

QUANTA HELPLINE 03 - Published in Vol 25 Issue 2, Apr/May 08

Q. I would like my BASIC program to start up filling the screen, no matter what type of system it runs on. How can I achieve this?

A. SBASIC has two functions, SCR_XLIM and SCR_YLIM which will tell us the maximum width and height of the screen. Unfortunately, SuperBASIC has no such functions, but as QDOS cannot cope with anything other than 512x256 screens (with the possible exception of uQLx emulator). The simplest solution is for your programs to assume a 512x256 screen, then if running on SBASIC to call SCR_XLIM and SCR_YLIM

```
100 wide% = 512 : REMark screen width
110 high% = 256 : REMark screen height
120 IF VER$ = 'HBA' THEN wide% = SCR_XLIM : high% = SCR_YLIM
130 WINDOW wide%,high%,0,0
```

This will work if QLiberator compiled, though Turbo may have a problem as it checks to see if all keywords required are loaded as the task starts.

Some toolkits such as the QL Today Display-Code extensions have keywords for checking the screen sizes. These can be included free of charge in any compiled programs, although the above solution is probably the simplest for an interpreted BASIC program.

Q. How do I include extensions in a compiled program?

A. The method varies for Turbo and QLiberator compiled programs, although the principles involved are the same. QLiberator users should study the details of the \$\$smb directive in the manual. Basically, the directive is placed in a REMark statement near the beginning of the program, along with filename of the extensions file, an offset value to a setup routine and an initialisation routine, the difference being that the first value is a setup routine (e.g. allocate working space, or set up a Thing) while the second value links the new keywords to S(uper)BASIC. Take as an example the QLiberator extensions file itself, called QLIB_EXT. We specify its filename and a pair of values like this. It has no setup routine, like many common extensions) and the table of new keywords starts 12 bytes into the file:

```
120 REMark $$smb=win1_qlib_ext,0,12
```

The filename may be case sensitive in some versions of QLiberator.

The Turbo equivalent (in more recent versions of Turbo only) is also specified by a directive in a REMark statement near the start of a program:

```
120 REMark %% win1_qlib_ext,0,12
```

By and large extensions files will have details of how to include them in a compiled program in their documentation. If not, you need to look at the assembler source to work out where the definition table lies. Most extensions have a few machine code instructions before the table to linking the extensions before the table of keywords, so a very broad rule of thumb is to try a value of 10 or 12 for the last number, and a value of 0 for the first one.

Not all extensions files are suitable for inclusion - some are best loaded from a boot program. Note also that as QLiberator generates mostly ROMable code, some extensions contain data or storage areas within the body of the extensions file. If one of these extensions modifies any part of its own area, the code generated will no longer be ROMable.

Q. I'm thinking of buying QPC. Can you tell me how much memory it can handle? I have a Super Gold Card at the moment and would be very happy if QPC could provide me with at least as much memory to run a fairly large program I use.

A. Good news - QPC2 can handle up to 128MB of memory, although if the PC does not have this much available RAM, QPC2 will try to use "virtual memory" whereby Windows will allocate disk space larger than RAM and swap the content of disk and memory as required from time to time so that QPC2 thinks it's running in that amount of memory, although the disk activity can slow things down a little. The good news, however, is that QL software does not usually need anywhere near this much memory. Most QPC users allocate anything from 2MB to 16MB and usually find that this is more than enough for a QL system.

Q. I am hoping to use a program of yours which states it needs "Window Manager 2". Does this mean that it needs SMSQ/E and colour drivers? I have a Super Gold Card QL with a version JS ROM and I do normally use pointer environment.

A. Not necessarily. Most programs which state that they need Window Manager 2 can work on traditional QDOS systems as long as you have pointer environment version 2.00 or later installed. Window Manager 2 is not the same thing as a colours drivers system, since Window Manager 2 can work even in good old mode 4, although not as colourfully of course. Most programs will tell you if they need to run on a high colour system.

Q. Which systems can give me more colours than a QL?

A. These are generally systems which run SMSQ/E, since QDOS does not provide facilities for handling the new colours. There are a few programs around which can run in 256 colours on an Aurora in QDOS by directly manipulating the hardware and screen, but these are very much in a minority. To the best of my knowledge, the following systems can offer more colours than a standard QL:

1. QPC2 (Windows) - 16-bit or 8-bit colour modes
2. Q40 or Q60 - 16-bit colour modes.
3. Aurora card - 8-bit colour mode. Also has a 4-bit colour mode, but this is not supported by SMSQ/E currently.
4. QXL card for PC - 16-bit colour modes.
5. QemuLator - has a mode which can run the 8-bit colour mode by using the Aurora "colour drivers" version of SMSQ/E, and more recent versions have a Q60-style display with 16 bit colours, running a custom version of SMSQ/E available from Daniele Terdina's website.
6. SMSQmulator

If anyone knows of other platforms I have not listed here, I will gladly update this list.