

**Helpline – Dilwyn Jones**

**Transferring Files Between QLs And Emulators**

A recurrent source of questions at QL Is 25 was how to transfer files between a QL and a QL emulator on a PC. This is a direct result of the number of people moving to using an emulator, plus a number of people who have gone back to using QDOS after a number of years of not using a QL. The QL's strong web presence manages to attract some ex-users back into the fold and they are mainly using one of the Windows QL emulators, QPC2, Q-emuLator, or QL2K/QLay.

So I decided to dedicate a couple of Helpline columns to this subject. I am fortunate to have received an article about serial port file transfer from a QL direct into Q-emuLator from John Nunez to get all this started, which I have reproduced below, with permission from him.

File transfer between QL and emulators can be broadly categorised as follows:

1. Serial cable file transfer on systems supporting serial ports.
2. Using floppy disks, where the QL system has a disk drive and the target system can read QL floppy disk programs.
3. Using QL disk readers in the target environment (e.g. a program like QLTools in DOS or Windows)
4. Specifically in the case of QLAY, QLAY2 or QL2K, the use of the specially written tools programs QLAYT and QLtoolsq to move files between environments. I intend to cover this in a future article, but in the meantime have a look at an article in QL World's review of QLAY 2, which included some details of how to use the qltoolsq and QLAYT tools programs to transfer files both between floppy disk using QLtoolsq and between the Windows hard disk and the native QLAY hard disk filing systems using QLAYT. The article was called "My Holiday Romance" and appeared in the December 2007 (Volume 12 Issue 2), on pages 42 to 51. The part of the article about the tools programs appears on pages 48 and 50.
5. The use of the Sernet software to network computers which have serial ports.
6. Understanding the file headers situation, where the loss of the executable program file header prevents the program running properly after transfer.
7. The use of Zip and Unzip programs to package a number of files together for transfer and to preserve file headers. How to get the Zip/Unzip executables over to the target system in the first place, either by manually "restoring" the executable file header, or by using Jonathan Hudson's self-extracting archive.
8. The use of the QL network to transfer files to a QXL or Aurora system, where one of the machines does not have floppy disk drives.
9. The advantages of having Toolkit 2 on your system and the advantages of the improved co-processor chips like Hermes in improving reliability of the QL serial ports.
10. The use of the DOS device driver in QPC2.
11. The use of terminal software on both source and target computers to transfer files, such as QTPI on QDOS/SMSQ systems and Hyperterminal in Windows. This is a subject I have never tackled and would be very grateful for any contributions about it.

I'll start with the use of the serial ports to transfer files directly from the QL to another machine. This subject came up on the QL Users mailing list recently and after a flurry of discussion between John and people with experience of this, John kindly wrote up the details of how he had approached and solved the problems. John uses Q-emuLator, which supports the use of the PC's serial ports as though they were QL serial ports, so his article is mainly about direct QL to Q-emuLator file transfer via the serial interface. As QPC2 also supports serial ports, most of this article should be applicable to QPC2 as well.

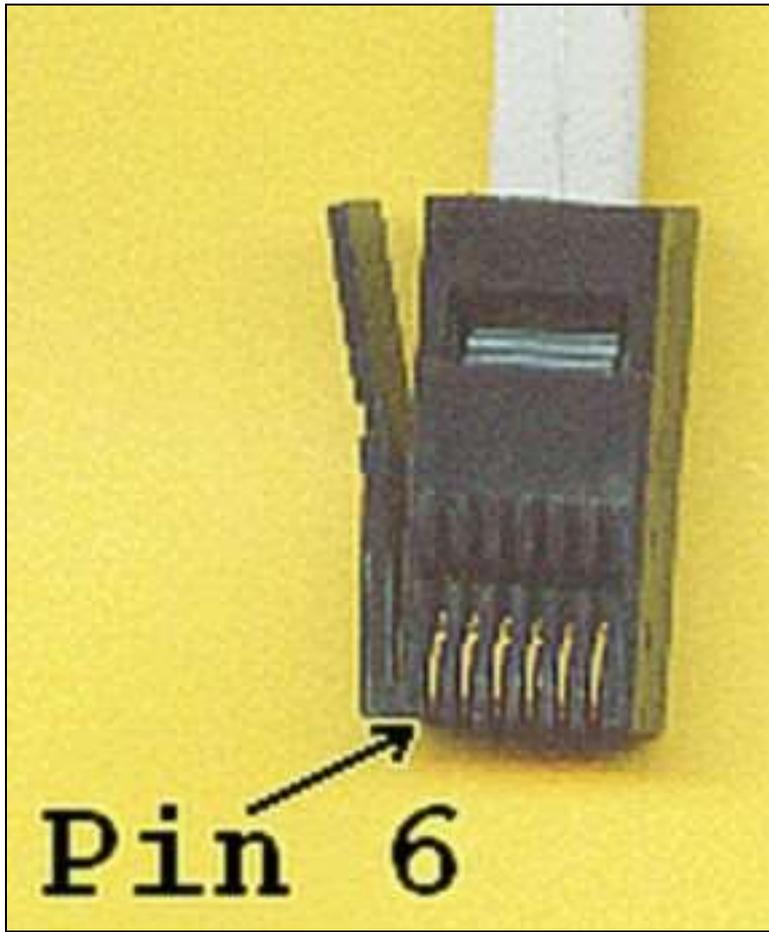
Once you have the required cable for serial port transfers, it is fairly straightforward to copy text files and other fairly straightforward data files with simple copy commands on the QL and in a DOS command line in Windows. Bearing in mind that most versions of the QL have a unique type of connector, a 6-pin plug which looks a bit like a British telephone connector, but is not compatible with it, so you cannot use a telephone extension to create a serial lead, for example, it might be best to ask someone like Tony Firshman to make a suitable lead for you. John has included details in his article of where he got details of a suitable serial lead from – follow the link to Sinclair Shop he includes in his article below.

There is a more detailed article on the subject of serial port linkages on my website at <http://www.dilwyn.me.uk/gen/serial/serial.html>

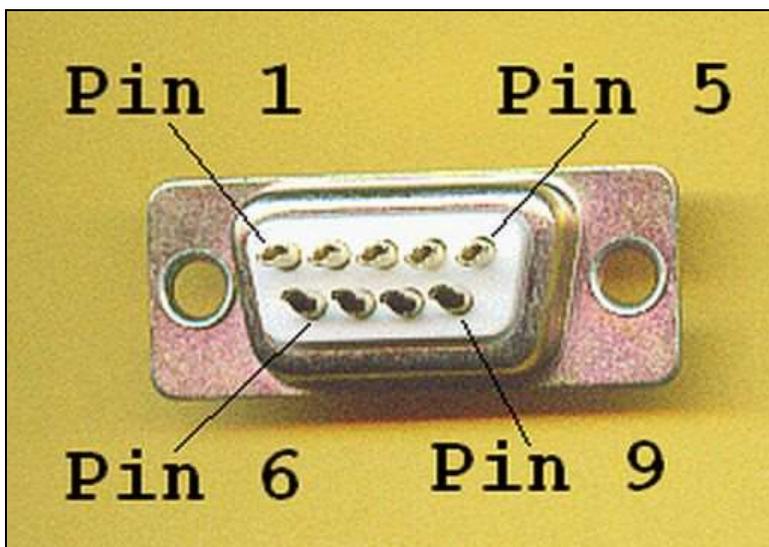
It's best to use serial port SER2 for QL to PC communication if you are making your own cable, as wiring is a bit easier. SER1 is basically wired as though it was a modem (called a DCE or Data Communications Equipment), with somewhat reversed wiring. SER2 is configured to be seen as a computer (called a DTE or Data Terminal Equipment). If you do use SER1, simply swap pin 2 and 3 over, and swap pin 4 and 5 over.

<b>QL SER1</b>	<b>QL SER2</b>	<b>QL CABLE COLOUR</b>	<b>DIRECTION</b>	<b>PC SIGNAL</b>	<b>PC 25-PIN D</b>	<b>PC 9-PIN D</b>
pin 3 RxD	pin 2 TxD	white	-->	RxD	pin 3	pin 2
pin 2 TxD	pin 3 RxD	green	<--	TxD	pin 2	pin 3
pin 5 CTS	pin 4 DTR (=RTS)	blue	-->	CTS	pin 5	pin 8
pin 4 DTR (=RTS)	pin 5 CTS	red	<--	RTS	pin 4	pin 7
pin 1	pin 1	black	---	GND	pin 7	pin 5
N/A	N/A	-	-	DSR	pin 6	pin 6
N/A	N/A	-	-	DTR	pin 20	pin 4

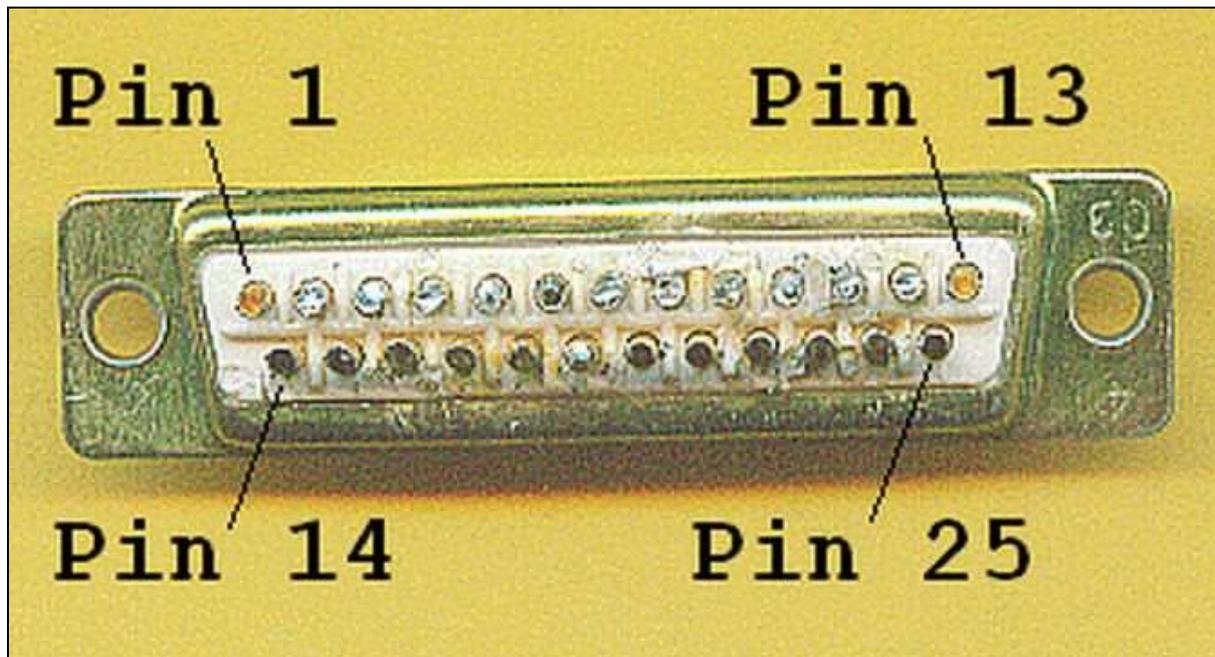
## **QL Connector Pin Numbers**



**9-Pin (DB-9) Cable Socket Pin Numbers (viewed from solder terminals)**



**25-Pin (DB-25) Cable Socket Pin Numbers (viewed from solder terminals)**



It's important that the handshaking lines are connected correctly, otherwise transmission may fail completely, or it may only work at very low baud rates.

Once you have a suitable cable, transferring a data file from QL to PC is as simple as this. Set a suitable baud rate at both ends. 9,600 baud is probably a good starting point, since this is the fastest speed at which a QL is likely to work (a PC can generally handle any baudrate the QL is likely to be able to throw at it, since the QL is unlikely to work at anything higher than 9,600 baud, in some cases it may struggle even with this baudrate, although the Hermes chips from TF Services can improve QL serial port performance to some degree). Set the baudrate on the QL with BAUD 9600 for example. The PC's baudrate can be set in the Control Panel / System / Hardware / Device Manager / Ports (COM & LPT) menu in Windows – you may find this is the default offered on some PCs.

In Windows, go to a DOS command line, e.g. from the RUN command in the start menu, enter the “program” name cmd, i.e. run cmd

In the command line, enter this command to copy a file coming in from the PC's serial port to a given filename. Note that there should be spaces between the three parts of the command.  
copy com1 filename

If you wish you can add a switch between the source name (com1) and the destination name (filename in the example) to specify that the file being copied is an ASCII text file or a pure binary file. The switches are:

Copy com1 /A filename

tells the PC that the file being copied from com1 is an ascii text file and the PC will handle it accordingly.

Copy com1 /B filename

tells the PC that the file being copied from com1 is a binary file and the PC will attempt to handle it accordingly.

Com1 should be changed to whichever COM port is being used, and filename is the filename you want the PC file to have, something like C:\qlfiles\test.txt

If transferring text files, another DOS command you can use is:  
Type com1 >filename

Where the >filename means that what would normally be screen output is sent to a file instead.

On the QL side, enter a command like:  
COPY\_N mdv1\_filename TO ser1hz

At this point there's two QL-specific issues to check:

1. The device name suffix, e.g. the hz in ser1hz
2. The difference between the various versions of the QL's COPY commands

The device name suffix takes a number of optional parameters such as parity, where e/o/m/s stand for even, odd, mark or space respectively. Don't worry too much about this for now.

Handshaking can be either i (ignore) or h (handshake). Normally, for file transfers, it is best to use h (use handshaking) to maximise reliability.

The protocol parameter can be r, z, or c. Each has its uses:

r – raw data, no EOF indication is sent (use for binary files)

z – CTRL Z (CHR\$(26)) is used to indicate end of file (also known as EOF), normally used for simple data file or ASCII text file transfer

c – this is like z but converts end of line characters between the QL's linefeed or CHR\$(10) and a PC's carriage return or CHR\$(13)

SMSQ/E serial port names can have additional port name qualifiers, but we'll ignore these for now (see under Serial IO Devices in the SMSQ/E manual).

## COPY

The standard QL has two versions of the COPY command:

COPY\_N copies the file without a header.

COPY will copy the file with or without a header depending on whether it thinks the destination is a directory device or not.

In addition there is a COPY\_H command added by Toolkit 2 which (in theory anyway) will ALWAYS copy a file with a header.

Some experimentation is needed here to find the combination of commands best suited to the type of file you need to transfer. Two problems can arise:

1. End of file. If transferring a binary file, e.g. machine code or an executable program, there is a chance it may contain a CHR\$(26) or CTRL-z. This indicates end-of-file to

the PC and transfer may stop at that point, leaving you with a truncated file, with parts of it missing. In this case, try using the binary non-EOF commands.

2. Transfers which don't end automatically. You may find that once all the file is transferred, one or both computers may 'hang'. The computer(s) concerned are waiting for an end of file signal. You may be able to manually abort and break out of this transmission stalemate using CTRL-space on the QL, or CTRL-C at the PC end. You can then compare file lengths to make sure that the file transfer was complete.

Don't get too discouraged if you struggle to get serial port links working, or they only work slowly. I remember only too well how I struggled with my early efforts and how many times I pestered helpful people like Tony Firshman for help. Serial transfers can work first time and work well, but if you mis-wire a cable, for example, it can be a source of great frustration. Very occasionally you will encounter machines that simply refuse to talk to one another for no obvious reason. I used to have a laptop which just would not talk to a QL at all, yet the same cable used on other PCs would work perfectly satisfactorily.

## **QL To Q-emuLator Serial Transfer**

I am grateful to John Nunez for permission to print the following article, which describes how to go about transferring files from a QL direct to Q-emuLator, using the QL serial port and the Q-emuLator using the PC's serial port. This is a very quick and useful way to transfer files directly from a QL into the emulator without having to resort to fiddly floppy disk transfers, or copy from QL to Windows and then have to worry about how to get the files from Windows into the emulator.

## **QL To QemuLator Via A Serial Link**

### **John Nunez**

Here is what I did; I wrote it so that hopefully even someone in my situation (i.e. pretty clueless on QL tech aspects) would be able to get results. I am not a regular QL user; but I am very fond of my QL, and occasionally fire it up to remember how things were. It was the first *real* computer I ever owned.

Just as a reminder, I have a QL that I haven't used since the late eighties when I was a teenager. It is a plain QL with no fancy bits and pieces that have come along since, like floppy drives, toolkit II's, gold cards, etc. I also have about thirty microdrives with old files I'd created as well as some commercial games with considerable sentimental value to me. A lot of these cartridges are on their last legs, and I wanted to get the files off and onto a PC where they will be safe, and I can run them on an emulator; I can't say that I miss that QL keyboard! And I wanted to do this using just a serial cable; what followed was hours and hours of frustration (well, if I'm honest, I actually quite enjoyed it, as all those QL memories came flooding back...).

I will not cover the usage of QemuLator as it already includes excellent documentation.

There are three steps.

1. Connecting the hardware
2. Getting software onto the QL that will allow us to copy files back to the PC (i.e. QemuLator) without losing header information.
3. Copying the actual files from the QL to the PC.

## 1. Hardware setup

This is what I used:

- My original QL, microdrive cartridges, etc
- Windows XP laptop with QemuLator v.2.5 installed
- Serial cable to link the two together (like the one here: [www.thesinclairshop.com/PDF/sinclairserialadaptor.pdf](http://www.thesinclairshop.com/PDF/sinclairserialadaptor.pdf) )

I did not have to change anything with com port settings on the PC, or change any of the defaults on QemuLator. My laptop has one physical COM port which is the usual DB9 male connection and is seen as COM1 by XP.

I connected one end of the cable to the PC's COM port, and the other to the SER1 port on the QL.

Connectivity can be tested by typing at QemuLator:

```
copy ser1 to scr_
```

the QL's flashing cursor will disappear as the command runs and waits for data on the SER1 port.

On the QL, type the following

```
open #4,ser1  
print #4,"Test"
```

You should see the word "Test" appear in QemuLator

still on the QL, type

```
close #4
```

In QemuLator, press ctrl-space, the cursor should reappear. A "not complete" message also appears (this message will appear later when we transfer files and press ctrl-space, but can be ignored as the files DO transfer correctly).

## 2. Getting the program to use for copying over to the QL

The first thing we need to do is copy the MdvToWin program (which is supplied with QemuLator) from the PC to the QL and fix it after the transfer so that it works (i.e. restore the file header). This program can then be used to transfer files from the QL back to QemuLator, correctly preserving them.

So first, we set up an mdv1\_ on QemuLator which holds the MdvToWin\_exe file. On the QL, place a writeable microdrive cartridge with free space in mdv1\_

To copy from the QL to QemuLator I did not have any problems with defaults (I believe the default rate is 9600 bps); however, I could not copy TO the QL if I didn't bring this down to 1200, so on both the QL and QemuLator, type

```
BAUD 1200
```

(there is no confirmation message from this command)

On QemuLator, type

```
copy mdv1_MdvToWin_exe to ser1
```

(the cursor stops flashing)

on the QL, type

```
copy ser1 to mdv1_MdvToWin_exe
```

the microdrive will whirr into life as the file is transferred. You will know when the file transfer has completed, because a cursor will start flashing on QemuLator, and then the QL microdrive will stop; at this point hit ctrl-space on the QL.

We now have the MdvToWin\_exe on mdv1\_ on the QL, but it won't run, because its header has been lost, so it needs to be recovered.

The first thing we need to know is the length of the mdvtowin\_exe file in bytes. We can get this by right-clicking on the file in Windows and bringing up its properties. For the version I used, it was 1058 bytes.

[If you have Toolkit 2 on the QemuLator or QL system, you can also use the FLEN extension like this from Basic:

```
PRINT FLEN(\mdv1_MdvToWin_exe)  
-DJ]
```

Taking this number, on the QL, we type

```
LET addr = respr(1058)
```

we then take these two numbers and on the QL we type

```
lbytes mdv1_MdvToWin_exe,addr  
sexec mdv1_mdvtowin,addr,1058,512
```

The mdvtowin file is the new file which WILL execute correctly, and is saved to mdv1\_ by the above command. I don't know what the 512 is, but (for me at least), it all works.

[it's called "dataspace" - the amount of memory a program needs for its data area when it is running -DJ]

On the QL, type

```
exec_w mdv1_mdvtowin
```

and you should get a program running with a title of MDV -> Q-emulator for Windows 95

We can now use this executable to transfer files from the QL to the PC (QemuLator)

3. Now to copy a file from a QL microdrive to QemuLator.

Restart the QL and the QemuLator. This is to set the baud rate back to 9600, and get everything to a standard state.

As an example I have a file on mdv1\_ on the QL called index\_dbf, which I want to copy across to the PC. In mdv2\_ on the QL I have the mdvtowin file I "created" earlier. In QemuLator, I have a (virtual) microdrive, mdv2\_ set up and mapped to a folder on the underlying PC

Connect the QL the PC, and power both on. On the QL:

```
exec_w mdv2_mdvtowin
```

The mdvtowin program executes and asks for the source file, give it

```
mdv1_index_dbf
```

the program scans the file, once finished it then prompts for the destination, enter

```
ser1
```

at this point the data is sent down the serial port and the QL no longer shows any cursor activity, as it waits for the data to be accepted from the other end, so...

On QemuLator enter

```
copy ser1 to mdv2_index_dbf
```

the mdv on QemuLator will light as data is copied to it and the file is created. Once the transfer completes, on the QL there will again be a flashing cursor. Once the copy finishes, i.e. flashing cursor on QL, AND mdv "light" extinguishes on QemuLator, press ctrl-space on QemuLator. The mdvtowin executable also reports a successful transfer.

The file is now accessible to QemuLator.

This can be repeated as many times as required to get as many files copied as possible. I successfully transferred the Psion Chess program from the original microdrive using this method, and can run it on QemuLator now.

\*\*\*\*\* THE END \*\*\*\*\*

I hope the above is all correct. I've checked it all, but even with the best of intentions, I tend to find that mistakes always seem to sneak in somehow.

And once again, thanks to everyone who responded so promptly with help and advice. It's nice to see that there's a community based around the QL that's still going strong, even now.