==========

Q E D

A QL Programmer's Text Editor

==========

Version 2.03 - May 2021

Hilversum, The Netherlands

LICENCE INFORMATION

As of 2018, QED is distributed under the GNU Public Licence v3. The released versions, source code and terms and conditions of licence can be found at:

https://github.com/janbredenbeek/QED

Feedback to the author can also be given at this site, or via e-mail: jan@bredenbeek.net.

1. INTRODUCTION

QED is a text editor for the Sinclair QL and other QDOS-compatible systems. It is intended for editing of line-based text files, such as assembler or C source files and the like.

QEDs design was largely based on existing QL text editors, but the main design consideration of QED was to be faster and more compact. For this reason the whole of it was written in assembly language, and this resulted in a program that, despite its power, only occupies about 11K bytes of code. This means that on an unexpanded QL you have more room for your text files. If you have expansion RAM and floppydisks, you will be able to edit files of hundreds of K's with QED while loading and saving still takes only a few seconds to complete.

It must be stressed that QED is only a text editor and **not** a full-blown word processor! Although you can use QED very well to edit "human readable" text, you will not find markup features and similar facilities found in word processors like QUILL and others. However, the present version of QED has some provisions for paragraph-oriented text files, where 'long' lines represent a paragraph of text, to be formatted so they will fit into the window and can be read and edited easily.

Features at a glance:

* Simultaneous editing of an "unlimited" number of files - "unlimited" means only limited by the amount of RAM you have!

* Full TAB-support - saves lots of space(s) in source files.

* Supports line lengths up to 254 characters. Longer lines will be soft-wrapped on word boundaries, optionally at the set right margin.

* Can read and write MSDOS/Windows text files (CR+LF line ends).

* Powerful command language allows you to perform repeated operations using a single command line.

* AutoIndent feature allows quick entry of program source text without having to use the TAB key on each new line.

* Full EX/EW command string support; allows easy startup by other programs.

* Recognises Toolkit II default directories.

* Fast, efficient screen output, taking advantage of screen accelerators such as SpeedScreen and Lightning.

* Supports high-resolution screen sizes offered by modern platforms such as QPC2, SMSQmulator and Q68.

* Carefully written - can be HOTKEYed without having to care about guardian windows, memory sizes or "impure" code. Fully ROMable.

2. GETTING STARTED

2.1. Installation

Since QED is "freeware", you may have obtained it in different ways. When you have downloaded it from a BBS or Internet, it will usually have been packed into a _zip file or something similar, but when it came on disk you might find the files separately on it. In the latter case, the files comprising the QED package can simply be copied to your daily work disk or directory. When QED came in archived form (_zip, _arc, _zoo file etc.), you have to unpack the files contained within the archive file into your work disk or directory.

The QED package consists of four files:

QED	The QED program itself, which is EXECable;
QED_HELP	The QED helpfile, displayed on pressing F1;
QED_DOC	This documentation;
CHANGES_TXT	Text file describing changes in versions;
QEDCONFIG_BAS	A SuperBASIC program to configure the QED default settings.

The QEDCONFIG_BAS program allows you to change many default settings of QED such as window position and colours, text margins etc. These settings will be written directly into the QED executable file itself, so I strongly recommend you to use a working copy for configuring your personal preferences and store your unconfigured original in a safe place.
A detailed description of the configuration process will be given further on in this manual. In most cases, the pre-configured default settings will be adequate enough so you can safely make a quick-start now.

2.2. Simple Startup

The simplest command to start QED from within SuperBASIC is:

EXEC_W <device>QED

(where <device> stands for the name of the drive containing QED, e.g. mdv1_ or flp1_ depending on your system).

In this case, QED will ask for the name of the file to be edited, and the size of the workspace to be used. In most cases, you can safely skip the "Workspace size" prompt by pressing ENTER, which will use a default workspace size depending on the file size.
If the file you specified is an existing file, it will be loaded and displayed ready for editing. If the file did not exist previously, you will see a blank editing window and a message "Creating new file" at the bottom.
Please refer to Chapter 3 now in order to edit the file.

Notes:

* On systems equipped with Toolkit II, the shorter **EW** command (synonimous for EXEC_W) can also be used. Furthermore, the default data directory will be used if no device and/or directory has been specified in the file name. For example, if your default data directory (set by the **DATA_USE** command) is **flp2_** and you have entered **fred_c** as the file name, QED will use **flp2_fred_c** as the file to be edited.

* QED can also be started using the **EXEC** and **EX** (Toolkit II only) command. In that case, SuperBASIC will continue running and you can switch away from QED using the CTRL-C key. If the Pointer Environment has been installed, the QED window will be saved and restored as you would expect. If you do not have the Pointer Environment installed, the **F5** key can be used to redraw the QED window.

2.3. More complex Startup - Using a Command String

The Toolkit II **EX** and **EW** commands allow you to specify a Command String to a program to be executed. QED will recognise this command string and can take the filename and several default startup options from it. Specifying a filename in the Command String will by-pass the initial filename and workspace prompt, and will thus put you directly into the editing screen.

The Command String should consist of the name of the file to be edited, followed by zero or more options, each separated by one or more spaces. An option always starts with a dash ("**-**") or slash ("**/**") sign, followed by a single letter and optionally an argument, which may be a number or + or - sign, depending on the option. Note that there should be **no** spaces between options and arguments.

The following options are possible:

* **-B**n specifies the workspace size to be used, in "n" Kilobytes. If this option is not given, the default buffer size will be used (see further on how QED determines the default buffer size).

* **-C** specifies that the workfile should be read in "Cooked" mode rather than "Raw" mode. This means that QED will read the file line by line and preprocess each line before storing it. This is useful when you have a file containing Carriage Return characters at the end of each line, or want to compress or expand TAB characters as the file is being read. CR characters at the end of each line will be removed, and TABs compressed or expanded according to the setting of the "Expand TABs" and "Compress TABs" options (see further on).
  As QED has to do much more processing on the workfile when reading in in "Cooked" mode, loading will be slower than in "Raw" mode. However, you usually do not need to load a file in "Cooked" mode more than once. When CRs are removed and TABs converted, you can save the file and load it back again in Raw mode.

* **-E** Expand TABs into spaces (**-E+**) or not (**-E-**). The number of spaces replacing one TAB character may be set using the **-I** option.

* **-I**n specifies the TAB distance in columns.

* **-O**n specifies the workspace overhead in "n" Kilobytes. The workspace overhead is the amount of memory added to the file size to calculate the workspace size used when loading an existing file. Note that this option has no effect when creating a new file.

* **-T** Compress multiple spaces into one TAB character (**-T+**) or not (**-T-**). The number of spaces per TAB may be set using the **-I** option. Note that spaces enclosed in single or double quotes will **not** be compressed. In the current version, when both TAB expansion and compression are enabled, TAB compression takes precedence.

* **-W** Enables (**-W+**) or disables (**-W-**) Word Wrap.

Examples:

```
EX QED;"john_c"             (loads file john_c)
EX QED;"fred_asm -B16"      (loads file fred_asm using 16K workspace)
EX QED;"mary_c -O4 -I3 -C" (loads file mary_c in Cooked mode using 4K
                            workspace overhead and 3 columns TAB
                            distance).
EX QED;"joe_pas -C -E+ -W-" (loads file joe_pas in cooked mode, expanding
                            TABs and no Word Wrap).
```

2.4. Editing Multiple Files

QED can edit multiple files simultaneously, although only one file will be visible in the QED window at the same time. Pressing the **F5** key will switch the display to the next file being edited, in a round-robin fashion. When multiple filenames are specified in the command string, QED will load all files into memory and display the first file specified ready for editing. If an error occurs when attempting to load a file, QED will display an error message and wait for you to press a key, after which it will continue with the next file.

When loading multiple files, it is also possible to specify options for each file to be loaded. The options should be specified after the filename and before the name of the next file. E.g.:

```
EX QED;"john_c fred_asm -B16 mary_c -O4 -T3 -C"
```

will load the files john_c, fred_asm and mary_c as described in the previous section, but this time all three files at the same time.

There is no limit on the number of files that QED can handle, other than the limit imposed by the amount of available RAM.

2.5. How QED calculates the workspace size

QED uses a workspace of fixed size to hold the entire file being edited. While using a fixed-size buffer does have the drawback of less flexibility,

it has the advantage of causing less heap fragmentation and enabling fast loading and saving of files.

When loading a file, it is possible to specify the size of the workspace to be used, either using the "Workspace size" prompt (when no filename has been given in the command string) or the **-B** option in the command string. In both cases, the number specified will be taken as the workspace size in **kilobytes**.
If a workspace size has been given that would exceed the amount of available RAM, you will be prompted again for the workspace size (in case you replied to the "Workspace size" prompt) or loading will be aborted with an error message (in case you used the -B option).

There are two values involved in determining the workspace size when an explicit workspace size has **not** been given: the **default workspace size** and the **workspace overhead size**.
The default workspace size comes in when no explicit workspace size has been given and the workfile does **not** exist, i.e. is being created. In such cases, QED will allocate a workspace of the default size, as specified using the QED Configuration Utility. This value is pre-set to 12K in the distributed version.
If the workfile **does exist** and no explicit workspace size has been given, QED will calculate the size of the workspace needed by adding the workspace overhead size to the length of the workfile. Thus, when the workfile has been loaded, you can add as many characters as the overhead size before the workspace overflows. The overhead size can also be adjusted using the QED Configuration Utility, the pre-set value being 4K.

In parallel to the workspace, QED will also allocate room for a **line-length table** so that it can keep track of the workfile in a fast and easy way. This table occupies 1 byte per line and is automatically built after loading the workfile.
As the workfile is being expanded, it is possible that this table will overflow and needs to be re-allocated. This is done automatically and virtually transparent to the user, who will usually only notice a small delay when a new line is being inserted. The only exception to this will be when there is so little memory left in the machine that it is impossible to expand the line-length table any further.
In this case, an error message "**No room for line table**" will be displayed and further editing will be prevented. However, it will still be possible to save the contents of the workfile without any loss of data.

3. USING QED

3.1. General appearance

The QED window is divided into two areas: the upper region (all except the bottom line of the window) is the "text area" where your editing takes place, and the bottom line of the window is used to show the command line, messages and status information.
The status line consists of, from left to right:

* The current line ("L: nnnnn"), starting at 1 from the top line;

* The current column position ("C: nnn"), starting at 1 from the leftmost column;

* The number of lines currently in the workfile ("N: nnnnn");

* Five letters indicating the write mode (Insert/Overwrite), the AutoIndent state, the TAB Expansion and TAB Compression state, and the Wordwrap state respectively. The first letter is either an 'I' or 'O' for Insert and Overwrite respectively, and the others show 'E', 'C', 'A' and 'W' when their respective settings are turned on.

* The name of the file being edited. If any changes have been made during the current edit session, this name will be preceeded by an asterisk character ("*").

As of version 2.03, QED has some provisions for editing paragraph-oriented text files. These are files where a physical line in the file (terminated by LF or CR+LF) represents a paragraph of arbitrary length. If the length of a paragraph is longer than the display width, it may be shown either by a single line or multiple lines, depending on the setting of the Left and Right Margin and Word Wrap. When Word Wrap is not active, paragraphs will be shown as a single line and the display will be panned horizontally as you move the cursor along. When Word Wrap is active, QED will display paragraphs on multiple rows according to the setting of the Left and Right Margin. Note that you may set the margins wider than QED's window width, in case of which you will see only part of the displayed rows and have to use the cursor keys to pan the display.

There is a limit of 254 characters that QED can display on a single window row. Paragraphs longer than this limit will be split at word boundaries, and a message 'Long lines split' displayed upon reading in. This doesn't affect the physical representation of a paragraph until you edit it, and it will retain its original structure when saved back.

The ability to edit paragraph-oriented text files is a feature currently under development, and subject to future enhancements (including display of physical end-of-paragraph markers).

3.2. Entering text

QED can operate in two write modes: **Insert** and **Overwrite**. In Insert mode, any character you type on a line will be inserted at the current position, and any characters further on the line will be moved to the right. Pressing ENTER will split the current line, moving any characters righthand of the cursor to a newly inserted line below the current line.
In Overwrite mode, the character at the cursor position will just be overwritten by the new character you type. Pressing ENTER will only move the cursor to the start of the next line. However, if you type beyond the end of the line the characters you type will just be appended to the line, and pressing ENTER on the last line of the file will generate a new line, just as with Insert mode.
To change the mode, press the **F4** key.

QED supports automatic **wordwrap** at the end of a line. If you reach the right margin of the text (which does not necessarily have to be the same as the rightmost column of the display!) and the Wordwrap option has been turned on, QED will perform an automatic newline and places the whole of the partly completed word onto a new line. Also, it is possible to specify a left margin, to which the cursor will move when you reach a new line (either due to an ENTER keypress or a wordwrap).

If the **AutoIndent** feature of QED has been enabled, then, on reaching a new line due to an ENTER keypress or wordwrap, the cursor will be automatically placed onto the same column where the first non-blank character of the previous line (skipping any blank lines) appeared. This feature is useful when editing files containing source text in programming languages such as C and Pascal. In most cases, you don't have to manually indent each line now.
Note that setting AutoIndent on disables any Left Margin which has been set previously!

When you edit a line, QED firstly copies it into an internal buffer. Any changes you make to the line take place in that buffer, not in the text file itself. You can restore the line to its pre-edited state by pressing the **ESC** key. However, if you move the cursor to another line the newly edited line will be placed back into the text file, thus replacing the original line.

3.3. Cursor control

The cursor can be moved one position by the cursor control keys **LEFT**, **RIGHT**, **UP** or **DOWN**. The display will be scrolled horizontally or vertically if necessary. However it is not possible to move the cursor beyond the start of the line, the 254th column, or the start or end of the text file.
The **SHIFT LEFT** and **SHIFT RIGHT** combination of keys will move the cursor to the start of the previous or next word respectively, while the **ALT LEFT** and **ALT RIGHT** keys will move the cursor to the start or the end of the line respectively.
The **TAB** key moves the cursor to the next tab position on the line (which is a multiple of the tab distance). In Insert mode, the characters after the cursor are also moved rightwards.

The **SHIFT TAB** key moves the cursor back to the previous tab position. In Insert mode, this will also delete characters between the cursor position and the previous tab stop.

Note that you can move the cursor beyond the last character of the line, but this will not generate extra spaces at the end of the line. However, if you type a character at this position, spaces will be inserted automatically between the previous end-of-line and the new character.

The **ALT UP** and **ALT DOWN** keys scroll the text one line up or down, while the cursor position on the screen will not move. The **SHIFT UP** and **SHIFT DOWN** keys move the cursor one page up or down, which is useful for quickly moving through the text.

3.3.1. Using PC keyboards

When using a PC keyboard, the HOME and END keys may be used to set the cursor at the start or end of a line, and the Page Up and Page Down keys may be used to quickly scroll through the text one page at a time. On some emulators and keyboard interfaces (e.g. the Jurgen Falkenberg one), you may find however that they do not perform the desired action. In that case, the logic of the SHIFT and ALT combination of keys may be reversed by setting the 'Swap SHIFT ?? / ALT ?? keys' option in QEDCONFIG_BAS to ON (See section 4 for configuring QED).

3.4. Deleting text

The **CTRL LEFT** and **CTRL RIGHT** keys delete one character to the left or right of the cursor. The **SHIFT CTRL LEFT** and **SHIFT CTRL RIGHT** keys delete one word to the left or right respectively.
The **CTRL ALT LEFT** key deletes the whole of the current line, while the **CTRL ALT RIGHT** key erases the current line from the cursor to the end.

A CTRL LEFT or SHIFT CTRL LEFT keystroke at the first column of the line will actually delete the "newline" at the end of the previous line: the current line will be appended to the previous line. Note that this will not work on the first line of the text, or if it would generate a line longer than 254 characters.

3.5. Function keys

The function keys F1 to F5 have the following function:

F1: Display HELP information. This is read from a HELP file, so this has to be present on the appropriate medium. For example, if QED uses flp1_QED_HELP as the help file, then the disk containing QED_HELP must be in drive 1. If QED cannot find the HELP file, it will display an error message.

F2: Re-execute last command line (see section 3.6).

F3: Allows you to enter and execute a command line (see section 3.6).

F4: Change Insert/Overwrite mode.

F5: Switch to the next workfile in the chain of files (if any), and redraw the QED display.

F6 (or SHIFT F1) : Toggle the AutoIndent state.
F7 (or SHIFT F2) : Toggle the Wordwrap state.
F8 (or SHIFT F3) : Edit the last command line entered.
F9 (or SHIFT F4) : Toggle the TAB Expansion state.
F10 (or SHIFT F5): Toggle the TAB Compression state.

3.6. Commands

Pressing F3 allows you to enter a command line. This consists of one or more commands, terminated by an ENTER keypress. While entering the command line the normal QDOS line editor keys (LEFT, RIGHT, CTRL LEFT and CTRL RIGHT) may be used. The length of a command line may be up to 255 characters, although the display will not allow you to see this number of characters at one time.
Entry of a command line may be aborted by pressing the arrow-up or arrow-down key. On systems equipped with a Minerva ROM, the ESC key may also be used. The old command line will then be retained.
You may recall the last command line by pressing SHIFT F3. This allows you to edit the command line again; pressing ENTER will execute it.
Pressing F2 will immediately re-execute the last command line entered; useful if you want to repeat a set of commands after looking at the text.

A command consists of the command name (one or two letters), possibly followed by a numeric or string parameter. A numeric parameter must be a decimal number in the range 1 to 65535 inclusive (there are currently no commands which accept a zero parameter).
A string parameter is a sequence of characters, possibly starting and ending with a delimiter character. A delimiter character must be used if the string contains spaces, semicolons or parenthesis characters or starts with a non-alphanumeric character. Delimiter characters must be non-alphanumeric and may not be a semicolon (";") or brackets.

Examples of valid strings are:

```
john
/fred bloggs/
'#1/2'
!(wombat)!
"Hello;world!"
```

Multiple commands on a single command line must be separated from each other by a semicolon ";". E.g. SA flp1_myfile_asm; Q will save the current file to floppydisk 1 under the name "myfile_asm" and then quit QED.

A command may be executed repeatedly by specifying a repeat count before the command name. E.g. the command 4N will move the cursor four lines down in the text file. The repeat count must be in the range 1 to 65535.
A special case of this is the RP specifier: this will repeat the following command indefinitely. E.g. RP E/mdv/flp/ will change all occurrences of "mdv" subsequently found to "flp".
Note that, regardless of the repeat count, a command will always be terminated when an error condition occurs or the ESC key is pressed. This will always return you to the editing mode.

Finally it is possible to combine groups of commands using brackets. A repeat count before the opening bracket will then repeat the group of commands rather than a single command. The commands within a command group may contain their own repeat count, and a command group may also contain further nested command groups. The following (quite silly) example demonstrates this:

RP(4N;12(20P;19N)) will move the cursor position back and forth within the text until it reaches the top or end of the file or you press ESC.

3.7. Description of Commands

3.7.1. Cursor control

There is a set of commands for moving the cursor. Some of these commands are only useful if used in conjunction with other commands and/or repeat sequences, as their effect can also be achieved by using the normal cursor control keys.

The **CL** command moves the cursor one position to the left, while the **CR** command moves it one position to the right.

The **NW** command moves the cursor to the start of the next word, the **PW** command moves it to the start of the previous word.

The **CS** command moves the cursor to the start of the line, the **CE** command moves it to the end of the line.

The **N** command moves the cursor to the next line in the text, the **P** command moves it to the previous line.

The **T** command moves the cursor to the top of the file (first line), the **B** command moves it to the bottom (last line).

The **M** command, which must be followed by a line number (starting at 1 for the first line), moves the cursor to a particular line in the text.

The **RT** command can be used to Return to a particular line. When you issue any command that moves the cursor away from the current line (except N or P), QED stores the current position automatically so that you can return to it later. This is useful if you temporarily want to look at some text elsewhere in the file (for example to check procedure parameters) but want to return to the current position afterwards.

Notes:

1. QED only remembers the last three positions in the file stored. This means you can, for example, use the M command three times, whereafter three subsequent RT commands will take you back to the former position. But if you do another RT, QED will move to the start of the file as it can no longer remember the "Return line number" which was in use at the time the M commands were issued.

2. In the present version of QED, the stored line numbers will **not** be updated whenever one or more lines are inserted or deleted **before** the stored position. This means that RT will not take you to the correct line in this case. However, if only a few lines are inserted or deleted, the position returned to by RT will not be far away from the original position.

3.7.2. Altering text

The **TY** command enters the characters of its string parameter into the text as if they were TYped from the keyboard. E.g. TY/fred/ will enter the characters "fred" into the text at the cursor position, using insert or overwrite mode as appropriate.

The **DC** command deletes the character under the cursor, in the same way as the CTRL RIGHT key does. The **DW** command deletes the word right from the cursor, as the SHIFT CTRL RIGHT key does.

The **I** command, which may be followed by a string parameter, inserts a new line containing the text of the string into the text between the current line and the previous line. If no string parameter is given, an empty line will be generated.

The **A** command is identical to the I command, except that the new line will be inserted between the current line and the line after it.

The **D** command deletes the current line, moving up all lines below it.

The **J** command joins the current line with the next, appending the text of the next line to that of the current. This will not work on the last line of the text, or if it would generate a line longer than 254 characters.

The **S** command splits the current line at the cursor position, moving the text from the cursor onward onto a new line.

3.7.3. Setting margins and tab distance

The **SL** command sets the left margin equal to the value of the numeric parameter (which must be from 1 upwards). The left margin is the column position to which the cursor will be set after an ENTER keypress or an A, I or S command. In Overwrite mode, the characters left from the left margin will not be erased. It is still possible to move the cursor to the left by using the cursor control keys.
Please avoid setting the left margin to daft values; it should certainly not be set beyond column 254.
Note: If AutoIndent mode has been turned on, QED will ignore any left margin set!

The **SR** command sets the right margin equal to the value of the numeric parameter. When you reach the right margin during typing in of the text and Wordwrap has been enabled, QED will automatically perform a newline and move any partially completed word to the next line. Note that this will **not** happen if your typing starts already beyond the right margin, or if the character you type is not the last on the line.
The state of the automatic line wrap can be toggled by pressing **SHIFT F2** or the **TW** command, and is displayed on the status line using the letter "W". In the same manner, pressing **SHIFT F1** or using the **TA** command toggles the Auto Indent state.

The **ST** command sets the tab distance equal to the value of the numeric parameter. When a TAB keystroke is received, QED will insert or overwrite

one or more spaces until it reaches the next multiple of the tab distance. However, when the TAB Compression feature is active (toggled by the **TC** command), these spaces will be replaced by TAB characters when the file is saved.
Loading a file containing TAB characters will not cause them to be expanded to spaces, unless the TAB Expansion feature has been enabled
The distance specified must be in the range 1 to (at most) 253.

The left and right margin and tab distance are initially set to default values. These default values can be configured with the QEDCONFIG_BAS program.

3.7.4. Searching and replacing text

The **F** command searches the text from the cursor position onwards for a target string. If a string parameter is given, this will be taken as the target string, if no parameter is given the target string specified in the last F or BF command will be used. E.g. F/wombat/ will search the text for the string "wombat". Note that the case (upper of lower) of the string is ignored and the surrounding characters are of no importance.
The **BF** command is the same as the F command except that the search will be made from the cursor position backwards into the text.
The **F** and **BF** commands have two immediate versions: the **CTRL DOWN** and **CTRL UP** keys respectively. CTRL DOWN will search forwards for the string given in the last F or BF command, CTRL UP will search backwards for it.
It should be noted that the F command moves the cursor one position forwards before the search, and BF moves the cursor one position backwards before the search. Hence, if the cursor is already positioned at the start of the target string, the F or BF command will skip to the next or previous occurrence respectively. This is useful if you want to search quickly for a particular occurrence; simply press the F2 or CTRL UP/DOWN key if the occurrence found is not the one you want.

The **E** and **EQ** commands exchange (replace) a particular string by another. The E command exchanges immediately, the EQ command queries first on finding an occurrence (press Y if you really want to exchange).
Two strings must follow, separated by a single delimiter character. E.g. E/cat/mouse/ will replace the next occurrence of "cat" by "mouse". If you want all occurrences to be replaced, use the commands T;RP E/cat/mouse/. If you do not respond with "Y" on an EQ query, the command will be terminated but the command line will not be aborted unless you pressed ESC. Thus if RP is specified EQ will skip to the next occurrence.
The E and EQ commands follow the same rules for searching as the F command. The first string specified will become the new target string.

3.7.5. Block commands

A block of text consists of one or more complete lines. The start of a block is defined by moving the cursor to the desired line, and then issuing the **BS** command. The end of a block is defined in the same way using the **BE** command. The block is then defined to be the text from the start of the "BS" line up to and including the text on the "BE" line.

If any line is inserted into or deleted from the text (except by a IB command), the block start and end become undefined once more.
Note that the start and end of the block do not have to be specified in strict order, so it is possible to specify first the end and then the start of the block. The BS and BE commands do not check whether the block end is after the block start, the validity of the block is only checked when the next block command is issued.

The **IB** command inserts a copy of the block between the current line and the previous line. It is not possible to insert a block within itself (use IB within copies if necessary).

The **DB** command deletes the block from the text file. The cursor is set at the position where the block has been deleted.

The **WB** command writes the contents of the block to a file, the name of which must be specified in the string parameter. If the file already exists, you may overwrite its contents with the new block or append it to the existing contents, or simply cancel the command.

The **SB** command sets the cursor at the first line of the block and at the top of the display. Note that the RT command may be used to return to the old position.

3.7.6. File commands

In all file commands, any filename specified can be either a full QDOS file name (e.g. flp2_fred_asm) or, if Toolkit II has been installed, a file name without directory (i.e. just "fred_asm"). In the latter case, the default data directory set by the SuperBASIC **DATA_USE** command will be assumed.

The **SA** command saves the text to a file. If no parameter is given, SA will use the current workfile name. If a string parameter is specified, this will be taken as the file name, but the current workfile name will not be altered.
If the file to be saved already exists, the old version will be renamed by appending the extension **_OLD** to the filename. This will take up more disk space, however it is much safer than just overwriting the old file with the new. Firstly, it allows you to recover the old version and secondly, if anything goes wrong during the save operation (power failure!) you will have at least the old version which can be recovered.

Example:

SA myfile_asm writes the text to file "myfile_asm", thereby renaming any existing file to **myfile_asm_old**.

The **SC** command saves the file as with SA, but with each line terminated by a Carriage Return/Linefeed combination of characters, rather than the single Linefeed character normally used to terminate lines in QDOS text files. This is useful when you want to port the file to a different operating system, such as MS-DOS where textfiles contain lines terminated by CR and LF.

The **R** command allows you to start up editing of a new file. The string parameter must contain the name of the file and startup options if any (see section 2 for a description of options). If no parameter has been given, you will be prompted for the new file name and workspace size (as discussed in section 2).
When the file has been successfully loaded, it will be presented ready for editing. Note that, unlike previous versions of QED, this command will **not** quit editing of the file being edited when the R command was issued, but simply append the new file to the list of files being edited! Pressing **F5** will switch the display to the next file in the chain.

The **RC** command works like the **R** command, except that the file is loaded in 'cooked' mode. Each line will be processed separately by removing the CR from CR/LF line endings (if any) and expanding or compressing TABs according to the TAB Expansion and TAB Compression settings. This command may be useful to load DOS or Windows files or files containing TAB characters.

Note: If you want to specify options using the **R** and **RC** commands, separated by spaces, you **must** enclose the argument by separator characters, e.g. **'RC/example_txt -O50/'**, else the option would be ignored. Also, file names containing spaces cannot be specified directly. In that case, use the **R** or **RC** command with no argument and specify the file name at the interactive prompt that follows.

The **IF** command inserts the contents of the file specified in the string parameter at the current position. The file will be inserted between the current line and the previous line. If there is not enough room in the workspace, an error message will be generated. In that case, you will have to re-start QED with a larger workspace size.
After the file has been loaded, there may be a few seconds delay during which QED is rebuilding an internal table.

3.7.7. Reformatting text

QED version 2.03 and later allows you to reformat the text to fit into the window, preserving the paragraph-structure of long lines. This will be actioned when reading files, either raw or cooked, when the Word wrap option is enabled. When Word wrap is disabled, lines longer than 254 characters will also be split over multiple screen lines, but still at word boundaries. A newline character will **not** be inserted at the split point, even if it appears to the user as multiple rows on the screen. Thus, when the file is saved back, paragraphs will retain their original structure.

However, when you edit a file, paragraphs are not automatically reformatted as you type. Thus, inserting a few words in a line may move the end of the line off the right edge of the window, losing the original formatting. To make the text fit again into the window, you can use the **RF** (reformat) command. This does not change the text by itself, but merely rebuilts the line table so that all lines fit into the window and redisplays the part of the text currently in the window.

3.7.8. Miscellaneous commands

The **Q** command quits all files in QED without saving the text. If any changes have been made to the text, QED will ask confirmation. Press Y if you want to leave QED, losing the changes.

The **QF** command quits the current file being edited. Confirmation will be asked if changes have been made. If the file quit was the only file being edited, an exit will be made from QED as with Q, otherwise the display will be updated with the next file.

The **X** command exits all files, writing the text back to the file if changes have been made (in the same way as with SA) and then quits QED. If any file error occurs (not found, drive full etc.), QED remains running.

The **XF** command does the same as X, but only with the current file being displayed. An exit from QED will be made if this file was the only one being edited, otherwise the next file will be shown.

The **U** command cancels any changes made on the current line. It does in fact the same as the ESC keypress.

The **SH** command displays the current status of QED. This includes the name of the workfile, the current string used for the F and BF commands, the TAB state and distance, left and right margin, block start and end line, and the workspace size and usage.

3.8. Error messages

If during editing or command execution an error occurs, QED will display an appropriate error message. In most cases this is self-explaining, but two cases might need further explanation:

"No room for text"
There is not enough room in the editing workspace left for what you are trying to do. You must save the text first (using SA) and then reload it using the R command (which allocates a new workspace large enough for the file plus 4K overhead).

"No room for line table"
QED maintains an internal table to keep track of the length of each line. When this table becomes full, QED rebuilts it using a larger table size. This is normally transparent to the user, but if there is insufficient memory in the QL to do it then QED will display this message and inhibit any further editing. You should save the text using the SA or X command, then do something to get more free memory, and then reload the file.

4. CONFIGURING QED

QED allows you to re-configure many of its startup parameters, such as window size and position, display colours, default margins etc. This can be done using the QEDCONFIG_BAS configuration program.
The configuration program is started by entering the SuperBASIC command:

LRUN <device>QEDCONFIG_BAS

(where <device> is mdv1_, flp1_ etc. as appropriate).
Once the program has been loaded, it will ask for the name of the medium containing the copy of QED to be configured. E.g. if you have a copy of QED in microdrive 1, you must enter mdv1_, and so on.
The configuration program will then read and display the currently installed values. You can then select the parameter to be altered with the UP and DOWN keys, and modify its value with the LEFT and RIGHT keys. The LEFT key will decrease the value by one, the RIGHT key will increase it. An exception to this is the name of the HELP file: pressing LEFT or RIGHT allows you to enter the new HELP file name, which must be terminated by an ENTER keypress.

The following parameters can be configured:

- The left and right margin and the tab distance to be used on startup, or after a R command.

- The initial workspace size in Kilobytes. This is the workspace size used whenever the default is specified on startup or the R command is used, and the workfile does not exist.

- The workspace overhead in Kilobytes. This is the **extra** workspace that will be allocated when an existing file is loaded.

- The colour and width of the border of the QED display.

- The PAPER and INK colour of the initial display (the display used on startup).

- The PAPER and INK colour of the text window.

- The PAPER and INK colour of the status and report line.

- The PAPER and INK colour of the command line.

- The horizontal character size. This corresponds to the first parameter of the SuperBASIC CSIZE command: 0 for 6 pixels, 1 for 8 pixels, 2 for 12 pixels and 3 for 16 pixels. Note that the command and status line must at least contain 55 characters, so the horizontal character size of this line will never be greater than 8 pixels, even if you specify 12 or 16 for the text window. Also, QED always uses MODE 4 for its display, switching the MODE on startup if necessary.

- The defaults for Insert/Overwrite, word wrap, auto-indent, TAB expansion and TAB compression when QED is started up.

- Swap SHIFT ?? and ALT ?? keys. This will reverse the function of the cursor keys in combination with SHIFT and ALT. On some PC keyboards, the Page Up and Page Down keys generate key codes corresponding to the ALT UP and ALT DOWN keys respectively, whereas they really should generate SHIFT

UP or SHIFT DOWN. When setting this option to ON, you can still use the Page Up and Page Down keys for the function they're supposed to do.

- The name of the HELP file. This is the file displayed when F1 is pressed. Note that this is just a normal text file that can be edited with QED, so you can easily create your own HELP file and install it using this configuration program. Is is best however to avoid lines longer than 55 characters, as QED might not be able to display them fully (the HELP display has no end-of-line wrap).

For the PAPER and INK parameters, consult the QL User Guide if necessary. For the PAPER (background) colour a range of 0 to 255 is valid, for the INK (foreground) colour only 0 to 7 as "stippled INK" text is somewhat difficult to read! The configuration program relies on your own confidence on the choice of colours, so don't use silly combinations like green INK on white PAPER and the like. On legacy QL hardware, QED will switch to 4-colour mode in order to have enough text columns available. On platforms supporting more colours, up to eight colours are supported.

When you have finished, press ENTER to continue with the window size and position configuration. A window is then displayed with colours, border, size and position equal to those of the window to be configured.
The window can be re-positioned with the LEFT, RIGHT, UP and DOWN keys. One keypress will move it two pixels in horizontal direction or one in vertical direction. The window can be re-sized with the ALT LEFT, ALT RIGHT, ALT UP and ALT DOWN keys. One keypress reduces or enlarges the window by one character column or line as appropriate. The minimum size is 55 columns and 5 lines. The maximum size depends on your particular screen size; on a standard QL you will have a maximum of 85 columns by 25 lines but QED also supports higher resolutions offered by emulators running SMSQ/E such as QPC2.

Any change you make will be displayed; also the current window width and height in **characters** (not pixels) will be displayed, with the pixel coordinates of the top left-hand side below it.
When you are satisfied with the window size and position, press ENTER. The program then finally asks whether you want to install the new parameters or not. Press Y to install them or N if you want to quit without installing. You may optionally specify a QED program file name different from the original. This allows you to transfer the settings between different copies of QED, e.g. when upgrading to a new version.

APPENDIX 1: COMMAND SUMMARY

A1.1. Immediate commands

TAB	Tabulate (i.e. insert/overwrite a number of spaces)
ENTER	Split and generate new line (insert mode)
	Move cursor to new line (overwrite mode)
LEFT	Move cursor left one character
RIGHT	Move cursor right one character
SHIFT LEFT	Move cursor left one word
SHIFT RIGHT	Move cursor right one word

CTRL LEFT	Delete left one character
CTRL RIGHT	Delete right one character
SHIFT CTRL LEFT	Delete left one word
SHIFT CTRL RIGHT	Delete right one word
ALT LEFT	Move cursor to start of line
ALT RIGHT	Move cursor to end of line
CTRL ALT LEFT	Delete line
CTRL ALT RIGHT	Erase to end of line
UP	Move one line up
DOWN	Move one line down
ALT UP	Scroll one line up
ALT DOWN	Scroll one line down
SHIFT UP	Move up one page
SHIFT DOWN	Move down one page
CTRL UP	Search backwards (see "BF" command)
CTRL DOWN	Search forwards (see "F" command)
ESC	Restore line to its pre-edited state
F1	Display HELP information
F2	Re-execute last command line
F3	Enter and execute command line
F8 / SHIFT F3	Re-edit and enter last command line
F4	Toggle Insert/Overwrite mode
F5	Redraw display
F6 / SHIFT F1	Toggle AutoIndent mode
F7 / SHIFT F2	Toggle WordWrap mode
F9 / SHIFT F4	Toggle TAB Expansion mode
F10 / SHIFT F5	Toggle TAB Compression mode

A1.2. Extended commands

In the following:
n indicates a number in the range 1 to 65535;
s indicates a string optionally starting and ending with a delimiter character;
/s/t/ indicates two strings separated from each other by a single delimiter character (which must be the same as the character which introduces **s** and terminates **t**).
The characters **[ ]** indicate an optional parameter.

A[s]	Insert line containing **s** after current line
B	Move to bottom of file
BE	Mark block end
BF[s]	Backwards find
BS	Mark block start
CE	Move cursor to end of line
CL	Move cursor one position left
CR	Move cursor one position right
CS	Move cursor to start of line
D	Delete current line
DB	Delete block
DC	Delete character under cursor
DW	Delete word right from cursor

E/s/t/ Exchange **s** into **t**
EQ/s/t/ Exchange but query first
F[s]   Forwards find
I[s]   Insert line containing **s** before current
IB   Insert copy of block
IF s   Insert file **s**
J   Join current line with next
M **n**   Move to line **n**
N   Move to next line
NW   Move to next word
P   Move to previous line
PW   Move to previous word
Q   Quit without saving text
QF   Quit current file without saving
R[s]   Read file **s**, keeping current file(s)
RC[s]   Read file **s** in cooked mode
RF   Reformat the text to fit into the window
RP   Repeat until error
RT   Return to previous line
S   Split line at cursor
SA[s]   Save text to file **s**
SC[s]   Save text to file **s** with CR/LF line endings
SB   Show block on display
SH   Show status
SL **n**   Set left margin
SR **n**   Set right margin
ST **n**   Set tab distance
T   Move to top of file
TY s   Type string **s**
U   Undo changes on current line
WB s   Write block to file **s**
X   Exit, writing text back

APPENDIX 2: FILE FORMAT AND MEMORY USAGE

QED is designed to handle line-based QDOS textfiles, which are files containing lines of printable characters terminated by a LF character.

QED restricts the maximum length of a displayed line to 254 characters; if you load a file containing longer lines these will be split to at most 254 characters (if Word wrap is disabled), or the right margin set if Word wrap is enabled. QED will give a warning message if it has done so.

The maximum size of the file you can edit depends on the amount of RAM in your QL (QED loads the whole file into RAM), but you must also ensure that your file does not contain more than 32767 **screen** lines (which is more than enough for most purposes). QED will however read files containing more than 32767 lines, but will ignore the extra lines. Again a warning message is displayed if this has happened.

Apart from the QED job code and data space (which are 11K and 1.5K respectively in the current version), QED uses two areas of RAM, both of which are allocated in the QDOS Common Heap area.

The first area is the textfile's workspace. This holds the text file in its original form, without any additional data. This has the advantage that LOADing and SAVing of the text file can be done very quickly using QDOS string I/O calls. However the disadvantage is that changing the length of a line near the beginning of a long textfile involves moving a large block of memory, which may slow down editing. (Note that editing the line itself does not take place in the text file but in a buffer, so this will not be slowed).
The QED code is optimised to speed this up as much as possible, but on an internal RAM machine it may take a second to enter a line near the start of a 160K textfile. If you find this too slow, remember that it is better to code a large program in parts which can be linked together by a linker (if possible), rather than in one very large source file. This will also save compiling/assembling time.

When a text file is loaded, QED will also allocate a line length table. This holds the length (one byte) of each line, enabling a particular line to be found quickly. In fact, if you want the cursor to be positioned at a particular line, QED will first determine the shortest path to take (which can be down from the top of the file, up or down from the current position, or up from the bottom of the file) and then search backwards or forwards for the line you want.

When the file is loaded, QED will allocate a line length table large enough for the text file itself plus an additional 256 lines. If during expansion of the file the table becomes full, it is reclaimed and a larger table built using the contents of the text file (this may take a few seconds). If the attempt to re-allocate the table fails due to shortage of memory, an error message "No room for line table" is displayed and any further editing will be prevented. However it is still possible to SAVE the file as the line length information is not required for SAVing.

APPENDIX 3: REVISION HISTORY

New features, bug fixes and known issues are listed in the text file CHANGES_TXT distributed along with QED.

=== END OF MANUAL ===