Toolkits. 600 reasons to own a QL

by Sid Martin and Timothy Green

Easily customised, the QL is an ideal machine for expansion. QL toolkits abound, giving hundreds of key commands

ONE OF the best features of the QL is its extensibility – it's easy to 'customise' the system to your own way of working. A cynic would say that this is because Sinclair was not sure how it wanted the QL to work while it was designing it! In any case, the QL design makes it remarkably easy to add new commands and functions to the system.

The standard QL has over 100 built-in commands, supplied as 'resident' procedures and functions. Almost all of these can be used in SuperBASIC programs, and SuperBASIC routine names are treated just like machine-code commands while the appropriate BASIC is loaded.

New collections of commands appeared soon after the QL launch, adding to the standard repertoire and making the keyboard and files easier to use. Such collections were known as 'Toolkits'.

The first was published by Sinclair and just called *QL Toolkit*. then came *Super-B*, supplied by HiSoft on a plug-in ROM (Read Only Memory – a pre-programmed chip).

Super-B is no longer available, and *QL Toolkit* has been superceded by the current market-leader, *SuperToolkit 2*, published by Q-Jump and normally supplied in ROM.

Other toolkits have followed, on disk and cartridge. 1986 saw the arrival of *Turbo-Toolkit* – the first part of Digital Precision's 'Turbo' programming environment. Then came *Mega-Toolkit*, a budget title from Utilising Software. The latest toolkit, confusingly called *Mega Toolbox*, is published by Compware, and boasts 170 machine-code routines in 25K of code.

Toolkits make life easier for QL users and programmers. To illustrate this, we've picked one routine from each of the toolkits reviewed.

The examples are arbitrary, but they do indicate the power of toolkits, especially when you realise that each toolkit contains over 100 routines with a wide variety of effects.

Have you ever wished you could edit the program listing that appears in window 2 as you type in SuperBASIC? *SuperToolkit*'s ED command lets you drive the cursor over the listing, skipping from line to line, scrolling up and down, making changes at will.

This 'screen editor' is much quicker and easier to use than the normal EDfT command. It lets you edit the current BASIC instantly, with no need to save the text, load it into a separate editor, change it, save it, and re-load it back to BASIC.

The QL. Easy to expand

Maybe you'd like to be able to input characters the way the Psion programs do it – supplying a 'default' entry that the user can confirm or edit? *Turbo-Toolkit*'s EDIT$ function will do the trick, and make sure the window is not messed up if the user types more than you expect. EDIT% and EDITF can stop people entering nonsense when a number is expected.

Reams of SuperBASIC can be replaced by simple, concise toolkit commands. For instance, *Mega-Toolkit* has a JOIN command, which takes a list of files and links them all together into one new file.

Mega Toolbox has CHOOSE$, a function which takes a numeric parameter and several strings. The value of the number determines which of the strings is returned. Of course, you could do this with SELECT or ON..GO, but CHOOSE$ is often neater.

Distillation

It's impossible to list and explain over 600 commands in a review – even in *Computer Shopper*. Features depend on the particular program – there's some overlap, but each toolkit has unique commands and effects. We've distilled our test results into a separate section about each toolkit, a comparison table, and a discussion of common features.

All the toolkits come with commands to load, monitor and set the priority of multi-tasking jobs. They let you turn cursors on and off, re-define the character set, and grab or release memory even when the normal RESPR function won't work, such as when tasks are running.

The toolkits include multitasking clocks, which display the date and time continuously onscreen, as well as alarms that make the computer beep at a preset time. They can copy between devices as you do something else – a useful trick known as 'spooling', supposedly an acronym for 'Simultaneous Peripheral Operation On Line'. You need a compiler to get multi-tasking effects out of *Turbo-Toolkit*.

The ALT key brings back the last command entered in *Mega-Toolkit* – this is useful, as it allows you to edit the last entry, but the key chosen is unconventional. The ALT key is meant to be used as an ALTernate shift, in conjunction with other keys rather than on its own.

More sensibly, *Mega Toolbox* uses Shift Enter to recall the last line. Both *Megakits* let you turn this feature on and off.

SuperToolkit uses ALT-ENTER to recall the last entry in any window. It lets you step backwards through all the old entries held in the keyboard queue – not just the last command. This is both an advantage and a disadvantage.

The entire entry – including insertions, deletions, and cursor moves – is re-called, so you can get strange results if the cursor position has changed, or single character responses have been typed since. Other strange things happen when the end of the buffer is reached.

The result is that ALT-ENTER usually does what you want, but sometimes it gets in a mess, and it can end up spewing out lines of garbled data, perhaps including ENTER characters. We've heard reports of FORMAT commands coming back to haunt *SuperToolkit* users ...

All the Toolkits let you assign strings of characters to the function keys, so that each key produces a different string. For instance, you could set key F2 to produce the string 'PRINT', which might be convenient when typing in a listing.

Mega-Toolkit allows 10 strings to be programmed, using the five function keys in combination with SHIFT. *Mega Toolbox* recognises CTRL and ALT as well, giving 20 programmable combinations. *SuperToolkit* doesn't stop at the function keys – it lets you assign a sequence to any character typed in conjunction with the ALT key, giving well over 100 possible combinations.

Mega Toolbox and *Turbo-Toolkit* let programs put characters into a keyboard input channel as if they had been typed by the user. This is the only way you can implement function keys and key re-definition with *Turbo-Toolkit*, and you need a compiler to make the code multi-task.

SuperToolkit has a LRESPR command which simplifies program initialisation. LRESPR

Continued on page 91

Continued from page 89

QL TOOLKITS – THE SHOPPER GUIDE

This table should be read in conjunction with the review.

loads and calls a file of system extensions, replacing RESPR,

	MEGA TOOLBOX	MEGA TOOLKIT	SUPER TOOLKIT	TURBO TOOLKIT
Price	£29.95	£12.00	£29.90	£29.95
No. of routines	170	130	90+24	72+129
Code size	25K	14.5K	16K ROM	6K
Manual (A4 pages)	80	13	80	80
Copy protection	N	Y	ROM	N
KEYBOARD IMPROVEMENTS:				
Function keys	20	10	175	(2)
Last line recall	Y	Y	Y	N
Automatic 'type in'	Y	N	N	Y
Command files	N	Y	Y	Y
Clear key buffer	Y	Y	N	N
Set keyboard timing	Y	Y	N	Y
Program queue change	Y	N	N	Y
Clear queue	Y	Y	N	N
Value editing FNs	N	N	N	Y
Improved line editing	Y	N	N	N
FILE HANDLING:				
Sub-directories	N	N	Y	N
Filename wildcards	N	(1)	Y	N
Date stamping	N	N	Y	N
Binary files	Y	(4)	Y	Y
Random access files	Y	Y	Y	Y
Virtual arrays	N	N	N	Y
Network driver	N	N	Y	N
DISPLAY HANDLING:				
User defined graphics	Y	Y	Y	Y
Area graphic commands	Y	N	N	N
Scaled text output	Y	N	N	N
Pop-up windows	Y	N	N	Y
Screen zoom	Y	N	N	(4)
Screen compaction	Y	N	N	N
Circular scroll/pan	Y	N	N	(4)
Mixed-mode screens	Y	N	N	N
Cursor control	Y	Y	Y	Y
Paged screen reports	N	N	Y	N
Paged file VIEW	N	Y	Y	N
MULTITASKING:				
Clocks and Alarms	Y	Y	Y	(2)
'Background' INPUT	Y	N	(5)	(5)
'Background' copying	Y	Y	Y	(2)
'Background' sounds	Y	N	N	(2)
Inter-task pipes	N	N	Y	Y
In-task pipes	Y	N	N	Y
Task list/control	Y	Y	Y	Y
EXEC with abort	N	N	N	Y
MEMORY MANIPULATION:				
Heat RAM management	Y	Y	Y	Y
TRAP call commands	Y	Y	N	N
Call and return regs.	Y	N	N	N
Base conversions	Y	Y	Y	N
Bit manipulation	Y	Y	N	N
Memory block move	Y	Y	N	Y
Memory search	Y	N	N	Y
Memory fill/exchange	Y	N	N	[illegible]
ERROR TRAPPING:				
Error trapping	N	(1)	(3)	(2)
Device name trapping	N	N	Y	Y
Numeric input trapping	N	N	N	Y
Overwrite Y/N?	N	N	Y	N
SUPERBASIC EXTENSIONS:				
Number formatting	Y	N	Y	Y
String formatting	N	N	Y	N
BASIC access functions	Y	N	N	Y
Name table access	Y	N	(6)	Y
Name list access	N	N	(6)	Y
BASIC area-size FNs	Y	Y	N	Y
Program profiling	N	N	N	(2)
TRACE line-numbers	N	N	N	(2)
BASIC cross-reference	N	N	N	Y
BASIC screen editor	N	N	Y	N
List extra routines	N	Y	Y	Y
SELECT-type functions	Y	N	N	N
Case conversion FNs	Y	Y	N	(5)
String utility FNs	Y	N	N	N
Numeric utility FNs	Y	N	N	N

Notes:

(1) Only for routines in the toolkit.
(2) Only if supplied routines are compiled.
(3) Only for late model (JS or MG) QL systems.
(4) After a fashion, but not as well as other Toolkits.
(5) Only using a compiler and 'pipes'.
(6) Parameters only.

LBYTES, CALL, removing the need for a temporary variable (the address) and a 'magic number' – the file length. RESPR-L and ALCHP-L do the same thing if you're using *Mega Toolbox;* the latter works when tasks are running, unlike a normal RESPR.

Both *SuperToolkit* and *Turbo-Toolkit* include improved versions of the EXEC command, which let you pass parameters to tasks and link them together with 'pipes' – communications channels that let separate tasks pass information between them with PRINT and INPUT.

Turbo-Toolkit also lets you set priorities as tasks are loaded, and 'break into' task execution – EXECUTE-A is much like EXEC_W, except it stops the task and closes files if particular keys are pressed.

Mega Toolbox has a unique EXECUTE command. Pipes are not supported, but you can set the priority and dataspace of a task when you load it, and – best of all – pass a string of characters which will be automatically typed into the task as if it had been entered from the keyboard. You can use this to get a task started to suit yourself – for instance, to pre-set Psion 'Design' options.

Turbo-Toolkit and *Mega Toolbox* include CONNECT commands that let a single task read and write a pipe – in effect, using it as a fast temporary file, held in memory.

Mega Toolbox and *Turbo-Toolkit* can automatically make a cursor flash when it's turned on. Other systems require the user to press 'Control C' repeatedly to select a particular cursor. The automatic change is not compatible with QRAM, so version 2 of *Turbo-Toolkit* lets you disable it by putting an exclamation mark at the end of the CURSOR-ON command.

Both *Mega* packages include functions to read the details of windows. *Turbo-Toolkit* has the most comprehensive range, but its functions give the wrong results if QRAM is running.

All but *SuperToolkit* boast commands to move memory contents around en bloc. *Turbo-Toolkit* can search memory for particular sequences. *Mega Toolbox* includes comprehensive commands to search, swap, dump or fill areas of memory.

SuperToolkit

SuperToolkit contains 90 new commands and souped-up versions of 24 standard commands. It also upgrades the QL's keyboard and file control software.

Both new and standard file-handling commands have been re-written to take account of a 'default device' assumed if no explicit device name is supplied. You can change the default for data and programs separately.

SuperToolkit file commands also ask questions if files might be overwritten. This makes the QL much easier to live with. For instance, if you re-save a program by typing SAVE DEMO, this message appears at the bottom of the screen:

```
FLP1-DEMO exists, OK to overwrite..Y or N ?
```

Type 'Y' to replace the old file. Sometimes the name defaults give odd results – for instance, if you type:

```
SAVE MDV1-TEST
```

without a cartridge in microdrive 1, *SuperToolkit* decides to use the default, rather than give an error message, so it creates a disk file called 'FLP1-MDV1-TEST'!

SuperToolkit supports 'wildcard' file names – these are like skeleton keys, in that one name may match several files. Names are divided up into sections separated by underscores. For instance:

```
WCOPY -BAS TO FLP2--BAS
```

performs a 'wild copy'. Files on the default device with names ending -BAS are listed at the bottom of the screen, and you're asked whether you want to copy each one to FLP2. You type Y or N, Q to quit, or A to copy 'all' without further questions. A similar command, WDEL, lets you delete groups of files with similar names.

You can include a prefix with a device default, so that commands refer only to files which start with a particular word or group of words. This means you can use the QL's 36 character file names to split files into groups; the effect is rather like separate directories in MsDos or Unix.

WDIR is like DIR, but only shows files with names that match a particular pattern – the full name is shown, unlike a true hierarchical directory. *SuperToolkit* adds 'date stamping' to all QL file devices. WSTAT lists file lengths and the time when files were written, as well as names.

Lists can be sent to the screen or any other device – if they're displayed, *SuperToolkit* pauses automatically as if you had pressed CTRL-F5 to halt scrolling after each 'page' of text. The next page scrolls into place when you press a key.

If you've got several QLs or CST Thor computers, each with a copy of *SuperToolkit*, you can use the 'network driver' to share devices between machines. This is a very powerful feature – it means any computer on a network can access any device on another machine that's running the 'file server' task.

Disk files can be shared, and you can even open a window on someone else's screen. The *SuperToolkit* network driver is just like the one on CST's new Thor XVI, reviewed in the August *Computer Shopper*. Other *SuperToolkit* commands let you rename and truncate files.

SuperToolkit's PRINT USING function lets you print numbers and text in strict formats; apart from *Mega-Toolkit*, the other programs let you format numbers, rounding them to a particular style and number of digits, but *SuperToolkit* allows the most comprehensive formatting.

SuperToolkit does not include error-trapping commands, but it has functions that let you check file and device names and contains corrections that improve the reliability of Sinclair's undocumented 'WHEN ERROR' trap, which is built in to later QL ROM versions.

Super variations

SuperToolkit is normally supplied on a chip, either as a plug-in cartridge to fit the ROM socket at the back of the QL, or as part of a disk system such as Sandy's SuperQboard of Miracle's Trump Card. Indeed, most disk systems contain some *SuperToolkit* commands – the *Toolkit* author, Tony Tebby, also wrote the disk control software used in the majority of QL interfaces.

The ROM packaging means that *SuperToolkit* is instantly available when you turn on your QL – as long as your ROM socket is not already occupied. The disadvantage, compared with the old Sinclair *Toolkit*, and competitors published on disk and microdrive, is that you don't get any example programs or utilities with *SuperToolkit*.

A special version, the 'configurable toolkit', runs in RAM and comes with a utility that lets you prune the toolkit to save memory, by excluding commands that you won't use.

We have not tested the configurable toolkit, but understand that the network handler will not work properly in RAM because it depends on precise instruction timings. As we explained in April, the speed of programs in RAM can vary wildly depending on the make and vintage of the memory.

The *SuperToolkit* manual is laser-printed, with dot-matrix additions. It's well-written in logical order, with examples and useful hints and tips. However, it doesn't warn you that memory allocated with ALCHP is released as soon as you load another program!

Despite its age, *SuperToolkit* is deservedly the most popular QL toolkit. It's worth having even if you're not a programmer, because it makes keyboard and file-handling much friendlier and more powerful. Programmers will also appreciate the extra commands and full-screen editor for BASIC, although *SuperToolkit* does not include many of the 'neat tricks' and low-level access routines in other toolkits.

Turbo-Toolkit

Turbo-Toolkit consists of a 6K kernel of 72 routines, concisely written in machine-code. Another 129 routines are supplied in the form of SuperBASIC source. This split minimises the amount of memory occupied by the Toolkit, which is particularly relevant when writing programs for unexpanded QLs. You must always load the 6K of machine-code into RAM, but you can pick and choose other routines individually from the source files.

The *Library Manager* program can extract named procedures and functions from SuperBASIC files. You specify the required routine names from the keyboard, or by reference to files of names, which the program can generate for you.

The source routines come in two large files, so you need the *Library Manager* to extract particular routines quickly. You can also use the program to hack named procedures or functions out of your own programs.

Turbo-Toolkit is best used with a compiler like *Supercharge* or its namesake *Turbo*. Compilation speeds up the routines supplied in source form, and some *Turbo-Toolkit* routines are only useful in multi-tasking programs. For instance, you must compile the 'TRACE' routine that monitors and displays the number of the SuperBASIC line currently running – it's no use unless it runs at the same time as a program you're testing.

Turbo-Toolkit includes unique BASIC debugging tools, including FIND, which locates and lists the first line of a named procedure or function, and HOW_COME which shows the procedure and function calls performed before an error, providing a useful trail backwards through faulty programs.

PROFILE must be compiled because it runs alongside an interpreted program or task, building a table showing the amount of time that was spent executing each group of lines. This is useful information when you're trying to speed up a program, because it shows you where the bottlenecks are.

Continued on page 94

Continued from page 91

Turbo-Toolkit lets you delve into the depths of the SuperBASIC interpreter. It has a job-lot of functions that let you read the interpreter's internal tables, even as they move around memory.

It has functions to trap errors when accessing files and devices, and when accepting numeric input. There are keywords to trap other errors, but these only work in programs compiled with *Turbo*.

Turbo-Toolkit comes with a three-part 'utility task'. Part One lets you design graphic character-sets on the screen, and comes with three sample fonts. The 'configurator' will customise the main toolkit to your choice of default device, pipe length, abort keys, and suchlike. The third part is a 'sound effect editor' which lets you experiment with the parameters of the BEEP command by moving slider controls on the screen.

Turbo-Toolkit is often used in commercial programs. It comes with a 5.5K 'runtime' version which you are allowed to distribute without restriction – the snag is that the runtime toolkit only works with compiled programs, and gives a 'not implemented' error if you try to use its commands from interpreted BASIC.

The *Turbo-Toolkit* manual is lucid but rather tattily printed in a mixture of dot matrix and daisy-wheel lettering, on coloured paper. It starts with a list of the machine-coded routines, then a 35 page discussion of them, in 15 small groups. Then comes a very comprehensive section discussion the utility task.

The last 20 pages document the 'demo' routines supplied in source form, explaining the purpose of each routine and naming other routines it calls, directly or indirectly.

Turbo-Toolkit seems aimed at SuperBASIC programmers rather than non-technical users. If you spend lots of time writing and testing BASIC it should serve you well. Digital Precision now only supplies *Turbo-Toolkit* with the *Turbo* compiler, but you can get the toolkit on its own from 'catalogue' suppliers like Super User Bureau.

Mega-Toolkit

This is the cheapest QL toolkit, and costs £12. *Mega-Toolkit* contains 130 procedures and functions, but they're rather unimaginatively chosen.

The majority of the *Mega-Toolkit* commands are direct calls to the QL operating system. Most system calls have a *Mega-Toolkit* equivalent, which is useful if you're learning about the QL. The other toolkits tend to tie these together to do complicated things automatically, but *Mega-Toolkit* leaves that to your SuperBASIC.

Default devices are recognised, but only by new *Mega-Toolkit* commands. You set the default device and drive number separately.

Mega-Toolkit supports wildcards, much like *SuperToolkit*. It uses the CP/M grammar, so an asterisk matches any sequence of characters and a question mark matches any one character. Names containing these symbols must be typed in quotes. WSTAT, WDEL and WCOPY can be used, with or without a 'query' option.

We found a few bugs. The RENAME command works with microdrives but not with disk systems. RECHP, meant to reclaim memory, crashed the system if it was given an unassigned variable as a parameter. LENJOB wouldn't find the length of job 0 (SuperBASIC), although it worked for other jobs; a separate function, LBASIC gave the correct length of BASIC work-space.

The *Mega-Toolkit* manual is much shorter than the others. We were supplied with a 13 page 'pre-production' photocopy that listed the commands and functions, with a syntax example and a short explanation of each. You'll have trouble using some of the routines unless you already own a book that documents QL operating system calls.

Mega-Toolkit comes on microdrive cartridge, 'protected' against theft. You can copy the files, but must put the original 'master' cartridge in drive 2 whenever you load the toolkit. For some reason this gave a consistent 'bad or changed medium' error, but the code still loaded and worked.

As well as the toolkit code, the cartridge contained a new character set and a program to load it, plus an eight line example of the way you can trap errors detected by *Mega-Toolkit* commands.

Mega-Toolkit is not as imaginative as the other toolkits, but it's still good value at £12, particularly if you can't afford disks or one of the £30 kits.

Mega Toolbox

Mega Toolbox is the latest QL toolkit, and has had surprisingly little publicity in view of its innovative features. Compware's main marketing ploy is an impressive demonstration cartridge which shows off the graphical power of *Mega Toolbox* – the demo costs £5, which you can set against the price of the full toolkit if you decide to buy it.

The demonstration is graphic in every sense – big patterns move around quickly, and intricate tricks are performed – but it doesn't tell the whole story. Many of the ideas in *Mega Toolbox* have been seen before, but there are enough good new ones to make it worth buying even if you already own *SuperToolkit* or *Turbo-Toolkit*.

Mega Toolbox improves the command line editor so that you can delete and move back and forth in steps of eight characters, and delete or move straight to either end of the line.

We did find one obscure problem – *Mega Toolbox* won't work properly with a Schoen keyboard if you've fitted Schoen's debounce chip. We reported the incompatibility to Schoen, which says that it is working on a new version of the chip. When the cure is ready it may also fix some Sinclair bugs in the co-processor, which controls the serial ports as well as the keyboard.

Mega Toolbox includes interesting multi-tasking commands, including an INPUT statement that lets you edit the next value while your program continues to run. The program can check the current value of the edited line at any time.

Other tasks repeatedly pan, scroll, recolour or animate the border of a window. You can also set up a slide-show that runs in parallel with the rest of your program. *Mega Toolbox* has lots of area graphics commands – it can save, pack, restore, mirror, rotate, merge and fade rectangles on the screen, at speed.

PAINT colours any area bounded by a line – it's nowhere near as fast as the standard FILL, but it can flood colour into any shape. The long-awaited COLOUR tells you the colour of any pixel in a window.

Mega Toolbox can split the screen into horizontal slices, displaying MODE 8 and MODE 4 simultaneously. It can 'zoom' in on pictures, re-scaling them. Even fractional scale factors are allowed – the results are a bit ragged, but still impressive.

MPRINT will print characters any size, but takes a couple of seconds to print five letters the size of the screen. PRINT-3D produces shaded lettering.

CHOOSE$ is not the only *Mega Toolbox* function that works with a list of values. COUNT and SEARCH find strings in other strings, while MIN and MAX return the lowest and highest values from a list.

Mega Toolbox includes string functions like RIGHT$, which returns characters from the end of a string, as in Microsoft BASIC. TRIM$ takes blanks off the end of a string, while REVERSE$("REVERSE$")="$ESREVER"!

PROMPT$ works like *Turbo*'s EDIT$, but *Mega Toolbox* does not include error-trapped versions for integer and decimal input. Curiously there are no error-trapping commands or functions in *Mega Toolbox*.

There are too many functions in *Mega Toolbox* to list them all; we've attempted to cover them in the table, but these deserve special mention: QCALL, QTRAP and QREG let you call machine-code and test returned values, even in compiled programs. FACT finds factorials, and STICK resolves joystick or cursor-key movements into co-ordinates.

The manual is laser-printed and equally clearly written. It summarises the routines by category, then lists in alphabetical order, with explanation and examples.

14 BASIC demo files illustrate the use of about 60 of the routines. Compware offers 'attractive' but un-stated terms to programmers who want to use *Mega Toolbox* code in commercial packages.

Mega Toolbox is a big, well-thought-out package which should appeal to many SuperBASIC programmers.

What's in a name?

It's possible to use several toolkits at once, but there are a few name-clashes, particularly between *SuperToolkit* and *Mega-Toolkit*. *Turbo-Toolkit* and *Mega Toolbox* are completely compatible with *SuperToolkit*, which is just as well given the number of disk systems that use at least part of the *SuperToolkit* repertoire.

Early QLs (versions 'AH' and 'JM') use the first definition of a particular command, if there's a clash – later versions use the last. You can get around most problems by loading toolkits in a particular order, or by patching the names in the toolkit code file to make each name unique. Replacement names must be the same length as the original.

You may run into trouble if toolkit commands use the same names as routines or variables in your old programs. The toolkit definition always over-rides a SuperBASIC program definition. You have to edit the program or the toolkit before they can get along together.

Contact

Compware: 57 Repton Drive,
Haslington, Crewe CW1 1SA

Digital Precision, 222 The
Avenue, London E4 9SE

Utilising Software: 75 Oakhanger
Drive, Lawrence Weston, Bristol
BS11 OJR

Q-Jump: 24 King St, Rampton,
Cambs CB4 4QD