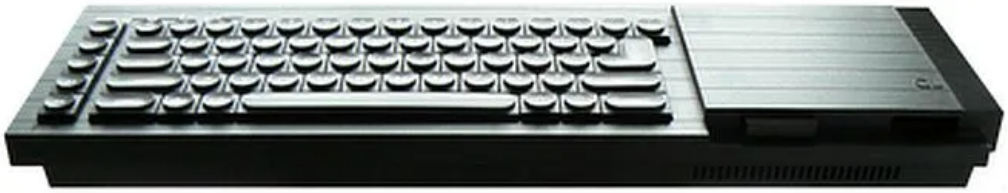

Dilwyn Jones Sinclair QL Blog

The world of the Sinclair QL at http://dilwyn.me.uk

Magic Fields (Executable Headers)

February 26, 2017 · *by dilwyn2*

One of the things I get asked most often about downloading files from my website is why programs don't work after unzipping.

The answer to that is "how did you unzip it?".

"I used Winzip/7-zip/whatever, it unzipped fine but when I try to start the program it just says Bad Parameter – is the zip file corrupt?"

OK, no it isn't and don't do that.

QL executable programs (executable=program you can start with an EXEC or EXEC_W command in QL BASIC) have what is known as an "executable file header" – some people call it the "magic header" for some reason. This is a part of the file header which marks it as a program, and holds what is called a DATASPACE value. Dataspace is a portion of memory allocated to a program to hold variables and such data, as the name implies.

Since this "dataspace" value is not normally physically part of the program (sadly! but QL programs are not meant to modify parts of themselves so the datspace is separate) it has to be held somewhere. So it's put in the file header. But Linux, Windows, Macs, etc don't understand file headers, so when you unzip something in a non-QDOS environment, all that happens is that you get the program code only and unfortunately not the executable file header and dataspace. You transfer the file to a QL and as far as the QL is concerned it's now no longer executable, it's just a normal data file without a dataspace. QLs have a habit of not being able to execute data files but don't really have a suitable error message for when that happens and just uses what the ROM thinks is least inappropriate, in this case "bad parameter" as something it needs is missing.

That's the bad news.

The good news is that with a little patience and effort things can be fixed.

The easiest way is to go back to the original zip file and transfer that to the QL, and only THEN unzip it using QDOS unzip.

Which is fine if you already have QDOS unzip, but how do you get that first zipped copy of QDOS unzip from the website to the QL in the first place, chicken and egg, catch 22? Well, you can use either the self-extracting version of unzip if it happens to work on your system – see http://www.dilwyn.me.uk/arch/index.html, or use a special version of Unzip on my website which has been converted to a BASIC program and

Search ...

RECENT POSTS

- QBASIC
- QL Homepage
- QL Technical Review
- International QL Report
- Graeme Gregory QL Website

RECENT COMMENTS


Petri Pellinen on AnimGIF


Outsoft on RIP Sir Clive Sinclair

QL-SD ROM news | Kil... on QL-SD-ROM


John Hitchcock on Spares And Repairs


France Emulation on Q-emuLator for macOS

ARCHIVES

- April 2023
- March 2023
- January 2023
- December 2022
- November 2022
- October 2022
- August 2022
- May 2022
- March 2022
- November 2021
- September 2021
- July 2021
- June 2021

rebuilds itself after being run on your QL. BASIC programs aren't "executable programs" so aren't affected as such by Windows or Linux or Mac… See Article on transferring files

Alternatively you can do this the hard way and repair the executable program file yourself by converting it from a data file to an executable with the magic word SEXEC.

SEXEC is a QL BASIC command to save a program and give it an executable header and dataspace value. Often you will need to guess this dataspace value unless you happen to know it, although programs made using some compilers store the dataspace in something called an XTcc field near the end of the program, which is a great help.

Where you have to guess the value, this could be anything from a few bytes to hundreds of kilobytes for programs with large amounts of data, unfortunately. If you make the value too small, the program will run out of memory space to store its variables and stop with an error message. If you make the value too big, it's wasteful of precious RAM memory.

As an example, I have a program called TEST_EXE which has lost its file header. I have transferred it to FLP1_ on the QL but it won't start using EXEC or EXEC_W so I'll try to repair it with this program, which needs Toolkit 2:

```
100 fileLength=FLEN(\"FLP1_TEST_EXE") : REMark hc
110 base=ALCHP(fileLength) : REMark some space to
120 LBYTES "FLP1_TEST_EXE",base : REMark load dam
130 SEXEC "FLP1_TEST_EXE2",base,fileLength,1024 :
140 RECHP base : REMark finished with the reserve
```

The "dataspace" value is the 1024 in line 130. Unless you happen to know it, this is just a wild guess to start with. You may have to try this many times until you stumble on a suitable value. Start small (to see if it's one of those programs that needs very little dataspace), then go big, then successively halve the value until you find what seems to be ideal.

Note that line 130 saves the file with a new name, so that you can go back to the original if something goes wrong. Once you've fixed it, you can delete the original and rename the working copy as you see fit.

Line by line:

Line 100 checks the actual size of the program, so that we know how much memory we need to store it temporarily while we fix it.
Line 110 allocates that amount of memory for this job.
Line 120 loads the broken program into memory as a data file using LBYTES.
Line 130 uses an SEXEC command to try to fix the problem program.
Line 140 releases the temporary memory used back to the system.

XTcc Fields

Now for the "slightly harder to explain but potentially easier to fix a program" method.

Programs made using some compilers such as C68, QDOS-GCC and XTC68 include something called an "XTcc Field" within the program code to indicate what the dataspace should be. It's usually found at the end of a program in the examples I've seen, though that might not always be the case. It's designed to allow program compilers running on other systems ("cross-compilers") such as XTC68 on a DOS or Linux computer to create QL programs and have a means of knowing what the dataspace will be when it eventually finds its way to a QL.

The four letters "XTcc" precede a long word value (high byte first, 68000 processor style). That long word is the dataspace required; in other words, the value you need to replace the 1024 in line 130 of the example above.

Checking for an XTcc field being present is relatively easy if you have an editor (e.g. Ralf Reköndt's S-Edit) which is able to load a binary file. In S-Edit, press F3 and use the Bin command to load the TEST_EXE or whatever your program is called. Use the Find command to search for the short text string "XTcc". Usually it'll find it near the end of the program. The 4 bytes after this will contain the values for the dataspace. By positioning the cursor over the 4 bytes in turn S-Edit will show the code value of the character under the cursor. Make a note of these and work out the long word value:

256*256*256*first + 256*256*second + 256*third + fourth

So if the 4 bytes are 0,0,16,52, we get a dataspace of 4148 bytes. Simples (as the Meerkats might say).

It would probably be fairly easy to write a program to load the file, find the string XTcc, then work out the dataspace after that, but it'd probably be too long for inclusion here, so I'll leave that to you!

This entry was posted in Programming, QL Software, Uncategorized and tagged dataspace, executable, ql program, XTcc. Bookmark the permalink.

« *BMP PROGRAM V1.03* *NEW PHOTON »*

4 thoughts on "Magic Fields (Executable Headers)"

tlosoft says:

February 26, 2017 at 9:02 pm

Why not have a link on your website download page
To this blog for those who download zips

Reply

Dilwyn says:

February 26, 2017 at 9:24 pm

Good idea-done.

Reply

Derek Stewart says:

February 26, 2017 at 9:19 pm

- February 2014
- January 2014
- December 2013
- November 2013
- October 2013

CATEGORIES

- Books
- Emulators
- Programming
- QL Events
- QL Hardware
- QL Magazines
- QL News
- QL Software
- QL Websites
- Quanta
- Uncategorized

META

- Register
- Log in
- Entries feed
- Comments feed
- WordPress.com

Maybe we need a new UNZIP programme for the PC operating system, Linux, Windows or IOS.

Not sure how to do this, but the source to the common unzip programmes are available.

Is it worth doing, personally, I use ACP via NFA, DOS device to unzip the zip file from SMSQmulator or QPC2

Reply

Norman Dunbar says:
October 30, 2018 at 2:19 pm

If anyone wants a SuperBASIC utility, with decent error trapping, that will read a file that was cross compiled for QDOSMSQ, and has an XTcc record with the desired dataspace, then have a look at https://qlforum.co.uk/viewtopic.php?f=3&t=2606 where a small, but perfectly formed utility awaits!

Cheers,
Norm.

Reply

Leave a Reply

Enter your comment here...