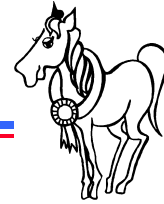


Qedit's Forgotten Commands



A Robelle Tutorial

By Hans Hendriks

August 1996



Copyright 1996, Robelle Solutions Technology Inc.

1

The Qedit full-screen editor for MPE and HP-UX has been in constant development for many years. Every new version contains many new features, so it isn't surprising that many experienced users are often unaware of better and easier ways to do common tasks. This tutorial will show you some of the commands and features that make Qedit more than just another text editor.

My name is Hans Hendriks. I've been a technical support consultant at Robelle since 1993. Before joining Robelle, I worked for many years as a Cognos and Robelle distributor, and have been a Qedit user since 1987.

Robelle Solutions Technology Inc. Toll-free: 1.888.762-3553
Suite 201, 15399 - 102A Avenue Telephone: 604.582.1700
Surrey, B.C. Canada V3R 7K1 Fax: 604.582.1799
E-mail: support@robelle.com
WWW: <http://www.robelle.com>

For Techies

References

For further information on topics covered in this tutorial, please consult the *Qedit User Manual*.

What's Inside

	<u>Page</u>
■ Line or command mode functions	3
■ Working with files	5
■ Hold files (Qedit's clipboard)	6
■ Strings and patterns	7
■ Formatting text	10
■ Column editing	13
■ Revising and repeating commands	14
■ Full-screen editing	16
■ Customizing Qeditmgr	18

This tutorial focuses on some of Qedit's many small features, rather than on a single, big block of functionality. It's broken down into three main areas: Line mode commands (also applicable on the homeline in full-screen mode), text formatting or word processing commands, and full-screen mode tricks and techniques.

This session is an opportunity for experienced users to share tips and techniques, so please feel free to contribute. Your ideas will make it interactive, interesting, and more valuable.

For Techies

References

Qedit commands

- Can be abbreviated to the shortest number of unique letters

- Shortcuts:

\	back one line]	last line
^	FINDUP	x/.y	x / x.y
[first line	Y	re-arm Control-Y trap
+5	*+5	\$	previous external file
CO	compile CM	%	external program
=	calculator	ZZ	marked range
?	Help	\$+/-	turn memory lock On/Off

- Function keys in Line mode

3

Qedit commands can be reduced to the minimum number of characters that uniquely identify them. In addition, you can use a number of other shortcuts that save typing. This slide contains a table of shortcut keys. To view them on-line, type **helpq shortcuts** or **hq s**.

Other shortcuts include the Qedit function keys that operate in Line mode, as well as in Visual mode. For example, F5 and F6 page through a file backward and forward respectively, and F3/F4 perform string searches.

Use "\$+" in column 1 to turn memory lock On, and "\$-" to turn memory lock Off.

For Techies

References

Configuring shortcut characters

- SET ZIP []@{} is default Zip list
 - Represents first, last, and all lines, left and right indentation
- An optional sixth character indicates auto-modify

```
set zip []@{}`
```
- Use Z to save entire commands

```
z=list *-5/*+5      {maximum 80 characters}
```
- Configure preferences in Qeditmgr file

4

You can change the defaults of the shortcut characters by setting Zip:

```
set zip []@{}      {default Zip list}
```

The position of these characters represents the following arguments:

[first line
]	last line
@	all lines
{	left indentation, used with AddJ
}	right indentation, used with AddJ

You can specify an optional sixth character that represents *auto-modify*.

A command or input line that ends with the auto-modify character is redisplayed for editing before it is executed.

Z can save and execute commands that are up to 80 characters long, including several commands separated by semicolons. If you pass parameters when you use Z, they are appended to its value.

You can customize your Qedit environment by adding the Set commands to your Qeditmgr file.

For Techies

References

For an example of a Qeditmgr file, see page 18.

Opening and closing files

- NEW, OPEN, TEXT commands
 - Closes current file, moves new file in for editing
 - OPEN * opens previous file; O * toggles back to current file
- Qedit version 4.4
 - Remembers last 10 files (e.g., O *-8)
 - OPEN *filename*,DEFER for unchanged modify date
- SET WORK LABELS ON or TEXT SUBFILE,LABELS
 - Allows editing of labelled files (e.g., PowerHouse subfiles)
 - KEEP preserves file code and labels

5

Qedit edits one file at a time. It also keeps track of up to 10 previously opened files (*, *-1, *-2, etc.), and the last external file referenced (\$). It is easy to toggle between the current and the previous files with `o *`, `o *-5`, etc. It is also easy to list and add lines from external files by using the "\$" shortcut.

```
/list template.doc  
/add 20 = $ 1/10
```

When you open a file, Qedit assumes that you want to make changes, so it opens the file with write access. This immediately updates the file's "modified date". You can override this by specifying ",Defer" when opening the file.

```
/open rdfile,defer {or Open Sdfile,Browse}
```

When you make a change to the file (Add, Modify, or Delete), Qedit reopens the file with write access. You can control the default open mode with

```
/set open defer ON|OFF
```

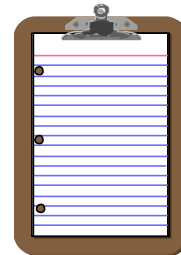
While Qedit always tries to remember the characteristics of files opened with the Text command, it ignores file labels by default. You can override this default by either specifying **labels** in your Text command, or changing the default with Set Work Labels On. If Labels becomes the default, Qedit copies the labels from the Text file and saves them when you use the Keep command. This default enables you to edit labelled files, such as PowerHouse subfiles.

For Techies

References

The Hold files -- Qedit's clipboard

- HOLD 20/30 copies lines to Hold file
HOLDJ 20/30 appends lines onto Hold file
- ADD includes an implied Hold
 - Hold0 is a temporary file
 - Applies only when ADD contains line range
`/add * = 45/52`
`/add] = tempt 1/10`
- Hold and Hold0 files can be explicitly referenced
`add * = HOLD`



6

Qedit makes it easy to move and copy text within, and between files. It provides two easy ways to repeatedly use the same text.

You can explicitly copy text to the Hold file by way of the Hold command. For example, you may have a template of a coding procedure that you want to repeatedly insert and edit while developing a program. You can put the template into the Hold file and copy it when required.

Whenever you add a range of lines to a file, Qedit first copies the lines to the Hold0 file. Then it counts the lines to determine the required line number increment for the file. If you need to repeatedly add the same lines, you can reference this Hold file directly. For example,

```
/add 27 = hold0
```

Remember that the next Add operation with a line range will overwrite the contents of the Hold0 file.

For Techies

References

Strings and patterns

- Strings can be used in many ways:
 - Global string changes
`/CHA "bug"undocumented feature" ALL`
 - Moving within files
`/FIND "feature"`
Last string specified becomes default (FIND, FINDUP, ^)
 - Qualifying a line range
`/delete "superfluous"`
- Combine strings into one command
`/change "bug"undocumented feature" "unexplained"`
Only change lines that contain "unexplained"

7

Strings play a variety of roles in Qedit. They can be delimited by quotes or by any of the following special characters:

	vertical bar
~	tilde
_	underscore
!	exclamation mark
#	pound sign
>	greater than sign
&	ampersand
:	colon

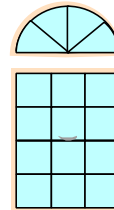
Most Qedit commands accept a string as a line range qualifier, in which case the commands apply only to lines that contain the string. This includes the Change command which can have three types of strings: change-from string, change-to string, and line range string.

For Techies

References

Qualifying strings

- Qualify non-matches with (NOMATCH) option
- Patterns as strings; you can use @, #, ~ and ? as pattern characters
`/delete "~" (pattern) {blank lines}`
- Strings within windows
`/find ".font" (1/5)`
- Exclude embedded strings
`/list "Frank" (smart) {ignores "Frankenstein"}`



8

By default, Qedit searches for matches to a string. If you specify (**nomatch**), it reverses the logic and returns lines that do not match. Likewise, you can restrict string searches to particular columns (e.g., 10/40) or ignore the case of the string (Upshift option).

If the string represents a pattern, you can specify any of these valid pattern characters:

@	anything, including nothing
#	single numeric digit
?	single alphanumeric digit
~	zero or more spaces

Qedit ignores strings embedded within larger words when you use the Smart option.

For Techies

References

Editing files containing escape sequences

- Working with non-printing characters

```
/set decimal on  
/list '7                {list lines with bell}  
/cha "^" '27 all       {change "^" to escape}
```

- ESCOFF.CMD command file

```
/set decimal on  
/cq '27 "^" all  
/set decimal off
```

- What if file already contains carets?



9

Qedit can edit non-printing characters by working with their decimal values. With Set Decimal On, any number prefaced with an apostrophe (') is interpreted as the character's decimal value.

When you edit files with escape sequences, it is often useful to change the escape characters to some readable character so that you can edit the text that follows. The Escoff command file in this slide changes escape (decimal 27) to "^". Make sure the file does not already contain caret (^) characters. How would you enhance this command file to check for pre-existing carets?

HINT: Qedit sets the Qeditcount JCW to the number of lines processed by the last command. This JCW is only set by commands that can print a total: List, Delete, Add, Change, Append, Justify, Keep, Proc, and Text.

For Techies

References

Formatting with Qedit

- Rapid-fire text entry with SET WRAPAROUND ON
 - Line wrap at word boundaries
 - Control-C to start new paragraph
 - "/" to end text entry
- Word wrap in Visual mode with R1DOS version 4.0 or higher
 - `/set vis wordwrap`
- ADD defaults to end of file
 - `/set addcurrent off`

10

Qedit was primarily developed as a home base for programmers. Since programmers also have to do documentation, Qedit includes some text formatting functions.

Set Wraparound On makes line overflow in the Add command more friendly by splitting long lines between words. Use Control-C to start a new paragraph, and Control-C twice to put a blank line between paragraphs. To complete an Add operation, use a double slash (/).

By default, Set Wraparound On allows you to enter up to 256 characters before pressing Return. You can increase this limit by specifying an alternate number, as long as it does not exceed 5,000 (e.g., Set Wraparound 1000).

Note that this word wrap feature only works in Line mode. If you have Reflection for DOS version 4.0 or higher, you can enable Reflection word wrap in Visual mode with Set Visual Wordwrap On.

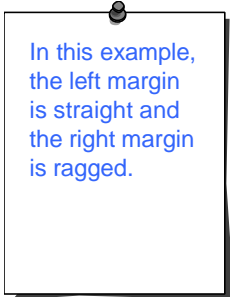
By default, the Add command inserts lines after the current line. You can use Set Addcurrent Off to change this default so that new lines are added to the end of the file.

For Techies

References

Justifying within margins

- SET JUSTIFY MARGIN 70
- SET JUSTIFY STOP "." START ""
 - `/justify left`
 - `/justify right 20/40`
 - `/justify centre 1/3`
 - `/justify format`
 - `/justify both 5/50`
- ADDJ 10
 - Automatically indents new lines
 - Adjust indentation with "{" and "}"



In this example, the left margin is straight and the right margin is ragged.

11

You can use the Justify command to center lines, left-justify or right-justify text, or fill text into margins. The Justify command is unusual in that you specify a single line number to indicate the beginning of the line range. The range either ends at the end of the paragraph, or at the justification stop indicated by the Set Justify Start/Stop characters. By default, a blank line indicates the end of paragraph.

You can justify the margins in just two simple steps:

1. Specify margins for justification
2. Enter one of these commands:

<code>/justify centre</code>	centre each line
<code>/justify left</code>	remove leading spaces
<code>/justify right</code>	straight right margin
<code>/justify format</code>	fill lines, ragged right margin
<code>/justify both</code>	fill lines, both margins straight

For hanging indents, use the Withindent keyword.

For Techies

Typically, you would configure your justification defaults by means of an appropriate Set Justify command in your Qeditmgr file.

References

Spelling Check functions

- SPELL 10/20 checks spelling in line range
- SPELLJ highlights errors, prompts for modification
- WORD looks up individual words

```
/WORD "right wrong david"
found : right
not found : wrong
wrong case : david
```
- Partial-word search

```
/WORD "quicks@"      {matches quicksand, quicksilver etc.}
```
- Soundex search

```
/WORD "quick!"      {matches quack, quake, quiche etc.}
```

12

Since 1991, Robelle has provided Spell as a free bonus program to its customers. You can invoke it from within Qedit via the Spell and Word commands. The Spell command checks a rangelist of lines for spelling errors against the main, auxiliary, and user dictionaries, and reports words that are not found or that have the incorrect case. The SpellJ command displays the offending line and highlights the error so that you can modify it. You can select the American or British spelling when installing the main dictionary.

There are two ways you can look up individual words with the Word command. You can either use a "partial-key" search by adding "@" to the end of the word, or a "sound-alike" search by adding "!" behind the word.

You can print the *Spell User Manual* with the following command,
:run printdoc.pub.robelle;info="spell.doc.robelle"

For Techies

References

For more information on configuring the Spell dictionaries and calling the Spell intrinsic from your applications, see the *Spell User Manual*.

Replace command enables column editing

- Displays current line; prompts for replacement

```
/replace 10/20
```

- Overlays from Hold file

```
/replace $hold 10/20
```

- Only replaces text in current window

Change 40 lines to 20 two-column lines:

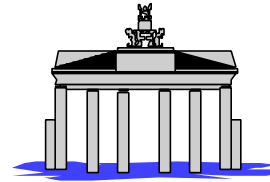
```
/hold 21/40
```

```
/delete 21/40
```

```
/set left 30
```

```
/replace $hold 1/20
```

```
/set left
```



13

Replace is an innocuous-looking command. At first glance, it doesn't seem to do much more than the Add command except overwrite an existing line and pass the line number to the newly-entered text. However, if you combine this with the fact that you can only overwrite text in the current window, it becomes apparent how the Replace command can enable column editing within Qedit.

In this example, lines 21 to 40 are first saved in the Hold file and deleted from the current file. The editing window is then changed to start at column 30 so that the Replace command will only overwrite the text in columns 31 onwards. By setting the left margin back to column 1 (default), you would be able to see that the leftmost columns remain untouched, and the rightmost columns contain the newly-pasted text.

For Techies

References

Redoing commands

- LISTREDO shows command history
 - Remembers last 1,000 commands
 - Shows last 20 commands
 - LISTREDO PURGE shows all PURGE commands
- REDO allows changes before executing command
- DO repeats command without changes
- MPEX-style shorthand to commands
 - "," - LISTREDO
 - "," - REDO
 - "," - DO

14

Qedit remembers your commands during each session. By default, it stores up to 1,000 commands, but you can increase this number by setting a file equation prior to running Qedit (e.g. :File Qedredo;Disc=5000).

Listredo	Shows your previous 20 commands
Listredo @	Displays all the commands that you entered in the current Qedit session
Listredo <i>string</i>	Lists all the commands in the Redo stack that begin with the <i>string</i> pattern
Do	Re-executes the specified line without first allowing you to modify it
Do <i>string</i>	Re-executes the last command that begins with the <i>string</i> pattern
Redo	Displays the specified line from the Redo stack; lets you edit the command before re-executing it

Alternatively, you can use Qedit's Before (B) command instead of Redo. BJ performs the same operation as Listredo, and BQ is equivalent to Do.

For Techies

Qedit also accepts MPEX-style shorthand characters for the Listredo (,), Redo (.), and Do (,) commands.

References

Saving useful commands in a command file

- Qedit can execute UDCs and command files
- Creating a command file:
 1. Send commands to a temporary file
`/listredo 35/42;unn;out=testcmd`
 2. Insert "/" in front of Qedit commands; command files can include MPE commands
 3. Save file on Hppath
- SET UDC in Qedit
`/set udc myudc {specific UDC file}`



15

Editing tasks are by nature repetitive. Since Qedit's goal is to simplify a programmer's work, Qedit offers an easy way to automate repetitive tasks with the Redo stack. You can echo useful commands to a disc file and build a command file that can later repeat the same task.

Listredo with the ;Unn option lists the commands as unnumbered lines, and with the ;Out=Testcmd option sends the output to a temporary disc file called Testcmd. Once they are in a file, you can insert "/" in front of Qedit commands and save them as a command file which you can invoke later.

Set UDC enables MPE/iX UDCs within your Qedit session. Qedit can also execute UDCs without setting them at the operating system level. While you are in Qedit, `set udc myudc` allows you to use the UDCs in the Myudc file without affecting other users. This enables you to use Qedit commands (prefaced with a "/") within your UDCs.

For Techies

References

Full-screen editing

- For fast text entry, delete template line;
end entry with "/" in columns one and two
- ==>"string" F7 {finds next occurrence of "string"}
 ==>"string" [12 F7 {finds 12th occurrence from start of file}
 ==>"string" 12 F7 {finds "string" after line 12}
 Use F3 for FINDUP, F4 for FIND
- ~ F7 {returns to recent screen}
- * F7 {refreshes current screen}
- . F7 {reset cut/paste operation}



16

When Qedit is updating a screen of data in Visual mode, it reads the screen memory until it finds a "/" in the leftmost two columns. Anything after "/" is ignored. Since Qedit cannot determine what is valid edited text without the "/", it cannot update the changes to the file. The absence of "/" explains the "No // at the end, so no update" error message that you may have seen.

You can actually use this feature to help you add text to the end of a file. Instead of inserting several blank lines, filling them, and then updating them, you can just delete Qedit's template line, type in the required text, and enter a "/" in the leftmost columns of the line following the text.

If you enter a string in quotations at the homeline and press F7, Qedit goes to the next line that contains the string. If you enter a number after the string, Qedit starts searching for the string from the specified line number. A string followed by "[" and a number (n) tells Qedit to start searching for the nth occurrence of the string from the beginning of the file.

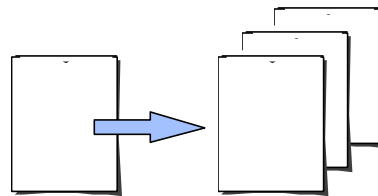
If you skip to a new position in a file and realize that it's not where you want to be, you can return to your previous position without knowing its line number. Simply enter "~" and press F7.

For Techies

References

Copy with multiple paste in Visual mode

- Mark block to be copied (CC/CC)
- Paste block in first location (A); copies text to Hold0 file
- A0 adds from Hold0 file thereafter



17

One often-requested Qedit enhancement for Visual mode involves multiple cut and paste operations within the same file. An implicit Hold is currently available with the Add operations that specify a line range. Copying (or moving) lines in Visual mode constitutes such an Add, and enables you to repeatedly add the same block of text from the temporary Hold0 file. A0 and B0 in the cut/paste columns indicate the position (After or Before) that you can insert text from the Hold0 file.

For Techies

References

Typical Qeditmgr file

```
set modify qzmodify
set zip [ ]@{ }`
set visual update on
set extprog MPEX.PUB.VESOFT
set udc myudc.udc.myacct
set justify margin 70
setjcw rlabeldefault,6
file phd=phd.dict.mis
file cogeditr=qedit.pub.robelle
set ynone on          {reply Y or N to yes/no questions}
set vis widen 80
set visual above 10 below 10
```

18

You can customize your Qedit environment according to your own tastes. Whenever you run Qedit, it automatically executes any commands in the Qeditmgr.Pub.Sys and Qeditmgr.Pub.Robelle files. If you run Qedit with the following Parm settings, it will execute these additional Qeditmgr files:

Parm = 1	Qeditmgr.Pub.Logon
Parm = 2	Qeditmgr.Yourgrp.Logon
Parm = 3	both files

See Chapter 5 (Running Qedit) in the *Qedit User Manual* for a detailed explanation.

Thank you for attending our Qedit tutorial. If you have any questions or need help, please call us at our toll-free number:

1-800-561-8311
7 a.m. to 4 p.m. (Pacific time)
Monday to Friday
Pager service available for after-hours emergencies.

For Techies

References