

Q. How can I rename the medium name of some of my QXL.WIN files to something more meaningful than something like 'WIN2'? As I keep my fonts, documents and graphics in separate QXL.WIN files, and reassign the various WIN drives depending on which I need to use, it would be handy if a name such as FONTS or DOCUMENTS or GRAPHICS could appear when I use a DIR command.

A. Upon receiving this query, I set about writing a program called WINDOS (see News column) to do this very task.

After writing it, I realised I was reinventing the wheel a little, since Per Witte has also written a program called QWIRC (QL Winchester Rename Console) which can do similar functions in a smaller display outline. Both programs are available to download from my website, and Qwirc v0.66 is also available from the Marcel Kilgus website www.kilgus.net/qpc/downloads.html

I'll try to explain here how you can write your own code to achieve this, although be warned - this involves direct sector access to write directly to the control structures of a QXL.WIN and it is VERY easy to destroy a QXL.WIN by making simple mistakes, so NEVER try this without making a backup copy of the QXL.WIN first. This is easy with a QXL.WIN, just create a folder in Windows called something like C:and right-drag a copy of the QXL.WIN into there (hold down the right mouse button in Windows while dragging the selected QXL.WIN until the name being dragged is over the backup folder and let go of the button, a small Windows menu appears asking if you want to Move or Copy the QXL.WIN here, select COPY of course).

When you use a FORMAT command to create a QXL.WIN file, it only allows you to specify a capacity figure, not a name. It automatically formats the device with a name based on the name WIN and drive number, so if it was WIN1_, it would usually end up with a name like 'WIN1'.

The QXL.WIN medium name is held in the first sector of a QXL.WIN file on the PC hard disk. To know how to change the name, we need to know the format of the relevant part of this first sector of the QXL.WIN.

The first four bytes hold the letters "QLWA". This is followed by a word value, holding the length of the medium name, followed by up to 20 bytes of a medium name.

DC.B	"QLWA"	
DS.W	1	;name_length
DS.B	20	;characters of name

This is the part we need to modify and to do this we need to use direct sector access to fetch a copy of the first sector. Then we patch the name length and characters as required. The name, although it has a word length count, can only be up to 20 characters long. Indeed, as a DIR command only shows the first ten, and even the DMEDIUM_NAME\$ function also only fetches up to 10 characters, it is probably more practical to limit the name to a maximum of ten characters.

Direct Sector Access on the QL is normally used to read or write specified sectors of a floppy disk or hard disk. Even if you know what you are doing, it is only too easy to accidentally destroy the medium so always make a backup copy first!

To use direct sector access, we open a special file on a disk or hard disk, called ""d2d". The letter d at the end can be d, h, or e depending on the density of the disk (h or H stand for High Density, while e or E stand for Extra High Density or an ED disk. For a QXL.WIN we typically use ""d2d".

The '2' stands for 512 byte sectors, and would be changed to '3' for 1024 byte sectors and 4 for 2048 byte sectors.

First, we open a channel to the QXL.WIN (which we'll assume in this case is a temporary test copy set up as WIN8_).

```
OPEN #3,'win8_*d2d'
```

Note that since it contains a '*' character the filename needs to be in single or double quotes.

Next we fetch the first sector:

```
GET #3,sector$
```

We use GET rather than INPUT since the sector may contain linefeed characters which might prevent INPUT getting the entire sector, even if INPUT knew where to stop reading the sector!

The string called sector\$ now contains a copy of the first sector of the QXL.WIN. By looking at the 5th and 6th character of the string, we can see how long the name will be - as it can not be longer than 20 characters we can disregard the high byte and only look at the sixth character:

```
namelen = CODE(sector$(6))
```

If you want to be stricter about this you could evaluate the full word value:

```
namelen = (256*CODE(sector$(5)))+CODE(sector$(6))
```

Now, we can extract the existing name, starting from the seventh character of sector\$:

```
name$ = sector$(7 to 7+namelen-1)  
PRINT name$
```

We can check this using the DMEDIUM_NAME\$ function, which takes either a channel number of an open file or a backslashed drive name as its parameter:

Note that the channel number opened above should be closed before using DMEDIUM_NAME\$.

```
PRINT DMEDIUM_NAME$(\win8_)
```

This should print the same name as held in name\$ above. What it may do differently is only show the first ten characters if the name happens to be longer than ten characters in length.

So, to change the name, what we need to do is to alter the name details within sector\$, update the name length part, and write it back out to disk to replace the original value, by resetting the file pointer to the start of the file before writing a new updated copy of sector\$. Let us change the medium name of Win8_ to 'QUANTA'

```
LET new_name$ = 'QUANTA'  
LET namelen = LEN(new_name$)  
LET sector$(6) = CHR$(namelen)  
LET sector$(7 TO 7+namelen-1) = new_name$  
PUT #3\0,sector$  
CLOSE #3
```

The "\0" after #3 in the PUT command simply resets the file pointer to the start of the file, to make sure sector\$ is written out to the correct location. And we must remember to close the channel afterward!!!

After closing the '*d2d' file, we can check if we were successful simply by issuing a DIR WIN8_

QUANTA QDOS

1234/20480 sectors

Hopefully, if you were using a Quanta Library CD copied onto your hard drive, it will now be called QUANTA to help you identify it!

I can't emphasise enough that we are writing to the actual structure of a QXL.WIN here and a mistake such as forgetting to position the file pointer correctly may well result in the destruction of your QXL.WIN - always make sure you keep a backup copy before embarking on this! After all, we write to the wrong place in a document, we damage that one document. We write to the wrong place in a QXL.WIN we may well damage all your files beyond repair!

Q. We keep getting told to make backup copies of our files. It's easy enough to copy floppy disks. Is there a proper backup program for QL hard disk systems?

A. Define "proper"! That said, there are plenty of hard disk backup programs for a QL available from my website at www.dilwyn.me.uk/backup/index.html

Some so-called backup programs are actually whole-disk copiers (e.g. my program called The Copier). More advanced copiers such as Norman Dunbar's Winback will look at the file dates to see if the backup medium already contains an up-to-date copy, so we don't need to make an extra copy, we can save time by skipping files which haven't been altered since the backup was made. This is called an Incremental Backup, so called because we only make backups by increment, i.e. updating the backup only when a new file is discovered or when we detect that a file has been changed since it was originally backed up. In other words, if there's already a suitable backup, don't bother trying to make another. Why copy all 1,000 files when only two or three have changed?

To some extent, the QPac 2 file menu can do this. It has both a COPY and BACKUP command. The difference is that BACKUP will only copy a file if there is a more recent version in the destination directory. Fairly basic backup, but sometimes good enough.

Of course, if your "QL" is an emulator which uses a QXL.WIN it is even easier than this. Just use Windows to copy the QXL.WIN, since a QXL.WIN is just one single very large file as far as Windows is concerned. So to make a complete backup in one step, just copy and paste the QXL.WIN to make a backup copy in another folder or better still on an external hard disk drive!

If you use an emulator which uses a native file access mode to store QL files on the computer's native format hard disk, make sure you copy the entire folder, not just the individual QL files, as some emulators store the QL file headers in a separate (sometimes hidden) file within the master folder. Loss of these file header information blocks can mean that executable programs, for example, may suddenly start giving "bad parameter" errors when you try to execute them. It is then a bit of an expert job to try to restore these file headers.

It is always better to make a backup onto a physically separate medium and ideally store it in a different location. That way, if some major issue such as theft or fire occurs, you may have lost your computer but still have your files, which are usually harder to replace than a computer!

Q. Is there an email program for the QL?

A. Almost but not quite is the true answer to that question. Several attempts have been made and for one reason or another none came to full fruition. The main problems have been writing a suitable database to store the emails and writing a suitable editor for writing emails.

That said, Jonathan Hudson did write a fairly simple to use email program - the program and C sources are available to download from his website at www.daria.co.uk/qdos/ - just search for QDOS Email and QPOP3 on his website.

Receiving emails is actually simpler than you might think. Here's a small and fairly basic listing from Jon Dent, slightly updated by me, which shows how to fetch emails using the POP3 (POP=Post Office Protocol, another standard is called IMAP or Internet Access Message Protocol) on a QL system with TCP/IP facilities, such as QemuLator, QPC2 and uQLx. In line 100 you should put your email provider's mail receive details after tcp_, put your email user name in line 210 after the word USER and your email password in line 270 after the word PASS. Note that the listing is for SBASIC, not SuperBASIC, as it uses un-named REPEAT loops.

Line 100 opens a TCP channel to your email provider using something called port 110. Lines 140 and 150 handle the characters sent back by the email system, while line 210 provides it with your user name details and line 270 your password details. Line 330 asks the email system for a list of emails ready to be sent to you, and stage 3 from line 360 asks which emails to receive the top 5 lines of. If you enter 0 it moves on to stage 4 to close down the TCP channel opened earlier.

This is far from a working email program of course, it merely illustrates the basic concept of how relatively easy it can be to read emails. The harder part is decoding the email headers to extract subject, to, from, reply to details and so on plus of course displaying the text.

Composing emails is a bit harder, you need an editor program to write the email, plus some code to build the header and reply details and any attachments for example. We can achieve most of this by using a plain text editor for the body of the text and suitable code to add the rest.

```
100 OPEN_IN #8,"tcp_mail.isp.net:110": REM your account: port 110
110 inst$="": stage=0
120 crlf$= CHR$(13) & CHR$(10)
130 REPEAT
140 a$= INKEY$(#8,100)
150 IF a$<>"":inst$= inst$ & a$
160 PRINT a$;
170 SElect ON stage
180 =0:
190 IF "OK" INSTR inst$ THEN
200 inst$= ""
210 PRINT #8, "USER your.username";crlf$;
220 stage= 1
230 END IF
240 =1:
250 IF "OK" INSTR inst$ THEN
260 inst$= ""
270 PRINT #8, "PASS your_email_password";crlf$;
280 stage= 2
290 END IF
300 =2:
310 IF "OK" INSTR inst$ THEN
320 inst$= ""
330 PRINT #8, "LIST";crlf$;
340 stage= 3
350 END IF
360 =3:
370 IF "OK" INSTR inst$ THEN
```

```

380     PRINT #0,"view which mail number ? 0 to QUIT":
390     INPUT #0,number$
400     IF number$<>0
410         PRINT #8,"TOP ";number$;" 5";crLf$;
420     ELSE
430         PRINT #8,"QUIT";crLf$; : EXIT
440     END IF
450     CLS:CLS #0
460     stage= 2
470     END IF
480 END SElect
490 END REpeat
500 CLOSE #8 : REMark close TCP channel

```

Figure 1. Jon Dent's bare bones email program listing

Want to send an email quickly and easily from QPC2? As long as you have an email program set up on your PC, the QPC_EXEC command can work a bit for its living. Here's how to send a Helpline query to Quanta from within QPC, using a QPC_EXEC command to send a mailto: instruction to your default Windows email program:

```
QPC_EXEC "mailto:helpline@quanta.org.uk?subject=A very difficult question"
```

QPC_EXEC sends the command to Windows, which starts the default email client to generate a message to Quanta Helpline with the subject line 'A very difficult question'

If you'd like to tinker with setting up an email receive utility, probably the easiest way to set up the database is to number the emails with an incremental number as they are received from the ISP and give them an identifiable filename extension such as _txt or _eml. That way, you can quickly and easily build up a table of emails by listing files with that extension. As you read the emails into your "database" for display, you'll need to create a list with details such as the sender, subject and so on to fill in whatever columns you wish to display.

Documents describing email formats, email headers and so on are available to download from the world wide web consortium website at www.w3.org - they are not exactly bedtime reading material but they are the definitive documents. They are mostly RFC (Request For Comment) documents, but they do contain good information. If you can pick out the relevant details, they show how to send and receive short email commands to and from the email system, and the structure of the email headers, which helps you decode the to, from, reply etc details if you'd like to try your hand at writing an email handler!