.

Another Issue

I have been asking readers of this magazine if they would rather it were on time and maybe a little thin because articles were late or had a reasonable standard of content and came out a little late. People seemed overwhelmingly for the former position because it did keep the 'News' section as real news as well as being predictable in its publication time. Jochen and I discussed this and decided that, from this issue on, we would set a publication date and publish that at the bottom of the last page where the shows are announced. That way you should all know when to expect it to arrive.

Honourable Mentions and Flags at Half Mast

I mentioned at the start of this about the very beginnings of QBranch, over 10 years ago now, and it was shortly after writing those words that I got the news about the death of Steve Hall who was QBranch's co-founder. I have written a formal obituary which appears elsewhere in the magazine but I also think it is fitting that I should give him an honourable mention here too. Without his help and enthusiasm in the early days we might not have got off to the start we did. He was present through most of the early days of the company, taking over the reins when I was away on my Rock'n Roll jaunts. Later, when I was running the Bank Volt, he got involved there too spending many late hours building systems and sifting through the second user hardware we accumulated. Neither of us made much from all that work but he never complained.

I attended his funeral and it was typical of him to have the Monty Python song 'Always Look On The Bright Side Of Life' playing as we entered the crematorium and to have the 'Thunderbirds are GO; countdown played as his coffin set off to the flames. In the addresses that people gave they all mentioned his indefatigability and how he would always see a problem as a challenge. I was reminded of him strongly the evening of the funeral when the QL User I mentioned above, who had a particularly ancient set of drives and interface called round. He had bought QPC2 but the disks, which were perfectly readable on his QL were impossible to read on his PC. As I sat there trying to fit a modern drive to one of his older ones so he could transfer the data to that, and struggling to overcome cables that were the wrong way round and power lines that were too short I realised that it was that need to make things work that we had in common although, on the practical side of things, he was always a craftsman, whereas I would often bodge it to work.

Above all he was a friend and good fun to be with. In the 12 years I knew him we never fell out once and, if we weren't debating politics or computer hardware and software we were often laughing. I took it upon myself to call round the Sussex QL User group and inform them of his death and they all had similar feelings about him. You may not really know some of the things he got started or some of the ideas that he put into the community because he tended not to be so outspoken about himself, but he deserveres this final salute from myself, the Sussex QL Users Group and from the many customers whose QL problems he solved and to whom he gave good advice.

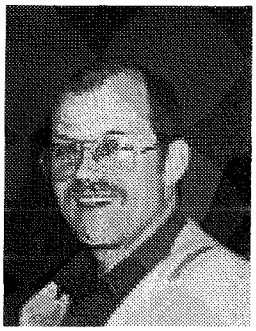
Bye my friend.

Obituaries

Stephen Hall (1955-2006)

News has just reached me of the sudden death by heart attack of Stephen Hall aged 50. Some years ago, when I was about to move from Hamburg, in Germany, back to the UK, I put a letter in Quanta asking if there were any QL users in Brighton. Steve responded to that and we met up soon after I arrived here. We became firm friends and remained so until his death. Steve was one of the pivotal figures, along with Keith Mitchell, in the resurrection of the Sussex QL Users Sub Group and was my partner in QBranch from the start. In fact, he gave the name to the company in a drive along the Worthing seafront to meet up with John Wakefield for an interview for a QL publication.

Steve was always a good person to spend time with and we had many enjoyable times together travelling to QL shows in the 90's. Although we were diametrically opposed politically we were always able to have good-natured and intelligent discussions which passed the time on the road enjoyably. As far as the QL went he was instrumental in getting Rich Mellor to write QRoute and in getting MasterSpy and some other programs available. Although his interest waned over the last few years we remained good friends and continued to stay in contact. He taught me a lot about the QL and, like everything else he did in his life, he approached every project with intelligence and understanding.



He leaves a wife and two teenage sons. I, for one, will miss him.
Roy Wood.

Dietrich Buder (1932-2006)

Many QLers around the world will remember Dietrich Buder, as he visited many QL shows, even the ones in the USA and Canada.

I heard about his illness when Dietrich had to cancel one of the regular visits of the US QL shows some years ago. Luckily, he recovered as the medicine seemed to have helped. So he came to the show the next year and to some shows in Europe, but in the end his illness won.

I have known Dietrich for many years as he was an enthusiastic QLer right from the start. He was not just a customer, we became friends, and especially when we travelled together with his wife and my girlfriend. We spent several days every holiday together - very enjoyable times.

Dietrich was also the helping hand for the German QL Today, especially proof reading. I heard at his funeral, that he was very encouraged in the German language, the new spelling etc. and even founds "bugs" in the main German dictionary.

Dietrich has always been very interested in technical things, and not just computing. Music, model trains and travelling are part of the list.

From what I heard, Dietrich had an interesting life all the time, before and while I knew him. He travelled a lot, and he continued to travel as long as his illness allowed it.

Especially the EXPO 2000 (World Exhibition) in Hannover, very close to where he lived, allowed him to visit the remaining parts of the world without driving or flying. He spent nearly every day at the EXPO.

Dietrich liked the USA and found several friends over there as well. He also liked the fact that smoking was strictly forbidden in public places and restaurants and he hoped that this would become fact here as well. We discovered that he recently made a statement in public, which you can see and hear yourself - just google for "Dietrich Buder" to find the video.

There are so many things, both QL-related and not QL-related, which will always remind me of Dietrich himself or the time we spent with him. I am glad I knew him. Jochen Merz

The QL Show Agenda

The Quanta 2-day Workshop & AGM

Venue: 3rd Davyhulme Scout Headquarters, Conway Road, Davyhulme, Manchester, UK

Sat., 8th April 12pm - 5pm and Sun., 9th April at 10am to 4pm

The AGM will be held at 2pm Sunday April 9th 2006

Hotel accommodation is in short supply that weekend as there is a Man U/Arsenal football match on Sunday. Our usual hotels are already full. There are vacancies at the Mersey Farm Premier Lodge in Sale, Tel No 08701 977 179

Directions to the venue by Road; From the M60, Leave at Lostock Junction 9, and take Lostock Road, towards Urmston. Pass the George H Carnall sports centre and take the next turn to the right, Conway Road. 3rd Davyhulme HQ is at the bottom of the road on the right hand side.

We hope to arrange a programme of presentations and demonstrations for the Sunday before the AGM starts. If you have any topics that you would like to know more about, please contact John&Sarah Gilpin: theGilpins@btinternet.com

The Hove Quanta Workshop

Portslade Townhall, UK - Sunday, 28th of May 2006

This will be the 10th Birthday of the Annual Hove Quanta Workshop. After a couple of years in very unsatisfactory Hotels we settled upon the Portslade Town Hall venue which has proved to be very popular with both visitors and traders. As usual the catering will be provided by our usual bunch of lovely ladies and we hope that the weather will be good enough to encourage those of you not from the area to come down for a weekend by the sea. Hotels do get booked up so do plan ahead!!

I am looking for people who would like to give a talk, or demonstration during the show. Any suggestions or volunteers should contact me via Q Branch.

The Next Issue

We plan to have the next issue ready for you at the Hove Quanta workshop, 28th of May. If you cannot pick it up there, then your issue will be on its way to you right after the show.

And to ensure the issue can be ready, please send your contributions for the next issue to us before the deadline, 1st of May.