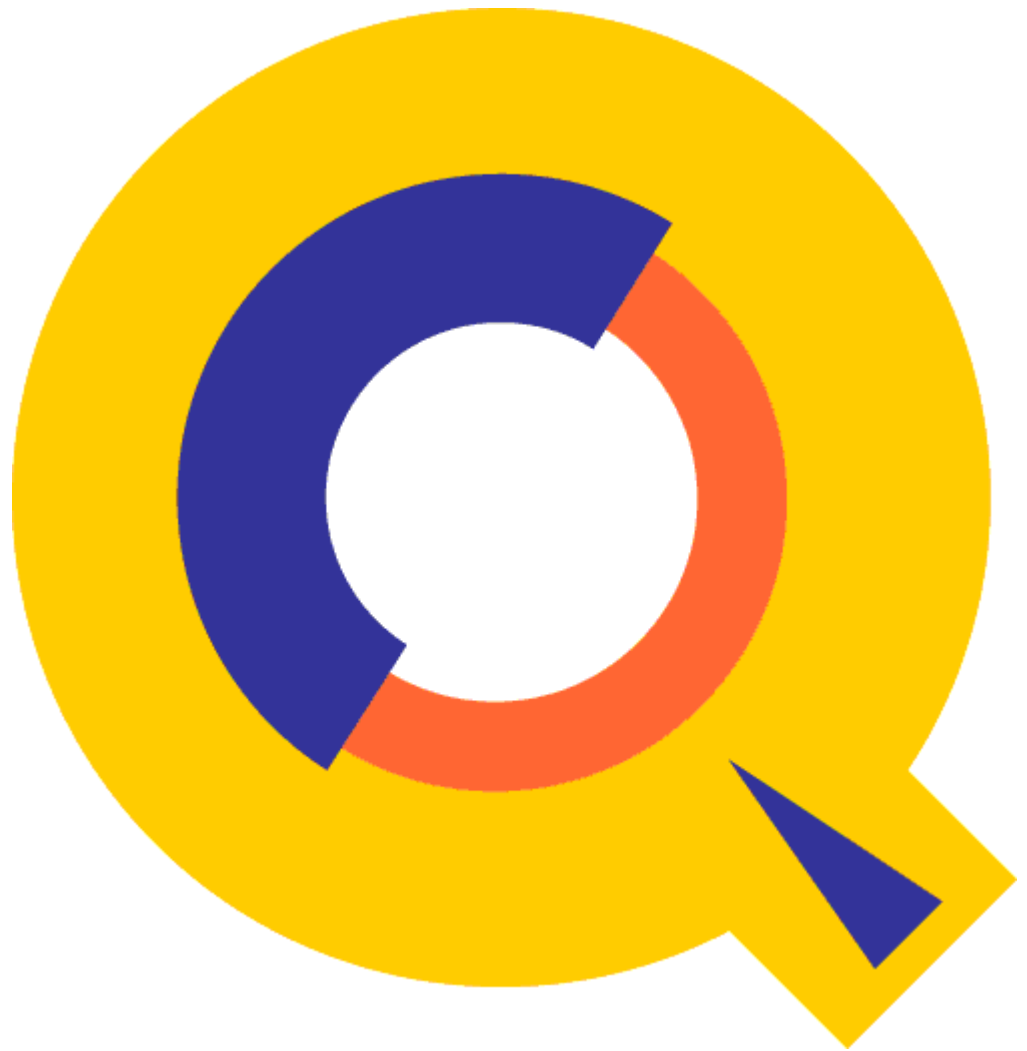


---

Qedit 5.4 for HP-UX

# Change Notice

by Robelle Solutions Technology Inc.



Program and manual copyright © 1977-2003 Robelle Solutions Technology Inc.

Permission is granted to reprint this document (but not for profit), provided that copyright notice is given.

Updated Thursday, April 10, 2003

Qedit and Suprtool are trademarks of Robelle Solutions Technology Inc. Windows is a trademark of Microsoft Corporation. Other product and company names mentioned herein may be the trademarks of their respective owners.



robelle  
*solutions technology*

Robelle Solutions Technology Inc.

13711 - 72nd Avenue, Suite 372

Surrey, B.C. Canada V3W 2P2

Phone: 604.582.1700

Fax: 604.582.1799

E-mail: [solutions@robelle.com](mailto:solutions@robelle.com)

E-mail: [support@robelle.com](mailto:support@robelle.com)

Web: [www.robelle.com](http://www.robelle.com)

# Contents

<b>Introducing Qedit/UX 5.4</b>	<b>1</b>
Introduction.....	1
Qedit for Windows.....	1
Highlights In Version 5.4.....	1
Line Mode .....	1
Full-screen Mode.....	2
Qedit for Windows Server.....	3
Compatibility.....	3
CPU Serial Number (uname).....	3
Known Problems .....	3
Documentation.....	3
User Manual .....	3
Change Notice .....	4
Printed Documentation.....	4
Online Documentation in HTML Help Format .....	4
CD or Web Download.....	4
<b>Installation</b>	<b>5</b>
Introduction.....	5
Installation Assistance .....	5
Step 1: Log On as Root.....	5
Step 2: Create Robelle Directory.....	6
Step 3: Restore Files .....	6
Step 4: Setting Up Variables.....	6
Step 5: Setting Up Links.....	7
Step 6: Start the Qedit for Windows Server (optional).....	7
<b>Enhancements In Version 5.4</b>	<b>9</b>
Introduction.....	9
String Range .....	10
Lines, Strings and Ranges .....	11
Delete With Confirmation .....	12
Most Recent Visual Line .....	12
Folding Wide Lines .....	13
Newline Delimiters .....	13
Fixed-length Cobol Source Files .....	14
Right Margin Control.....	14
Type 1234 Terminals and Emulators.....	15
Visual Blockemulation on HP-UX .....	16
List Nearest Line.....	17
<b>Qedit for Windows</b>	<b>19</b>
Introduction.....	19

Server Process.....	19
HP-UX 10.20/11.0.....	19
Port Number .....	20

**Bugs Fixed** **23**

Bugs Fixed Since Version 5.3.....	23
Search for String on Visual.....	23
Fixed-length Cobol Source Files .....	23
Host File Corruption.....	23
Temporary Window Not Used.....	23
Find and FindUp Aborts .....	24
Debugging Message .....	24
Large Cobol Source Files .....	24
Server Loop .....	24
File Permissions.....	24

# Introducing Qedit/UX 5.4

---

## Introduction

Qedit is our fast and efficient full-screen editor and development home base for HP-UX systems, both workstations and minicomputers. One of our objectives for Qedit is to give PowerHouse and COBOL developers all the necessary capabilities to perform their work from a single program.

We are constantly enhancing Qedit with new features and fixes requested by users. Not only does this change notice include a description of Qedit's enhancements, it includes detailed installation steps and a list of other changes for host-based Qedit. This change notice also includes a description of the server portion of Qedit for Windows.

Host-based Qedit is also available for MPE, with virtually identical features. If you would like more information, please contact Robelle Solutions Technology Inc.

---

## Qedit for Windows

Our newest editor, Qedit for Windows, now gives you the familiar Windows interface for editing both MPE/iX and HP-UX files, all from a single program. Qedit for Windows is the next generation of our popular editor, which consists of a Windows editing client and either an MPE/iX or HP-UX editing server that work together to edit host files. Since its release, we have focused our development efforts on Qedit for Windows, adding more functionality and continuing to expand its capabilities. If you would like to upgrade to Qedit for Windows, please contact Robelle Solutions Technology Inc.

---

## Highlights In Version 5.4

These are the changes made to Qedit since the last major release (version 5.3).

### Line Mode

- Most commands now support a string range to identify which lines to work on. A string range is made up of a start string (the first string of the block), a slash and a stop string (the last string of the block). For example, `List "Start-Procedure" / "End-Procedure"`

finds the first occurrence of "Start-Procedure", finds the next occurrence of "End-Procedure" from that point and displays all the lines between the 2 points including the start and end lines themselves. A string range can be further limited by specifying a line number range as in `List "Start-Procedure" / "End-Procedure" 10/50`.

- The **Text** command has a new option: **Length**. When specified, Qedit folds lines at the specified number of characters. This option makes it easier to handle all types of files whether they contain Newline delimiters or not.
- The **List** command has a new **\$length** option to perform a similar operation while listing a file.
- Newline delimiters are treated as data when the **Length** option is used on the **Text** command.
- **Set Keep LF ON** requests that the file be kept with Newline delimiters after each line. **Set Keep LF OFF** requests that the file be kept without Newline delimiters. Similar file formats are obtained using the **LF** and **NOLF** options of the **Keep** command.
- The tilde character "~" can be used in line-mode to reference the last line accessed in full-screen mode (**Visual**). To make it work as a line reference, it has to be removed from the string delimiter list with the **Set Stringdelimiter** command. If you enter it at the `Next` command [VISUAL] prompt, Qedit goes back into Visual immediately.
- **DeleteJ** prompts the user to confirm the deletion of each selected line. If confirmed, the line is deleted immediately. Deleted lines can be recovered with an Undo command. In batch, the lines are automatically deleted.
- The **List** command supports display wider than 266 characters. The new limit is now determined by the terminal emulator.
- Cobol source files can now be saved with fixed-length records using **Set Keep Var Off**. If required, you can use the **Set Text Cobolfixed** settings to force Cobol source files to always be read and saved as fixed-length files.
- If **Set Text Exclusive** is enabled, a user needs read and write permissions to be able to Text in a file. If he has read permission only, he has to use the **Browse** option on the **Text** command (Read-only in the Qedit for Windows client).
- Qedit and the server can now recognize large Cobol source files (more than 65,535 lines) and assign the Cobol language properly.

## Full-screen Mode

- Full-screen mode is now available on HP-UX 11.0 and later. **Set Visual Blockemulation On** emulates HP's block-mode feature available on earlier HP-UX versions. This feature does not work on the hpterm terminal emulator at the moment.
- Searching for a string on the **Visual** command e.g. `Visual "string"` while **Set Visual Home** is Off puts the cursor at the string location on the screen.

- Set the RCRTMODEL jcw to 1234 to identify a terminal or emulator supporting a subset of standard HP terminal features such as the hpterm terminal emulator running under X-window.
- **Set Visual Marginfixed** requests that Qedit never change the terminal right margin. The right margin should always be set at the physical display width.
- The value of **Set Visual Buf** is now displayed on a Verify output. In the past, the information was only displayed if it was different from the default.

---

## Qedit for Windows Server

- The server does a better job dealing with supplementary groups and related permissions.
- The server now honors file permissions properly.

---

## Compatibility

Qedit/UX is compatible with most versions of HP-UX 10.20 and later.

IMPORTANT: As of HP-UX 11.0, HP has dropped support for block-mode terminals. For this reason, full-screen editing is only available in Screen mode on HP-UX 11.0 and later.

In the examples in this document, we have used HP-UX 10.20 conventions and assumed Qedit is installed in the /opt/robelle directory.

---

## CPU Serial Number (uname)

Qedit/UX runs only on CPUs whose serial numbers (the "uname" on HP-UX) have been encoded. If it fails to run and you see an "invalid HPSUSAN" error message, call Robelle for assistance.

---

## Known Problems

There are currently no known problems.

---

## Documentation

Qedit comes with a User Manual and a Change Notice. You may have received printed copies of these.

They are also available as PDF or HTML Help files. You may have received a documentation CD with these files, or you can download the files from the Robelle web site.

### User Manual

The user manual contains the full description of all the Qedit commands, as well as usage tips. The manual is up-to-date with all the latest changes incorporated in Qedit.

To see only the changes in the latest version, see the "What's New" section of the manual, or see the change notice.

## Change Notice

For a complete description of the latest changes made to Qedit, the installation instructions, and any compatibility issues, see the change notice that was included with the release.

## Printed Documentation

The latest user manual and change notice are available in Adobe PDF format. If you do not already have the Adobe Acrobat Reader, you can get a copy from <http://www.adobe.com/prodindex/acrobat/readstep.html>.

## Online Documentation in HTML Help Format

The Qedit user manual and change notice are available in HTML Help format. Support for compiled HTML help (CHM files) is built into recent versions of Windows.

## CD or Web Download

If you received a documentation CD with this version of Qedit, you will find the documentation files on the CD. The installation program will start automatically when you insert the CD into the drive. If it does not start automatically, run *D*:\setup.exe, where *D* is the letter assigned to your CD-ROM drive.

If you do not have a documentation CD, you can download the files from the Robelle web site at <http://www.robelle.com/library/manuals/>.



# Installation

---

## Introduction

If you are a *new Qedit/UX user*, please follow the installation instructions in the *Qedit User Manual*. The notes below assume that you are an existing Qedit/UX user who is upgrading to version 5.3.13.

To install this upgrade of Qedit, follow these steps:

1. You must log on as root.
2. Create the correct directory structure.
3. Restore Qedit/UX and its associated files from the distribution tape.
4. Set up variables to let Qedit know where some of its supporting files are located.
5. Set up a link between /var/tmp and /usr/tmp if you are installing on HP-UX 10..x.
6. If you have the Qedit for Windows server, you need to start the daemon process. (optional)

The installation should take about half an hour, during which time no one can use Qedit.

---

## Installation Assistance

If you have any questions about the upgrade or run into any problems, please call us. Technical support is available on weekdays from 7 a.m. to 4 p.m., Pacific time.

---

## Step 1: Log On as Root

There are two ways you can log on as root:

1. Exit from HP-UX and log on with root as the user name.
2. If you are already logged on, you can execute this command:  
`su -`

In either case, you have to know and supply the user password for root.

---

## Step 2: Create Robelle Directory

Before restoring files, you must first create the directory in which Qedit will reside:

```
mkdir /opt/robelle
```

---

## Step 3: Restore Files

Use the following command to restore the Qedit/UX files from the distribution tape:

```
tar xv /opt/robelle
```

This command assumes your tape device is /dev/rmt/0m. If it is not, you need to specify your tape device using the "f" option in the tar command. For example, if your tape device is /dev/rmt/1m, you need to use the following command to restore the files:

```
tar xvf /dev/rmt/1m /opt/robelle
```

---

## Step 4: Setting Up Variables

You can set up numerous variables to make running any of the Robelle products easier.

The three variables that we recommend setting are ROBELLE, PATH and MANPATH.

- The ROBELLE variable tells the Robelle products where it's supporting files are located.
- The PATH variable, when set, allows the user to run Qedit by just typing "qedit" instead of the fully qualified program name.
- The MANPATH variable, when set, allows users to access the manpage (Help page) for Qedit.

You can invoke Qedit with the following command:

```
/opt/robelle/bin/qedit
```

The easiest way to add variables accessible to all users is to set the variables in the global exec file for the shell you use.

For the Bourne and Korn Shells we recommend the following additions:

```
export ROBELLE=/opt/robelle
export PATH=$PATH: $ROBELLE/bin
export MANPATH=$MANPATH: $ROBELLE/man
```

Remember to delete any PATH or MANPATH settings in /etc/d.profile, so that new users do not override your changes. You also have to warn existing Bourne and Korn shell users to change their .profile file in their home directories.

For the C Shell we recommend that you add the following commands after any existing PATH or MANPATH statements in the /etc/csh.login file:

```
setenv ROBELLE /opt/robelle
set path=($path $ROBELLE/bin)
setenv MANPATH "$MANPATH": $ROBELLE/man
```

Remember to delete any PATH or MANPATH settings in both /etc/d.login and /etc/d.schrc, so that new users do not override your changes. You also have to warn existing C shell users to change their .login and .cshrc files in their home directories.

---

## Step 5: Setting Up Links

When you run Qedit, it creates and uses certain temporary files. On HP-UX 10.x systems, temporary files are generally created in the /var/tmp directory. To get Qedit to keep an undo log and redo stack, you must set up a link between /usr/tmp and /var/tmp. To check whether this has already been done, enter

```
ls -l /usr/tmp
```

If a link exists, you will see a line such as this:

```
lrwxrwxr-t 1 root sys 8 Aug 16 1996 /usr/tmp -> /var/tmp
```

If the link is not set up, you can create it by entering

```
ln -s /var/tmp /usr/tmp
```

---

## Step 6: Start the Qedit for Windows Server (optional)

If you have the Qedit for Windows server software, you must start the Qedit/UX daemon process before any Qedit clients can connect to your HP-UX machine. To allow users to connect to the Qedit/UX daemon process, you must log on as root and issue this command:

```
qedit -d
```

The Qedit server process requires three log files. By default, these files are located in the following directory:

```
/opt/robelle/log/qedit/
```

If you have moved Qedit to a different directory, you will have to perform the following steps:

1. Set the ROBELLE environment variable with the new directory name before you start the Qedit server process.
2. Make sure the new directory has exactly the same structure as the /opt/robelle directory.



# Enhancements In Version 5.4

---

## Introduction

The following sections describe the new enhancements to Qedit/UX since the last major release. The number in brackets at the end of each section indicates the version in which the feature was first introduced.

- Most commands now support a string range to identify which lines to work on. A string range is made up of a start string (the first string of the block), a slash and a stop string (the last string of the block). For example, `List "Start-Procedure" / "End-Procedure"` finds the first occurrence of "Start-Procedure", finds the next occurrence of "End-Procedure" from that point and displays all the lines between the 2 points including the start and end lines themselves. A string range can be further limited by specifying a line number range as in `List "Start-Procedure" / "End-Procedure" 10/50`. [5.3.14]
- The **List** command supports display wider than 266 characters. The new limit is now determined by the terminal emulator.
- DeleteJ prompts the user to confirm the deletion of each selected line. If confirmed, the line is deleted immediately. Deleted lines can be recovered with an Undo command. In batch, the lines are automatically deleted. [5.3.14]
- Newline delimiters are treated as data when the **Length** option is used on the **Text** command. [5.3.14]
- The tilde character "~" can be used in line-mode to reference the last line accessed in full-screen mode (**Visual**). To make it work as a line reference, it has to be removed from the string delimiter list with the **Set Stringdelimiter** command. If you enter it at the **Next** command [VISUAL] prompt, Qedit goes back into Visual immediately. [5.3.14]
- The **Text** command has a new option: **Length**. When specified, Qedit folds lines at the specified number of characters. This option makes it easier to handle all types of files whether they contain Newline delimiters or not. [5.3.13]
- The **List** command has a new **\$length** option to perform a similar operation while listing a file. [5.3.13]

- **Set Keep LF ON** requests that the file be kept with Newline delimiters after each line. **Set Keep LF OFF** requests that the file be kept without Newline delimiters. Similar file formats are obtained using the **LF** and **NOLF** options of the **Keep** command. [5.3.13]
- If **Set Text Exclusive** is enabled, a user needs read and write permissions to be able to Text in a file. If he has read permission only, he has to use the **Browse** option on the **Text** command (Read-only in the Qedit for Windows client). [5.3.13]
- **Set Visual Marginfixed** requests that Qedit never change the terminal right margin. The right margin should always be set at the physical display width. [5.3.13]
- Set the RCRTMODEL jcw to 1234 to identify a terminal or emulator supporting a subset of standard HP terminal features such as the hpterm terminal emulator running under X-window. [5.3.13]
- The server does a better job dealing with supplementary groups and related permissions. [5.3.13]
- Full-screen mode is now available on HP-UX 11.0 and later. Set Visual Blockemulation On emulates HP's block-mode feature available on earlier HP-UX versions. This feature does not work on the hpterm terminal emulator at the moment. [5.3.12]
- **Set List Nearest** option determines whether a List command on a non-existent line number displays a warning or the nearest line. [5.3.10]
- The value of **Set Visual Buf** is now displayed on a **Verify** output. In the past, the information was only displayed if it was different from the default. [5.3.10]

---

## String Range

Most commands now support a string range to identify which lines to work on. A string range is made up of a start string (the first string of the block), a slash and a stop string (the last string of the block). For example, `List "Start-Procedure" / "End-Procedure"` finds the first occurrence of "Start-Procedure", finds the next occurrence of "End-Procedure" from that point and displays all the lines between the 2 points including the start and end lines themselves. A string range can be further limited by specifying a line number range as in `List "Start-Procedure" / "End-Procedure" 10/50`.

```

/list "print-item"/"edit-test" 36/
37.2 02-print-item.
37.3
37.4 read frdcust at end
37.5 move true-value to end-of-items-flag.
37.6
37.7 if not end-of-items then
37.8 move cust-account to report-account.
37.9 move name-first to report-name-first.
38 move zip to report-zip.
38.1 move report-record to line-record
38.2 write line-record after advancing 1 lines.
38.3 edit-test.

```

When used on a **Find** command, the ZZ marker is updated with the start and end line numbers as in:

```
/f "start"/"end" [ {find string range and set ZZ}
Lines 5/11 save in ZZ
```

## Lines, Strings and Ranges

Character strings can be used of line numbers to qualify lines on most commands. In its simplest form, a command can have a single string using all the search window defaults.

```
/List "enhancement"
```

The search string can be further qualified using temporary window settings as in:

```
/List "enhancement" (Upshift 20/50)
```

This example searches for the word `enhancement` regardless of the case used in columns 20 to 50.

Qedit allows up to 10 search strings on a single command. Individual strings are separated from each other with the **OR** keyword. Each string can have its own temporary window.

```
/List "enhancement" (U 20/50) or "bug" or "customer" (1/30)
```

The search range can be different depending on the command it is used on. For example, a **List** command searches all the lines in the file by default while a **Find** command starts from the current line. The search range can be specified on individual commands using a rangelist. A rangelist is often specified using line numbers (absolute or relative), special keywords (**First**, **Last**, **All**) or characters (**@**, **\***, **[**, **]**). To define a block of lines, the user can enter 2 line numbers separated a slash `"/` e.g. `1/6`.

It is also possible to define a block of lines using a string range. This syntax allows the use of strings to define the start and end of the range. A string range can also be combined with a numeric line range to further define the block. Here are some examples:

```
/List "start-proc" / "end-proc"
/Change "a" "b" "start-proc" / "end-proc"
/Delete "start-proc" / "end-proc" 20/100
```

The **List** command above finds the first occurrence of `start-proc` in the file and uses it as the range start location. It then finds the first occurrence of `end-proc` starting from the start location. It uses that line as the range end location. Finally, it lists all the lines between the 2 locations. By default, **List** starts at the beginning of the file.

The **Change** command above replaces all occurrences of the letter `a` with a `b` in the lines between (and including) `start-proc` and `end-proc`. By default, **Change** starts at the current line.

The **Delete** command above removes all the lines between (and including) `start-proc` and `end-proc` found in lines 20 to 100. By default, **Delete** starts at the beginning of the file.

A string range does not behave like a rangelist e.g. `1/20` in all cases. For example, the first statement is not a valid construct with the second statement is.

```
/Delete "bug" "start-proc"/"end-proc"  
Error: Linenum  
/Delete "bug" 10/30
```

You can use the **Find** command and the **ZZ** marker to work around the problem. If you enter a simple strings on a **Find** command, Qedit stops at the first string occurrence and sets the current line. You can then perform any operation on that line or use it as a starting point. If you specify a line range, the **Find** command sets the **ZZ** marker to the block of lines. You then use the **ZZ** marker on subsequent commands.

```
/F "start-proc" first  
5 Start-Procedure.  
(1)^  
/F "start-proc"/"end-proc" first  
Lines 5/11 saved in ZZ  
/Delete "bug" zz  
8 _bug-display-section.  
1 line Deleted!
```

[5.3.14]

---

## Delete With Confirmation

Use DeleteJ (DJ) to give yourself approval over each delete before it is carried out. With DJ, Qedit displays the line (even if the Quiet option is used) and asks you for a Yes, No, or Stop answer.

Answer **No** or Return to keep the line.

Answer **Yes** to delete the current line. Unlike the basic Delete operation where lines are removed with the next command, lines confirmed in DJ are deleted immediately. They can be recovered with an **Undo** command.

Answer **Stop** if you wish to stop the delete process. When you use Stop, lines that have been deleted are not recovered automatically. Use Undo to recover them.

In the following example, only line 4 is deleted.

```
/dj 3/66  
3 this is line 3  
Delete it (Y,N or Stop) [No]:  
4 this is line 4  
Delete it (Y,N or Stop) [No]:Y  
5 this is line 5  
Delete it (Y,N or Stop) [No]:n  
6 this is line 6  
Delete it (Y,N or Stop) [No]:S  
1 line Deleted!
```

[5.3.14]

---

## Most Recent Visual Line

The tilde character "~" can be entered on the Visual homeline to return to the "most recent" screen. If you jump from line 1500 to line 451, ~ sends you back to 1500. This is handy if you jump briefly to another part of your file to check something then want to get back to your original location.

The tilde is also available from line-mode but it has to be enabled by removing it from the list of string delimiters. In order to do this, you could do the following:



```
/V stringd
Set STRINGDelimiters "|\\~{}[]_@?!#>%&:' "
/S stringd "|\\{}[]_@?!#>%&:' "
```

Notice that tilde has been removed from the delimiter list entered on the Set command.

This is most useful when you occasionally enter line-mode commands on the homeline that change the current line pointer. For example, let's say you are editing line 1500 in full-screen mode and you enter `/List "display"` on the homeline. At the end of the list, the current line is going to be the last line listed, not 1500. To get back to line 1500, you only need to enter the tilde at the `Next Command [Visual]` prompt.

[5.3.14]

---

## Folding Wide Lines

There are 2 file types on UNIX: files with Newline delimiters at the end of each line and files without Newline delimiters. By default, Qedit/UX can not handle files without Newlines or files with lines longer than 8,172 characters. It is possible to edit these files by folding the content into manageable pieces. This is done using the **Length** option. Use this option to specify the size of each line. The maximum value is 8,172.

When reading the file in, each Qedit/UX read retrieves the specified number of characters until it reaches the end of the file. Lines will all have the same size except the very last line, if the total size of the file is not evenly divisible by the specified size. For example, if the file contains 8,000 characters and the specified Length is 80, Qedit/UX creates 100 lines. If the file contains 8,020 characters, there will be 100 lines of 80 characters and the last line will only have 20 characters.

If the Length option is used, Qedit/UX assumes the file does not have any Newline delimiters even if it actually had some. These characters are processed as if they were part of the data. In this case, Qedit/UX automatically disables **Set Keep LF**. To insert Newline delimiters, you can enable the option with

```
/Set Keep LF On
```

or use the **LF** option on the **Keep** command as in

```
/Keep myfile,lf
```

The List command has the `$Length` option to perform a similar operation when listing an external file. The `$length` option is used to specify the maximum number of characters per line.

```
/List $length 50 extfile.txt
```

In this example, all lines in `extfile.txt` are listed in 50-character pieces.

[5.3.13, 5.3.14]

---

## Newline Delimiters

On UNIX, some files have Newline (LF) delimiters at the end of each record and others don't. By default, Qedit/UX reads and writes files with Newline characters. To read files with no Newlines, you can use the **Length** option on the **Text** command or the **\$length** option on the **List** command.

To write Newline delimiters, you can use `Set Keep LF ON`. A delimiter is added at the end of each line whether there was one or not in the original file. The same function can be invoked using the `LF` option on the `Keep` command. The following `Keep` examples would then produce identical files:

```
/Set Keep LF On
/Keep firstfile.txt
/Keep secondfile.txt,lf
```

To create a file without Newline delimiters, you can use `Set Keep LF OFF` or specify `NOLF` on the `Keep` command. No delimiters are inserted at the end of each line.

```
/Set Keep LF Off
/Keep firstfile.txt
/Keep secondfile.txt,nolf
```

[5.3.13, 5.3.14]

---

## Fixed-length Cobol Source Files

By default, all UNIX files are processed as variable-length files. If needed, you can override this option using **Set Keep Var OFF**. Some Cobol compilers prefer to have fixed-length records. It can quickly become tedious to enter the `Set Keep` command after every **Text** command, not to mention the likelihood of forgetting to do it.

If you wish to force all Cobol source files to be processed as fixed-length files, use **Set Text Cobolfixed ON**. Every Cobol source texted in from that point will be fixed-length. If a file has already been texted in, the **Keep** command will switch to fixed-length records automatically. Qedit displays a warning in this case.

```
qux/v text
Set Text Exclusive OFF Cobolfixed OFF
qux/t mysource.cob
26 lines in file
qux/v k
Set Keep Ascii OFF Cctl OFF CCode 0 Lab 0 Num ON Var ON Checktimestamp
ON
Set Keep COBfree ON Name /users/robdev/qedit/test/file1CobFixed
Set Keep LF ON Saved modification timestamp 2002/11/29 14:04:26

qux/s text cobolfixed on
qux/k testcob.txt
Warning: Set Text Cobolfixed is On. File will have fixed-length
records.
/users/robdev/qedit/testcob.txt #Records = 26
Purge existing file [no]? Y

qux/set keep var on
qux/t mysource.cob
26 lines in file
qux/v k
Set Keep Ascii OFF Cctl OFF CCode 0 Lab 0 Num ON Var OFF Checktimestamp
ON
Set Keep COBfree ON Name /users/robdev/qedit/test/file1CobFixed
Set Keep LF ON Saved modification timestamp 2002/11/29 14:04:26
```

[5.3.13]

---

## Right Margin Control

Qedit normally adjusts the terminal right margin and display width based on the file's record length. This caused some terminal emulators like `hpterm` to behave erratically.

When Set Visual Marginfixed is enabled, Qedit does not change the terminal settings. It assumes the right margin is the physical display width. This should work properly as long as the user lets Qedit poll the terminal for the information. If the user decides to override this function by setting the RCRTWIDTH variable or change the width with **Set Term Columns**, Qedit trusts that the user has set the terminal properly.

Users should be aware of two things when Set Marginfixed is ON. If a file is narrower than the configured width, nothing prevents the user from entering text beyond the file's right edge. This extra text will simply be ignored.

If the file is wider than the configured width, Qedit tries to display as much text as it thinks it can. This causes the text to overflow. However, instead of truncating the extra characters, the emulator writes them out on the last displayable column. For example, if a file has 100 characters but the configured width is 80, the first 79 characters are displayed correctly. Characters 80 through 100 are written to column 80. The net result is characters 80 to 99 are lost and character 100 ends up in column 80 on the screen.

[5.3.13]

---

## Type 1234 Terminals and Emulators

Set the RCRTMODEL to 1234 if the terminal or emulator you are using does not support all the standard HP terminal features. For example, you should use this setting with hpterm. hpterm is a UNIX terminal emulator running under the X window system. It's a basic 2392 emulator. hpterm can not identify itself to Qedit, nor let Qedit change the display width by escape sequence (although you can configure the display width manually).

When RCRTMODEL is set to 1234 before you run Qedit, Qedit functions in the following manner:

1. It accepts the terminal as a terminal that is capable of more than 80 columns of display memory and of doing full-screen mode.
2. If you set RCRTWIDTH to some value between 81 and 256, Qedit accepts it as the manually set display width.
3. If you do not set RCRTWIDTH, Qedit attempts to sense the current display width and sets the jcw accordingly. The maximum width is 256 columns. Qedit can support up to 999 columns but, in these instances, the width has to be entered using the RCRTWIDTH variable or the **Set Term Columns** command.
4. Qedit sets the option that eliminates changes to display width: `Set MarginFixed On`.

This option also ensures that the right margin is always set at the right edge of the display width. Normally the right margin is set at the last valid column of the file, which might be less than the display width. You can use this option with other emulators if you wish to stop Qedit from changing the display width.

Please read the section on **Set Visual Marginfixed** to learn about its advantages and disadvantages.

5. If you use the **Set Term Columns** command to specify a new width, Qedit does not attempt to change the terminal with an escape sequence.

Instead, it displays the following message and waits for you to change the width manually:

```
Please change display width and press Enter:
```

Qedit does not verify that you have done this correctly, so if you make a mistake, do another Set Term Columns command to fix the width.

6. **Set Visual Stop** normally resets all the jcw's to their default state, forcing Qedit to re-identify the terminal. However, for hpterm, the RCRTMODEL and RCRTWIDTH jcw's are not reset, since the terminal cannot be identified automatically. If you wish to stop using **1234** mode, you must reset RCRTMODEL to 0 manually.
7. **Set Visual Widen** should normally be set to 76 or 80 (default) with hpterm. Otherwise you will not be able to use the extra columns beyond 80.

[5.3.13]

---

## Visual Blockemulation on HP-UX

*This feature does not work on the hpterm terminal emulator at the moment.*

As of HP-UX 11.0, HP has dropped support for block-mode terminals. For this reason, full-screen editing as implemented on HP3000 computers only works on HP-UX versions earlier than 11.0. On HP-UX 11.0 or later, full-screen editing is available in Screen mode (Set Visual Screen On) on VT-type terminals or in Visual mode emulation (Set Visual Blockemulation On) on HP-type terminals.

As its name implies, Blockemulation emulates block-mode operations by reading each line one by one instead of reading the whole screen in a single operation. Depending on the type of connection, this process might take a few seconds as the cursor moves down the screen.

In most cases, Qedit is able to detect that block-mode is not available and activates Visual Blockemulation automatically. If it does not, use the Set Visual Blockemulation command.

Visual Blockemulation also has the following limitations:

- Does not allow more than 20 contiguous blank lines
- Can not add more than 30 new lines at a time
- Can not display more than 260 characters on a line

Visual Blockemulation sometimes can not detect it has read all the lines. To work around this, Qedit assumes that 20 contiguous blank lines indicates a potential transmission problem and stops reading the screen. If you have to insert more than 20 empty lines, make sure you insert less than 20 and hit Enter. Repeat until you have all the lines you need. Of course, you can use the Add command outside of Visual mode.

Similarly, you can not add more than 30 new lines to the current screen before transmitting your changes. Make sure you insert lines in smaller numbers then hit Enter or use the Add command.

Visual Blockemulation can not display more than 260 characters on each line. To edit wider lines, change the Set Left and/or Set Right values.

[5.3.12]

---

## List Nearest Line

If you specify a single line number as a rangelist on a List command and that line does not exist in the current file, Qedit's action depends on the new Set List Nearest setting. If the option is Off, the default, Qedit displays a No Line warning. If the option is On, Qedit displays the nearest line. For example, if lines 100 to 120 are missing from a file, here is what would happen:

```
/List 100
Warning: No Line
/Set List Nearest On
/List 100
  121   This is line #121.
```

If you are trying to do something similar on an external file, Qedit does not display anything.

[5.3.10]



# Qedit for Windows

---

## Introduction

Qedit for Windows lets you edit local MPE/iX and HP-UX files from a single MS Windows program. It consists of a Windows editing client and either an MPE/iX or HP-UX editing server that work together to edit your host files. To take advantage of Qedit for Windows, you need both the Qedit client and the Qedit server.

Qedit for Windows uses the popular TCP/IP protocol for communicating between the client and the server (this is the same protocol that you use to access the Web). Configuring the Qedit server software requires creating the correct TCP/IP environment for Qedit for Windows.

---

## Server Process

Qedit clients can connect to Qedit/UX only if the Qedit/UX daemon process is running (the Qedit/UX daemon process cannot be started from inetd). This process listens for connections on a registered port number (described below). To allow users to connect to the Qedit/UX daemon process, you must log on as root and issue this command:

```
qedit -d
```

The Qedit/UX daemon process should always be running, so it is a good idea to automatically start the daemon as part of the system startup process. On HP-UX 9.0, this is done by adding the following command to the file `/etc/rc`:

```
/opt/robelle/bin/qedit -d
```

### HP-UX 10.20/11.0

On HP-UX 10.20 and up, you do not modify the startup shell script. Instead, you need to create a number of files. The HP-UX documentation states that the following characters cannot be used as part of the file names: `[ . , ~# ]`. Otherwise, you can choose any name for these files, as long as the names are consistent throughout the process. In our example, we use `qedit_server`.

You first need to create a control file in `/etc/rc.config.d`. This file sets a control variable that will be checked by the startup script. If the control variable is set to 1, the server will start; if it is not equal to 1, the server will not start. We will use

QEDIT\_SERVER as our variable name. The /etc/rc.config.d/qedit\_server control file will now contain the following:

```
# ***** File: /etc/rc.config.d/qedit_server *****
# Qedit for Windows server configuration.
#
# QEDIT_SERVER: Set to 1 to start
#               Qedit for Windows server
QEDIT_SERVER=1
```

Next, you need a shell script that will actually start the server. You can copy a template of the shell script from /sbin/init.d/template.

```
cd /sbin/init.d
cp template qedit_server
```

Modify the file so that it contains the necessary commands to start the server. You have to change all occurrences of CONTROL\_VARIABLE to the variable name you used in the control file (i.e., QEDIT\_SERVER).

You also need the execute command for the server program. Insert this command (underlined below) in the section after the 'start' string. The section looks like this:

```
'start')

    # source the system configuration variables
    if [ -f /etc/rc.config ] ; then
        . /etc/rc.config
    else
        echo "ERROR: /etc/rc.config defaults file MISSING"
    fi

    # Check to see if this script is allowed to run...
    if [ "$QEDIT_SERVER" != 1 ]; then
        rval=2
    else

        # Execute the commands to start your subsystem
        /opt/robelle/bin/qedit -d
    fi
;;
```

Finally, you need a symbolic link to specify when the script in /sbin/init.d will be executed at boot time. Typically, you would start the server as the last step at run level 3. Get a list of all the startup files in /sbin/rc3.d with

```
ls /sbin/rc3.d/S*
```

Link names in this directory follow a set of conventions. The names start with the letter S or K. S links are startup scripts; K links are shutdown or "kill" scripts. The next three characters in the name represent an execution sequence number. This number must be 3 digits, and its value should be a number greater than the highest value on the ls listing. For example, if the last link is called S100nfs.server, you could use S111qedit\_server. Create the symbolic link with

```
ln -s /sbin/init.d/qedit_server /sbin/rc3.d/S111qedit_server
```

For the time being, you do not need a "kill" link.

## Port Number

By default, Qedit/UX listens on port number 7395. This port number has been registered with the Internet Naming Authority, so you should not have any conflicts with other HP-UX tools for the same port number. If there is a conflict, you can start the Qedit server process with a different port number. For example,



```
qedit -d5678
```

tells Qedit to listen to port number 5678 instead of the default number (7395). If you change the port number on the Qedit/UX server, you must also change the port number on every Qedit client to the same value (5678 in this example). Client port numbers can be changed in the Server Settings dialog box of the Options menu.

If you want listings from netstat and other networking tools to identify the port number as "qwin" instead of just "7395," you have to change the /etc/services file so that it includes the Qedit port number.

```
qwin    7395/tcp    #Robelle Qedit for Windows
```



# Bugs Fixed

---

## Bugs Fixed Since Version 5.3

The following problems have been fixed since the last production version.

### Search for String on Visual

**Set Visual Home Off** requests that Qedit puts the cursor at the start of the search string occurrence. This worked fine once inside full-screen mode. It was not working on the initial search when initiated from the **Visual** command as in `Visual "string"`.

[5.3.13]

### Fixed-length Cobol Source Files

Set `Keep Var ON` now works when saving Cobol source files.

[5.3.13]

### Host File Corruption

Under certain conditions, the Qedit for Windows client's view of a file was different from the server's view. This could lead to corruption in the host file.

[5.3.12]

### Temporary Window Not Used

The `Change` command was not always using the temporary Window columns in the `rangelist` string. For example,

<code>Set Window (1/1)</code>	<code>{ Set the default window }</code>
<code>Change 1/1 "X" "ABC" (13/16).</code>	<code>{ Change with a temporary window }</code>

The `Set Window` command changes the default left and right columns. In this case, the window has only one column: 1. The `Change` command puts the letter X in column 1 on all lines that contain ABC in columns 13 to 16. These columns represent a temporary window. Starting in version 5.0.11, Qedit was not using the temporary settings which caused errors or incorrect file changes.

[5.3.10]

## Find and FindUp Aborts

Under certain circumstances, Qedit aborted on a Find or FindUp command. The problem occurred mostly in batch or from a session with the terminal set in 132-column mode. The string marker information does not display anymore if Qedit is running in batch.

[5.3.10]

## Debugging Message

The server displayed a debugging message in hexadecimal every time a file was opened. The message appeared on the session, if it was still active, from which the main server had been started. It only appeared if the trace feature was disabled. Now, the message is written to the trace file only if trace is enabled.

[5.3.10]

## Large Cobol Source Files

Normally, host-based Qedit and the UX server are able to recognize Cobol source files from the file extension (.cob, cbl) or by analyzing the first few lines of the file. However, if the source file contained more than 65,535 lines, Qedit changed the Language to Data instead of Cobol or Cobx. This is not the case anymore.

[5.3.10]

## Server Loop

The server went into a loop, filling up the error logfile. A number of conditions had to be in place for this problem to manifest itself:

- a connection was interrupted e.g. communication problem
- the Set Text Exclusive feature was enabled
- a workfile was dirty i.e. not saved.

If a connection is interrupted, the server now closes all the files. Qedit workfiles should contain all the latest changes. These changes can be recovered by opening the workfiles (QED##### on MPE, qscr.\* on HP-UX).

[5.3.10]

## File Permissions

In some cases, the server did not act on file permissions the same way terminal sessions did e.g. telnet.

[5.3.11]