

## QLPAK

A QLPAK is, as the name implies, a method of packaging QL software into one file so that it may be run quickly and simply using the QL emulator called QemuLator.

The format of a QLPAK file is essentially a zipped file, containing the QL software and emulator settings so that as long as you have a registered version of QemuLator on your computer you can download the software in QLPAK format and use it straight away without even having to unzip anything. Indeed, if you visit the QemuLator website, you can click on links to QLPAK software such as a couple of sample games, and find that it can even start the copy of QemuLator on your computer and run the software immediately!

QemuLator can of course use a zip file as though it was a microdrive or floppy disc drive just by attaching a zip file as one of its eight drive slots and I expect that many QemuLator users use this feature already - QLPAK takes things a step further and lets you include the emulator settings needed for the program as well. For example, if a game needs a particular ROM version, or will only run in 128K RAM, you can specify this in the configuration file . That configuration file has the same format as the ones which the emulator already uses - users may be familiar with saving configuration files with names such as DEFAULT.QCF to save the preferred settings for the emulator.

Daniele Terdina is the author of QemuLator. He responded to my request for information about QLPAKs and I have used the information he kindly gave as the basis for this article.

I quote here from his email, and have added some notes where I felt

it clarified something.

These steps should be sufficient for most programs:

1. Copy the .QCF (ql configuration, specifying emulation options) file you want to use into the same directory as the QL program. To do this, just attach the directory containing the .QCF file to a QemuLator drive slot and copy it to wherever you have stored the remainder of the program to be made into a QLPAK.

2. You may need to edit the .QCF file with a text editor, for example to remove references to file paths on your machine. Don't associate anything to MDV slots, by default the QLPAK will be mounted as MDV1.

3. Zip all of the files together (including the .QCF file).

4. Rename the zipped file extension from .zip to .qlpak

5. Optional: if posting on a web page, associate the MIME type "application/x-qdos-qlpak" to the link - example below. This is what enables the browser to automatically start QemuLator when clicking on the link to a .QLPAK file. Hosting providers may sometimes disallow arbitrary MIME types by default, but there is usually a way to add new ones.

The above information is usually enough for most QLPAK files, but see also PAK: Drive Slots below.

For more complex cases, there are .QCF commands to perform operations such as:

- load a QL ROM image from the QLPAK
- mount different parts of the QLPAK to different microdrives
- specify and amount of memory, e.g. a game which will only run in 128K RAM

First of all we'll look at an example of a fairly simple game which only has two files - a BOOT file written in SuperBASIC and a single QL executable program called GAME\_EXE. This game needs 640K of RAM to run, but there are no other special requirements. It does not matter if it needs to run from MDV or FLP devices, as QemuLator treats these as the same device by default.

So we create a directory somewhere in QemuLator to hold the files so that we can zip them up to prepare them to become QLPAKs.

Say we have put them in RAM1\_. We have saved a suitable .QCF file called GAME.QCF somewhere from QemuLator with the settings needed to run the game and have copied it to RAM1\_. We now run the ZIP program (perhaps using Archivers Control Panel or Zip Manager or some such front end program for ZIP) and create a single zipped file called GAME.zip containing all three files. The last step to rename the zip file to a qlpak file as follows:

```
RENAME "flp1_game.zip" TO "flp1_GAME.QLPAK"
```

The quotes around the filenames are needed in QDOS since QDOS doesn't allow unquoted filenames to include the period '.' character. Filename case does not matter. You can do the renaming on the host operating system instead, depending on how and where you zipped

up the file.

If you need specific emulator settings such as a specific QL ROM version or a specific RAM size, you can tinker with the settings in the .QCF file if you know what the various lines mean. There is some information about this in the QemuLator manual (appendix 1) and here are some specific examples.

The .QCF file is a plain text file and so can be edited by hand in a text editor if you wish, and of course you can alter settings within QemuLator and save the current configuration ready made as a .QCF file by going to QemuLator's File menu, and using the "Save Current Configuration As..." command.

### **Specific Memory Size**

In a text editor, load the .QCF file. If loading it from QemuLator's default home directory on its Windows home drive, this will be probably ben in the Program Files\Qemulator directory on the C:\ drive. Look for a line which says something like:

```
Ram=640K
```

This sets the amount of RAM available to QL programs as 640K. If, for example, you have an old game which will only run on a 128K RAM system you would need to change this to:

```
Ram=128K
```

Alternatively if your program has a huge memory requirement such as a minimum of Gold Card sized memory, you could change this to:

Ram=1920K

Ram can also be specified in megabytes using M in place of K, for example, Ram=4M to have the memory of a Super Gold Card.

## **Speed**

Many old games had no real speed control and ran as fast as possible since the original QL was relatively slow for games. On modern fast QL compatible systems, these games may run too fast and become unplayable.

Emulator speed is controlled by a line which contains text like this:

Speed=Full

To ensure the emulator runs at a similar speed to a real original QL, change the line to:

Speed=QL

Or to make it run at Gold Card speed if the game is a bit sluggish at QL speed, try:

Speed=GoldCard

## **QL ROM Version**

If the program will only run on, say, a version JM or JS QL, or specifically needs an updated ROM such as Minerva, you need to alter the line which starts with 'MainRom='

For example:

```
MainRom=QL ROMs\QL_ROM_JS
```

This tells the emulator to start with the specified ROM image, in this case a ROM image with the filename 'QL\_ROM\_JS' contained in the directory called 'QL ROMs'.

For the QCF file included with the game, we could edit this to load the ROM image from the QLPAK itself. By default, QLPAKs are attached to drive slot 1 in the emulator, so assuming we need a version JM ROM image called JM\_ROM to be loaded from the QLPAK attached to drive slot 1, we could use:

```
MainRom=Slot1:JM_ROM
```

### **PAK: Drive Slots**

You may also need to tell the emulator that Slot1 is a QLPAK. Look for a line which starts with 'Slot1=' and change it if required to:

```
Slot1=PAK:
```

The other slots may be left as 'Slot2=Empty' etc. unless specifically needed.

### **Software Needs A Specific Back ROM**

Some software may need Toolkit 2 or another EPROM which was plugged into the QL's EPROM expansion slot. Some early QL language compilers and interpreters needed a plug in EPROM to operate.

Suppose your game needs Toolkit 2 to run. Many QemuLator users will already have Toolkit 2 available on their system, but you cannot

guarantee this. Some users simply don't know a Toolkit 2 ROM image is supplied with current versions of QemuLator, some don't know how to implement it, and some just don't bother with it. So, assuming your software complies with the licensing requirements for Toolkit 2 ROM images (distributed on a non-profit basis for use with QL emulators) you may choose to include Toolkit 2 and tell QemuLator to use your Toolkit 2 ROM image called TK2\_rom from the QLPAK.

This is what QemuLator's QCF definitions for Toolkit 2 would normally look like this:

```
BackRom=QL ROMs\TK2_rom  
BackRomActive=Yes
```

So the .QCF file included with your game could change this to load the copy of Toolkit 2 from your own QLPAK file:

```
BackRom=Slot1:TK2_rom  
BackRomActive=Yes
```

### **Auto-Starting The Software**

QCF files contain some useful options controlling how the emulator starts up. For example, it is possible to auto-start without having to press the usual F1 or F2 keys of a "real" QL.

```
FastStartup=No  
AutoStartSession=No  
FirstKey=None
```

Changing the first line to FastStartup=Yes makes the emulator skip

the initial QL memory test, to make the emulator start a bit faster than it might otherwise when emulating the QL startup memory test and tweed screen.

The AutoStartSession line selects whether or not to launch the emulation session when QemuLator starts, in other words, whether or not you need to click on the picture of the QL keyboard or click the command 'Start' in the 'QL' menu

FirstKey controls which key if any is pressed automatically when the QL displays the F1/F2 startup option. This can be None, F1, or F2. Minerva may also allow F3 or F4 for additional special startup options.

By now you should be able to appreciate the possibilities. Remember that Appendix 1 of the QemuLator manual describes the format of these .QCF files and the options available.

### **Web Page Link To A QLPAK**

In order to link to a QLPAK on a web page, you need to include code to associate the MIME type "application/x-qdos-qlpak" to the link. This fairly horrendous looking statement is actually quite simple to encode once you know how and have a few examples to hand. See how this is incorporated within the anchor (link) tag in this example which offers to download the qlpak from the same folder as the web page:

```
<A type="application/x-qdos-qlpak" HREF="game.qlpak"> Download my game</A>
```

This ensures that as long as a registered copy of QemuLator exists on the computer downloading the QLPAK, the emulator is started up, the QLPAK file is associated with drive slot 1 by default and the emulator starts up with the settings specified within the QCF file. Windows, for example, will ask if you want to Run or Save the file at first, hopefully if you specify 'Run' the QLPAK will be associated with QemuLator which will be started and hopefully run the program according to the list of settings in the .QCF file.

## **Conclusion**

This article reads like a highly technical document, but in fact once you have made the effort to create a QLPAK or two and seen how each permutation changes how the QLPAK is started and handled, you will find that this is an extremely useful facility, making it possible (to all intents and purposes) for QemuLator to run games direct from a web page!

All this is pretty new to me – it was only when Daniele Terdina supplied the basic information at the beginning of this article that I realised it would be fairly straightforward once I had the necessary information to implement this. Time permitting, I may try to make some of the games on my website available in QLPAK format to see how users respond. You do need a registered version of QemuLator (costs \$45.95 for Windows or Mac versions, or \$55.95 for both, or \$33.95 to update from earlier versions) on your computer to run QLPAKs of course, plus some knowledge of zip to create the QLPAK in the first place, but I think the work needed to learn how to use this facility makes it very worthwhile and above all a very convenient and underused facility.