

Members wishing to submit helpline requests via email can use the email address helpline@quanta.org.uk or if you prefer to use traditional post, please send the helpline request to me via the address printed inside the front cover of the newsletter.

Obviously, we cannot guarantee to answer every query we receive, but we will do our best! Where we have been unable to answer the queries, we may print the help request as an open request in the newsletter to ask if any of the readers can come up with a solution. And, of course, if readers feel that they have a better solution than we came up with, or would like to correct any errors we make, please write to us!

Q-emuLator and Zip files

This one proved to be a bit of a headache to document, although it was relatively straightforward for me as a relatively experienced QL user to sort out. It came about when Rich Mellor sent me a few zip files he had created on Q-emuLator, which I could not for the life of me figure out how to decode.

After a while it became obvious that not all zip files are equal and that there would need to be different methods of handling them depending on how they had been created and what you hadn't been told by the sender.

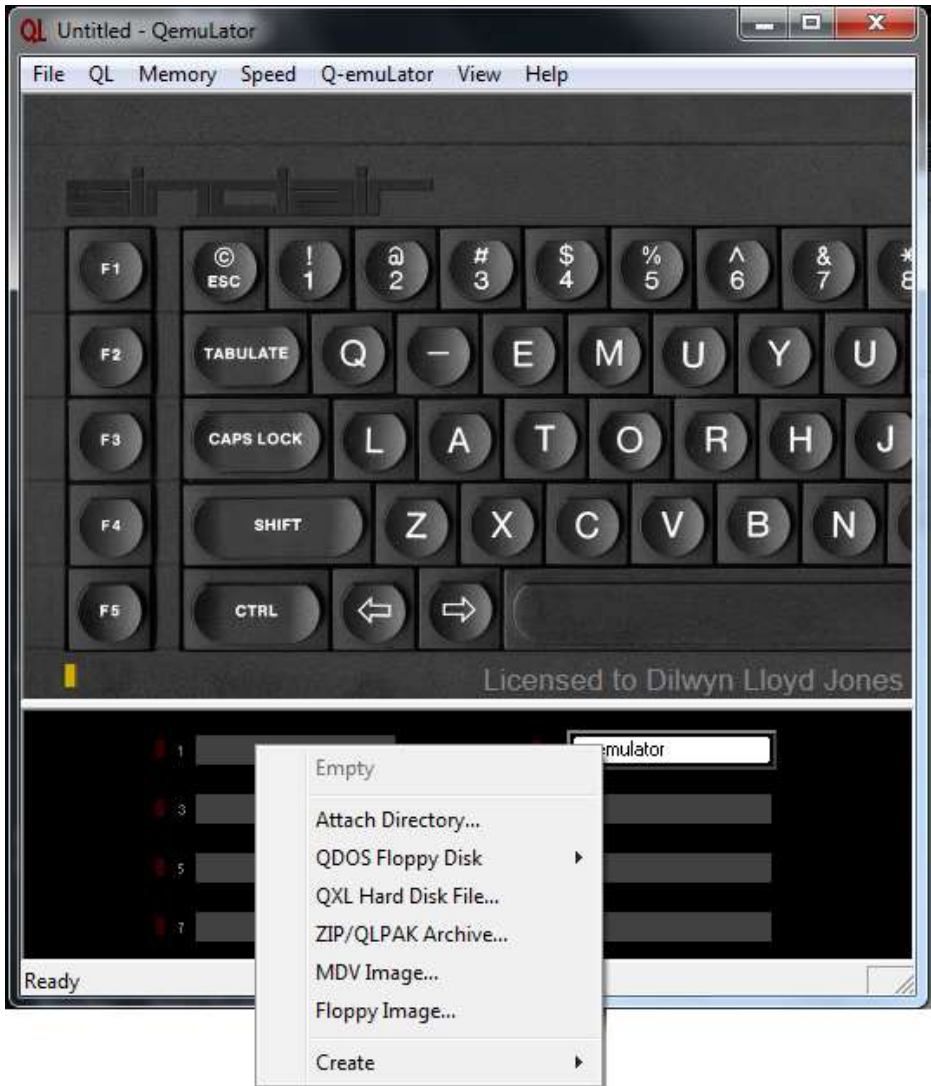
First of all, let's be clear that files zipped on a QL with a QL version of Zip are not the same as those created by, say, Windows. The QL version puts extra fields in the archives to store the file headers. Copy a QL executable program to Windows, zip it up and bingo, it's lost its executable file header and will never again start with an EXEC command – "bad parameter" will be the norm when you try.

That said, some emulators store QL executables in a native file format which is a bit different to standard QL file format. They add little bits of information to the file to make up for the missing bits lost by non-QL format media such as Windows, Linux and Mac formats. Ironically, copy these unmodified to a QL and these won't run either. Note that this generally only applies to executable files, data files can by and

large be moved between QL and non-QL media without too much harm where the file is not physically modified.

So we need to identify the various ways in which zip files are created and in this article we will concentrate on QemuLator. Most of the examples will be Windows format, as that's what I use. And you need to get the sender of these files to tell you what they are and how created for you to be able to successfully decode them.

Zipped QemuLator native file format folders – these are generally zipped on the host operating system, contain all the QL files from that directory on the sender's copy of QemuLator and will be in non-QL zip format created by software such as Winzip, 7-zip or even ordinary Windows zipped folders. They store the QemuLator file folders in non-QL format and need to be unzipped in the standard way for Windows, Linux or Mac systems. Once unzipped, attach them as a 'drive' to one of QemuLator's eight drive slots by right clicking on one of them and selecting the Attach Directory command from the menu.



Standard QDOS Zipped Files – these should have no emulator-specific data inside, so you should be able to read files from them just by attaching them directly to a drive using the 'ZIP/QLPAK Archive' command from the menu above. This would be the case when downloading a zipped QL software package from my website, for example. After downloading the file to a normal Windows folder on the hard disk, attach a QemuLator drive slot direct to the zipped file and simply do a DIR MDV1_ (if attached to slot 1) to see what's in there.

This won't let you save additional files to the zip file, but it's the quickest and simplest way to view and extract files in a QDOS zip file.

Zipped individual files from a QemuLator folder – these will be QL files, but stored in QemuLator format, that is, the executable files will be modified in line with the QemuLator format, adding extra information to a QL file to ensure it has its headers preserved while stored on a non-QL format drive such as a QemuLator native file format folder on a Windows drive. To unzip them just use a Windows program to unzip them directly into a QemuLator folder on the host OS drive.

Converting between QemuLator and standard QL file formats

Rule of thumb – to convert from QemuLator modified QL files to standard QL files, copy them from a native file format folder to a non-native format device such as a ramdisk or QL formatted floppy disk. The copying process to these devices removes any QemuLator specific file modifications and cleans them out back to standard QL format.

Zippping Files In QemuLator

If the files are in standard QL formats, simply use Jonathan Hudson's QDOS Zip program, or Archivers Control Panel, or my Zip Manager running in QDOS or SMSQ/E on QemuLator. Once in a QDOS format zip, the job headers are safely preserved for you to store the zip file on a hard disk, CD, website, email, etc as long as you unzip them using the corresponding unzip program later.

To send a whole QemuLator native format directory of files to someone else using QemuLator, just zip up the entire folder in Windows and send that, remembering to tell the recipient what you did, so that they can just unzip it in Windows and attach it to a QemuLator drive slot.

To send files from QemuLator to a QL user, or user of another emulator, ensure the files are on a QL format device such as RAMdisc or QDOS floppy disk first before zipping them – this helps prevent QemuLator specific headers creeping in which will render the file unusable on a standard QL.

If you make use of a QXL.WIN system with QemuLator, the same rules do not apply. These store the QL files in a standard format and so do not need to be copied to ramdisk or floppy disk to zip up.

In summary, without a good understanding of what QemuLator does to QL executable files, it can be hard to follow what's going on, but with a bit of knowledge of what's happening, and most importantly the sender and recipient telling each other what's going on, it's fairly easy to work out what's going on.