

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 our 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. I'd like to be able to print on both sides of the paper from a QL. My PC can do this without any problem. Can I do this from a QL too?

A. The feature is built into some Windows printer drivers but sadly is not usually accessible from QL emulators. One way around this is to use a program like my "2Sides" software for printing plain text files at least - download it from <http://www.dilwyn.me.uk/printing/index.html> (scroll down the page until you find the "sides" package).

If anyone has any further information about printing on both sides of the paper from a QL, or knows of another program able to do this, please do get in touch.

Q. I have been sent some QL text and database files via email and no matter how I try to save them from Outlook Express, they always end up with the wrong filename extension - usually it adds a ".dat" to the filename which prevents them from loading when I copy them to a QL disk.

A. In general, once they have been saved to the PC, you can probably rename them before transferring them to the QL. Usually this is as simple as clicking on the filename once, then either pressing F2 or clicking on the filename again to be able to retype it to correct the extension, removing the ".dat" being OK in many cases. Sometimes, you may find that certain QL extensions get in the way of the PC if they have been sent with a Windows-style "." separator instead of the QL "_" separator. An example is a BASIC program, sent as "myprog.bas", which may trigger security software on the PC if it thinks it may be a PC program such as a virus. In this case, transfer it to the QL and rename it there if you have problems with removing the ".dat" extension on a PC.

If your QL disk transfer software has copied the filename to a QL disk "as is", e.g. it has copied to QL disk as something like "report_doc.dat" the '.' in the filename may present difficulties to QDOS systems in renaming it unless the filename is in quotes, e.g. `RENAME "flp1_report_doc.dat" TO "flp1_report_doc"` should work fine on most systems. SBASIC usually allows you to use a dot in unquoted filenames, but putting the filename in quotes should ensure it works on all QL systems.

Q. Is it possible to call hotkey definitions from within my programs?

A. For some hotkey actions, the answer is yes. There is a command called `HOT_DO` which will execute the action defined to a given key. For example, if you want to send something via the stuffer buffer without having to manually type Alt-Space to recover the content of the stuffer buffer, use a `HOT_STUFF` command to send the string to the stuffer buffer, then a `HOT_DO " "` command (a single space within the quotes) to effectively issue an Alt-Space keypress:

```
HOT_STUFF "some text"  
HOT_DO " "
```

This can be useful for testing a program action, for example, although it does not ensure the string is sent to the required program. For this, we would need a `HOT_PICK` function as well

If a `PICK` hotkey has already been defined, e.g. `ERT HOT_PICK("b","")` - no spaces between the second set of quotes, you could `HOT_DO` that definition to force the machine to pick BASIC, pause

slightly for that command to have time to switch to BASIC then HOT_DO the stuffer buffer to make it send the command to the job, in this case BASIC.

Should you need to assign a hotkey temporarily for a given hotkey action, you can use a rather quick and dirty routine like this to temporarily find and assign an unused hotkey:

```
100 key = 0
110 FOR a = 33 TO 127
120   IF HOT_TYPE(CHR$(a)) = -7 THEN
130     ERT HOT_PICK(CHR$(a),"jobname") : PAUSE 10
140     HOT_DO CHR$(a) : PAUSE 10
150     HOT_STUFF "some text" : PAUSE 10
160     HOT_DO " " : PAUSE 10
170     ERT HOT_REMV(CHR$(a)) : REMark remove the temporary
definition
170   END IF
180 NEXT a
190 PRINT "Oops, no undefined hotkey!"
200 END FOR a
```

This is rather an extreme example, and also not particularly good practice in a multitasking system where another program may define or change a hotkey.

Also, the above example is probably better written using the HOT_CMD function, which is expressly written to be able to send commands direct to BASIC!

Q. I'd like to set up a hotkey to send several lines of text at a time. Is this possible?

A. I didn't know this until I looked it up in response to this question. Apparently, a HOT_KEY function can include several strings, all of which are sent separated by newlines (CHR\$ 10). Here's an example, which can be used to type your address into a letter:

```
ERT HOT_KEY("a","address line 1","address line 2","and so on")
```

Q. I've seen some "colour theme" files on your website. My system does have "colour drivers" but how do I use these theme files?

A. They can only be used on systems with Window Manager 2 and preferably colour drivers too (colour themes don't work too well in just 4 colours!). Window Manager 2 is available in SMSQ/E version 3.00 and later, plus on QDOS systems with pointer environment version 2 or later (although with QDOS you'd still only get mode 4 colours).

Here's a short BASIC program which can be used to install the colour theme files. The ones on my website at <http://www.dilwyn.me.uk/gd2/index.html> were created using Wolfgangh Uhlig's QcoCo program, where you can define your own set of system colours. These files usually have 57 entries (hence the 57 in line 130) - the maximum number of entries in a colour theme list, or 57 separate elements such as "window background colour", "loose item selected", or "loose item unavailable". The benefit of using Wman 2 colour themes is that many modern programs out there which understand the process can use the same colour scheme and take on a similar appearance to each other, hence the name.

```
100 REMark apply colour theme to System Palette 0
110 theme$ = "filename_thm"
120 OPEN_IN #3,theme$
130 entries = 57 : REMark number of word length entries in list
140 memory = ALCHP(2*entries)
150 addr = memory
160 FOR addr = addr TO addr+2*(entries-1) STEP 2
170   INPUT #3,colour_word$
```

```
180 colour = HEX(colour_word$(2 TO LEN(colour_word$)))
190 POKE_W addr,colour
200 END FOR addr
210 SP_SET memory,0,entries : REMark apply
220 RECHP memory : REMark release heap memory used
230 CLOSE #3
240 QUIT
```

Q. I'm interested in starting to write programs using pointer environment, using either TurboPtr or Easyptr, but am confused by all the terminology like "loose item", "information window", "application menu" and so on. Where do I start, as neither program seems to have a nice gentle introduction to the basics?

A. Probably the best introduction to this subject is Norman Dunbar's "Pointer Environment Idiot's Guide". Like most "Idiot's Guides" this one sets out to teach the basics from first principles. It won't teach you to program, that's beyond its scope, but it does teach you things like the terms in a simple and fairly non-technical way. The PE Idiot's Guide is available from the usual sources of free QL software, e.g. The PE Idiot's Guide from my website at <http://www.dilwyn.me.uk/docs/ptr/index.html>